

**JavaScript programuotojo modulinė profesinio mokymo programa, IV lygis**

**Teorinių ir praktinių užduočių**

**mokinio sąsiuvinis**

Teorinių ir praktinių užduočių mokinio sąsiuvinis parengtas įgyvendinant iš Europos Sąjungos struktūrinių fondų lėšų bendrai finansuojamą projektą „Lietuvos kvalifikacijų sistemos plėtra (I etapas)“ (projekto Nr. 09.4.1-ESFA-V-734-01-0001).

Teorinių ir praktinių užduočių mokinio sąsiuvinio (JavaScript programuotojo modulinė profesinio mokymo programa, IV lygis) autoriai patvirtina, kad šiame teorinių ir praktinių užduočių mokinio sąsiuvinyje pateiktos užduotys nepažeis autorių, kurių kūriniai naudojami, teisių ir visa užduotims rengti ir iliustruoti naudota literatūra ir šaltiniai yra pateikti sąsiuvinio gale.

Teorinių ir praktinių užduočių mokinio sąsiuvinio autoriai:

Virginija Putnaitė

Laura Vaišnorė

Rita Daukšienė

Jaroslav Grablevski

Jolita Bačkienė

Justina Balsė

Konsultantai:

Martynas Kašelionis

Linas Būtėnas

Žydrūnas Tamašauskas

**Modulis** **„Įvadas į profesiją“**

***TESTAS ĮSIVERTINTI GEBĖJIMAMS PRIEŠ PRADEDANT MOKYTIS***

1. Ar terminai: žiniatinklis (World Wide Web) ir internetas yra sinonimai:

a) Taip

b) Ne

2. Koks protokolas naudojamas komunikavimui tarp naršyklės ir serverio?

a) HTML

b) Ajax

c) HTTP

3. Ar naršyklė yra būtina norint nusiųsti POST užklausą į aplikaciją aptarnaujantį serverį?

a) Taip

b) Ne

4. Pagrindinės WEB technologijos?

a) C++, JavaScript, TypeScript, HTML

b) HTML, CSS, JS, OS X, PHP

c) HTML, CSS, JavaScript, PHP

5. Kokia technologija leidžia atnaujinti HTML dokumento dalį neperkraunant viso dokumento (pvz.: įvykus įvykiui, kai gauta nauja informacija)?

a) HTML „dynamic“ extension biblioteka

b) Ajax

c) SASS

6. HTML, tai:?

a) Programavimo kalba tinklapio interaktyvumui kurti

b) Tinklapio apipavidalinimą aprašanti kalba

c) Tinklapio struktūrą aprašanti kalba

7. Kodėl CSS vadinami pakopiniais stiliais?

a) Kodas rašomas pakopomis

b) Pakopos nusako kuri taisyklė yra svarbesnė ir bus taikoma elementui

c) Teisingo atsakymo nėra

8. Kas bus išvesta įvykdžius pateiktą žemiau esantį kodą?

class Vartotojas {

public $vardas;

public $pavarde;

public function pasisveikink()

{

return "Labas, " . $this -> vardas;

}

}

$vartotojas1 = new User();

$vartotojas1 -> vardas = "Jonas";

$vartotojas1-> pavarde = "Jonaitis";

echo $vartotojas1 -> pasisveikink();

a) Jonas Jonaitis

b) Jonas

c) Klaida

9. Ką vadiname CSS preprocesoriumi?

a) Preprocesorius leidžia rašyti css kodą html elemento viduje

b) Preprocesorius yra procesas, kuris serveryje įvykdo CSS kodą

c) Preprocesorius - skriptų rašymo kalba, kuri išplečia CSS ir leidžia kompiliuoti kodą į standartinį CSS kodą

10. Kuris CSS selektorius yra teisingas norint stilių taikyti tik nuorodmis į pdf tipo failus?

a) a[href >='.pdf']

b) a[href $='.pdf']

c) Nėra teisingo atsakymo

11. Kuris iš kintamųjų yra masyvas?

a) var pastraipos = document.querySelector('p')

b) var patarimai = document.querySelectorAll('p')

c) var pastraipos = document.intoArray('p)

12. Kaip kintamajam antraštė, kuriam priskirtas elementas HTML elementas header pakeisti fono spalvą naudojant JavaScript?

a) antraste.style.background-color = 'tomato'

b) antraste.style.backgroundColor = 'tomato'

c) antraste.css.background-color= 'tomato'

13. HTML5 naudojam semantinius elementus. Koks semantinis elementas aprašo tinklapio navigacijos bloką?

a) navigation

b) links

c) nav

14. Turim 5 poras juodų ir 5 poras baltų pirštuotų pirštinių. Kiek maksimaliai traukimų reikės, kad užtikrintai turėtume vienos spalvos teisingą porą pirštinių?

a) 5

b) 10

c) Teisingo atsakymo nėra

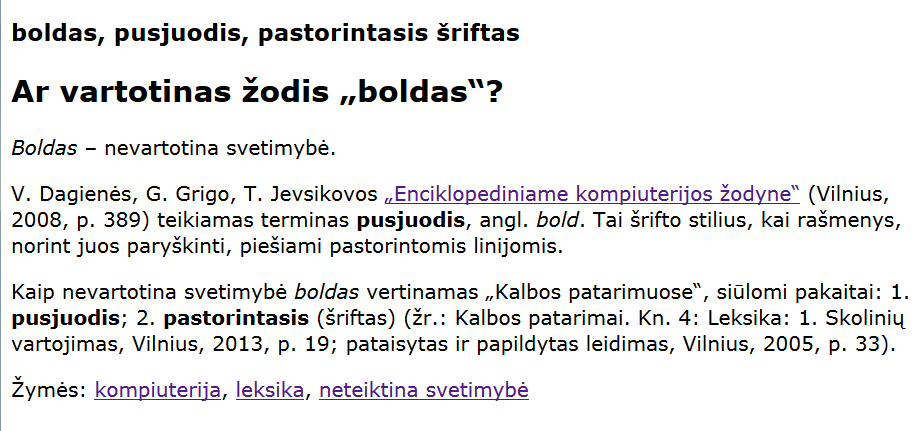
**Modulis „Informacinių sistemų projektavimas ir kūrimas (JavaScript)“**

3. **Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.**
   1. **Mokymosi rezultatas. Pateikti internetinio puslapio turinį naudojant kompiuterinę žymėjimo kalbą.**
      1. **Tema. Pagrindinės HTML kalbos žymės.**

*1 užduotis.* PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (STRAIPSNIS).

UŽDUOTIES SĄLYGA: naudojant HTML antraščių, pastraipos, teksto formatavimo, nuorodų elementus, sukurkite tinklapį pagal pateiktą pavyzdį.

**Pavyzdys:**



UŽDUOTYJE NAUDOJAMAS TEKSTAS:

boldas, pusjuodis, pastorintasis šriftas

Ar vartotinas žodis „boldas“?

Boldas – nevartotina svetimybė.

V. Dagienės, G. Grigo, T. Jevsikovos „Enciklopediniame kompiuterijos žodyne“ (Vilnius, 2008, p. 389) teikiamas terminas pusjuodis, angl. bold. Tai šrifto stilius, kai rašmenys, norint juos paryškinti, piešiami pastorintomis linijomis.

Kaip nevartotina svetimybė boldas vertinamas „Kalbos patarimuose“, siūlomi pakaitai: 1. pusjuodis; 2. pastorintasis (šriftas) (žr.: Kalbos patarimai. Kn. 4: Leksika: 1. Skolinių vartojimas, Vilnius, 2013, p. 19; pataisytas ir papildytas leidimas, Vilnius, 2005, p. 33).

Žymės: kompiuterija, leksika, neteiktina svetimybė

* + 1. **Tema. HTML kalbos turinio žymės.**

*2 užduotis.* PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (KELIONIŲ AGENTŪRA).

UŽDUOTIES SĄLYGA: naudojant HTML antraščių, pastraipos, sąrašų ir kitus elementus, sukurkite puslapį iš kelių tinklalapių, susietų nuorodomis.

**Reikalavimai:**

1. Failų pavadinimai: index.html, poilsines-keliones.html, kelioniu-datos.html.
2. Grafiniai vaizdai saugomi aplanke, pavadinimu IMG.

ATSKIRŲ TINKLAPIO PUSLAPIŲ TURINYS:

-----------------------------------------------------------------

**Kelionės**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Paskutinė minutė**

Ispanija, Kosta Dorada



Paskutinės minutės atostogos Ispanijoje, Salou! Skrydis iš Vilniaus, bagažas, pervežimai ir 7 n. viešbutyje su pusryčiais ir vakarienėmis – 379 €!

**Išvykimo laikai:**

* Vilnius – Barselona 14:00 – 16:30
* Barselona – Vilnius 17:30 – 22:00

**Į kelionės kainą įskaičiuota:**

1. Skrydis Vilnius – Barselona – Vilnius;
2. Pervežimas oro uostas – viešbutis – oro uostas;
3. 20 kg registruojamas ir rankinis bagažai;
4. 7 n. apgyvendinimas pasirinktame viešbutyje su nurodytu maitinimo tipu.

-----------------------------------------------------------------

**Kelionės**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Poilsinės kelionės**

Ispanija, Kosta Dorada



Paskutinės minutės atostogos Ispanijoje, Salou! Skrydis iš Vilniaus, bagažas, pervežimai ir 7 n. viešbutyje su pusryčiais ir vakarienėmis – 379 €! Plačiau...

Kreta



Rugsėjį keliauk į vaizdingąją Kretą! Skrydis iš Vilniaus, bagažas, pervežimai ir 7 n. viešbutyje su „viskas įskaičiuota“ – 499 €! Plačiau...

-----------------------------------------------------------------

**Kelionės**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Kelionių datos**

|  |  |  |
| --- | --- | --- |
| **Data** | **Kryptis** | **Kaina** |
| Rugsėjo 18 - Rugsėjo 28 | Turkija | 345 € |
| Rugsėjo 20 - Rugsėjo 25 | Kipras | 299 € |

-----------------------------------------------------------------

*3 užduotis.* PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (NAUJIENŲ SRAUTAS).

UŽDUOTIES SĄLYGA: naudojant HTML antraščių, pastraipos, sąrašų ir kitus elementus, sukurkite vieno puslapio tinklalapį su vidinėmis nuorodomis. Tinklapio turinys: laisvas pasirinkimas.

**Reikalavimai:**

1. Failų pavadinimai: index.html.
2. Grafiniai vaizdai saugomi aplanke, pavadinimu IMG.
3. Turinį gaubti į struktūrinius elementus.
   1. **Mokymosi rezultatas. Apipavidalinti internetinį puslapį naudojant pakopinius stilių šablonus ir karkasus.**
      1. **Tema. CSS pagrindai.**

*1 užduotis.* UŽRAŠYKITE CSS STILIAUS TAISYKLES (CSS SELEKTORIAI).

UŽDUOTIES SĄLYGA: naudojant skirtingas CSS selektorių kategorijas, aprašykite antraščių, pastraipų ir kt. elementų nurodytus stilius.

Duoti failai:

1. HTML (index.html)

<html>

<head>

<meta charset="UTF-8">

<title>Selectors Exercise</title>

</head>

<body>

<h1>Selectors Exercise</h1>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<div>

<p class="hello">I am a paragraph with a class</p>

<p id="special">I am a paragraph with an ID</p>

<h2>I am an awesome h2 </h2>

<h3>I am an awesome h3 </h3>

<p id="special2">Roof party yr hella synth, Wes Anderson narwhal four dollar toast before they sold out retro lo-fi. Austin iPhone pop-up farm-to-table, PBR McSweeney's ennui messenger bag distillery before they sold out Portland wolf fanny pack YOLO. Locavore slow-carb trust fund farm-to-table. Pinterest gastropub lo-fi, McSweeney's trust fund VHS shabby chic ugh Austin twee. Messenger bag banjo lumbersexual, whatever 3 wolf moon <span>XOXO (red)</span> normcore. Pug fanny pack 3 wolf moon, typewriter organic chia mustache scenester seitan shabby chic Blue Bottle salvia ugh iPhone. Fanny pack Williamsburg direct trade, cold-pressed disrupt flannel listicle health goth asymmetrical freegan mixtape street art pour-over whatever.</p>

</div>

<div>

<h2>Things I need to do </h2>

<ul>

<li>Walk Dog <input type="checkbox" checked> </li>

<li>Feed Dog <input type="checkbox" checked> </li>

<li>Wash Dog <input type="checkbox"></li>

</ul>

</div>

<div>

<h2 class="hello">I am another awesome h2 </h2>

<p>Cardigan Tumblr mlkshk, fap tilde 3 wolf moon Portland. Heirloom health goth taxidermy blog lo-fi selfies, post-ironic master cleanse fingerstache normcore. Kickstarter plaid twee, bespoke single-origin coffee sustainable lo-fi vinyl Pinterest pork belly <em>cronut skateboard</em> 3 wolf moon. Normcore single-origin coffee salvia, bespoke Austin swag Godard before they sold out kogi disrupt locavore. Lumbersexual Shoreditch Vice, artisan American Apparel master cleanse yr salvia vegan. Bespoke letterpress <span>XOXO</span> heirloom kale chips deep v four loko. Lomo sustainable put a bird on it trust fund post-ironic</p>

<p>I'm the second paragraph inside this div!</p>

</div>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<div>

<h2>A less awesome h2 </h2>

<p>Roof party yr hella synth, Wes Anderson narwhal four dollar toast before they sold out retro lo-fi. Austin iPhone pop-up farm-to-table, PBR&B McSweeney's ennui messenger bag distillery before they sold out Portland wolf fanny pack YOLO. Locavore slow-carb trust fund farm-to-table. Pinterest gastropub lo-fi, McSweeney's trust fund VHS shabby chic ugh Austin twee. Messenger bag banjo lumbersexual, whatever 3 wolf moon XOXO normcore. Pug fanny pack 3 wolf moon, typewriter organic chia mustache scenester seitan shabby chic Blue Bottle salvia ugh iPhone. Fanny pack Williamsburg direct trade, cold-pressed disrupt flannel listicle health goth asymmetrical freegan mixtape street art pour-over whatever</p>

<p>One last paragraph here!</p>

<p> List inside a DIV</p>

<ul>

<li>Dog

<ul>

<li>Dog Dog</li>

<li>Dog Dog</li>

</ul>

</li>

<li> Cat </li>

<li> Mouse

<ul>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

</ul>

</li>

</ul>

</div>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<p> List NOT inside a DIV</p>

<ul>

<li>Dog

<ul>

<li>Dog Dog</li>

<li>Dog Dog</li>

</ul>

</li>

<li> Cat </li>

<li> Mouse

<ul>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

</ul>

</li>

</ul>

</body>

</html>

1. CSS (style.css)

/\*

Style the HTML elements according to the following instructions.

WRITE ONLY CSS!

\*/

/\* Give the <body> element a background of lightgray\*/

/\* <body> elemento fono spalva: lightgray \*/

/\* Make the <h1> element purple \*/

/\* <h1> elemento spalva: purple \*/

/\* Make all <h2> and <h3> elements orange \*/

/\* Visų <h2> ir <h3> elementu spalva: orange\*/

/\* Make all the <p>'s that are nested inside of divs 25px font(font-size: 25px) \*/

/\* Visų <p> elementų, esančių <div> elemente, šrifto dydis: 25px \*/

/\* Give everything with the class 'hello' a white background\*/

/\* Visų elementų, turinčių klasę "hello", fono spalva: white\*/

/\* Give the element with id 'special' a 2px solid blue border \*/

/\* Elemento, turinčio id "special", rėmelis (border): 2px solid blue \*/

/\* Make the <span> element in <p> element with id 'special2' a 2px solid red border \*/

/\* <span> elemento, esančio <p> elemente su id "special2", rėmelis: 2px solid red \*/

/\* Change the style of multilevel list in <div> element: first level of list darkblue, second - pupple \*/

/\* Kelių lygių sąrašo, esančio div elemente, pirmojo lygmens spalva: darkblue, antrojo: pupple \*/

**Reikalavimai:**

1. HTML dokumente, HEAD dalyje, įterpti nuorodą į CSS failą.
2. CSS faile aprašyti stiliaus taisykles pagal reikalavimus.

*2 užduotis.* UŽRAŠYKITE CSS STILIAUS TAISYKLES (RECEPTAI).

UŽDUOTIES SĄLYGA: naudojant skirtingas CSS selektorių kategorijas, aprašykite antraščių, pastraipų ir kt. elementų nurodytus stilius.

Duoti failai:

1. HTML (index.html)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Recipes</title>

</head>

<body>

<header class="container">

<h1>Recipes</h1>

<h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit. </h3>

<nav>

<ul>

<li><a href="#">sandwiches</a></li>

<li><a href="#">cupcakes</a></li>

<li><a href="#">hot drinks</a></li>

</ul>

</nav>

</header>

<main>

<article>

<section class="container description">

<h3>Description</h3>

<p>This peanut butter and jelly sandwich is my favorite sandwich. It has the perfect balance of ingredients and looks great when made right.</p>

</section>

<section class="container">

<img src="https://image.flaticon.com/icons/png/512/63/63035.png" alt="sandwich" width="200" height="200">

<h3>Ingredients</h3>

<ul>

<li>2 slices of white bread</li>

<li>1 jar of grape jelly</li>

<li>1 jar of creamy peanut butter</li>

<li>A butter knife</li>

<li>A sharp knife</li>

<li>A cutting board</li>

</ul>

</section>

<section class="container description">

<h3>Directions</h3>

<ol>

<li>Lay both slices of bread next to each other on a cutting board.</li>

<li>With the butter knife, spread a 1/8 inch layer of peanut butter on the left side</li>

<li>On the right side, spread a 1/8 inch layer of jelly.</li>

<li>Carefully place the two halves together so that the jelly is on top.</li>

<li>With the sharp knife, carefully cut the sandwich in half.</li>

<li>Enjoy the PB&amp;J!</li>

</ol>

</section>

</article>

</main>

<footer class="container">

<nav>

<ul>

<li><a href="#">sandwiches</a></li>

<li><a href="#">cupcakes</a></li>

<li><a href="#">hot drinks</a></li>

</ul>

</nav>

<h5>Copyright &copy; 2008 — 2018 WEBMASTER. </h5>

</footer>

</body>

</html>

2. CSS (style.css)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Numatytųjų naršyklės stiliaus taisyklių perrašymas

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Visos išorinės paraštės (margin) lygios 0 \*/

\* {

    margin:0;

}

/\* Šrifto tipo ir dydžio taisyklės visam tinklapiui \*/

body{

    font-family: 'Questrial', sans-serif;

    font-size: 12pt;

}

/\* Kelios klasės \*/

.description{

    background-color: #F3F3F3;

}

.container{

    padding: 30px 10px;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Tinklapio antraštė

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* header:

    fono spalva: #16A085,

    teksto - balta.

\*/

/\* <h1>, esantis <header> dalyje:

    tekstas: visos didžiosios raidės,

    vidinės paraštės (padding): 20px 0,

\*/

/\* <h3>, esantis <header> dalyje:

    tarpai tarp raidžių: 0,3 em

\*/

/\* <nav>, esantis <header> dalyje:

    viršutinis rėmelis: 2px dashed white;

    viršutinė vidinė paraštė: 20px;

\*/

/\* nuoroda, esanti <header> dalyje:

    spalva: balta;

    teksto pabraukimo nėra;

\*/

/\* sąrašas, esantis <header> dalyje:

    sąrašo ženklinimo nėra;

    visos vidinės paraštės lygios 0;

\*/

/\*

sąrašas išdėstytas eilutėje:

\*/

header nav li{

    display: inline-block;

    width: 150px;

}

/\* Pseudo klases nuorodoms, <header> dalyje \*/

/\* :link

    teksto spalva: balta;

\*/

/\* :visited

    teksto spalva: balta;

\*/

/\* :hover

    teksto spalva: balta;

    tesktas pabrauktas;

\*/

/\* :active

    teksto spalva: pilka;

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Tinklapio pagrindinė dalis | Main

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* <h3>, esantis <article>:

    tekstas: visos didžiosios raidės;

    teksto spalva: #16A085;

    tarpai tarp raidžių: 0,3em;

\*/

/\* <ul>, esančio <article>:

    ženklinto sąrašo stilius: kvadratas;

\*/

/\* <ol>, esančio <article>:

    numeracijos stilius: mažosios romėniškos raidės;

    eilutės aukštis: 2;

\*/

/\* pirmoji <section>, esanti <main> dalyje:

    teksto lygiavimas: centruotas (center);

\*/

/\* paveiklas:

    rėmelis: 1px solid #16A085;

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Tinklapio baigiamoji dalis | footer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* <footer> dalys:

    teksto lygiavimas: centruotas;

    fono spalva: #16A085;

\*/

/\* nuorodos, esančio <footer>:

    teksto pabraukimo nėra;

    teksto spalva: balta;

\*/

/\* <h5>, esančio <footer> dalyje:

    viršutinės vidinė paraštė: 20px;

\*/

**Reikalavimai:**

1. HTML dokumente, HEAD dalyje, įterpti nuorodą į CSS failą.
2. CSS faile aprašyti stiliaus taisykles pagal reikalavimus.
   * 1. **Tema. CSS tinklalapio maketo kūrimo technikos.**

*3 užduotis.* UŽRAŠYKITE CSS KLASES (PARAMETRAI „FLOAT“ IR „CLEAR“).

UŽDUOTIES SĄLYGA: įterpti į HTML dokumentą jau aprašytų klasių atributus. Duoti HTML (index.html) ir CSS (style.css) failai.

**Reikalavimai:**

1. HTML dokumente, HEAD dalyje, įterpti nuorodą į CSS failą.
2. HTML dokumente įterpti atitinkamų klasių atributus.
3. **Kompetencija. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.**
4. 1. **Mokymosi rezultatas. Suprasti skaičiavimo sistemas.**
      1. **Tema. Skaičiavimo sistemos.**

*1 užduotis.* APRAŠYKITE KAIP NUMERAVO SENOVĖS BABILONIEČIAI, EGIPTIEČIAI, GRAIKAI, SLAVAI.

*2 užduotis.* SURAŠYKITE KOKIOSE ŽMOGAUS VEIKLOS SRITYSE SKAIČIAVIMO SISTEMOS TURI DIDŽIAUSIĄ REIKŠMĘ.

*3 užduotis.* SURAŠYKITE VISAS JUMS ŽINOMAS SKAIČIAVIMO SISTEMAS (POZICINĖS).

*4 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 125, 256, 2019 DVEJETAINIAIS SKAIČIAIS.

*5 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS DVEJETAINĖS SISTEMOS SKAIČIUS: 101100110001, 1010101, 001110011.

*6 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 14, 56, 2019 AŠTUNTAINIAIS SKAIČIAIS.

*7 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS AŠTUNTAINĖS SISTEMOS SKAIČIUS: 77, 66, 32.

*8 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 62, 216, 2019 ŠEŠIOLIKTAINIAIS SKAIČIAIS.

*9 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS ŠEŠIOLIKTAINĖS SISTEMOS SKAIČIUS: 1D5F, A9BC, FF.

* + 1. **Tema. Informacijos matavimo vienetai.**

*10 užduotis.* UŽRAŠYKITE VISUS JUMS ŽINOMUS INFORMACIJOS KIEKIO MATAVIMO VIENETUS.

*11 užduotis.* TURITE 16GB USB ATMINTUKĄ. PASKAIČIUOKITE KIEK TOKS ATMINTUKAS TURI MEGABAITŲ (MB), KILOBAITŲ (KB) IR BAITŲ (B). PASKAIČIUOKITE KIEK GALITE PATALPINTI Į TOKIĄ LAIKMENĄ BITŲ INFORMACIJOS.

*12 užduotis.* JŪSŲ INTERNETO PASLAUGŲ TIEKĖJAS PER MĖNESĮ UŽTIKRINA JUMS 8589934592 BITŲ DUOMENŲ KIEKĮ. KIEK NUOTRAUKŲ GALITE PERSIŲSTI KIEKVIENĄ MĖNESĮ, JEI VIENOS NUOTRAUKOS DYDIS BUS 5MB?

*13 užduotis.* JŪSŲ INTERNETO GREITAVEIKA 100MBPS. KIEK LAIKO UŽTRUKS 100MB FAILO SIUNTIMAS? LAIKOME, KAD INTERNETO GREITAVEIKA NESIKEIČIA IR YRA PASTOVI.

*14 užduotis.* KOKIA TURI BŪTI JŪSŲ INTERNETO GREITAVEIKA, KAD PER 5 MINUTES GALIMA BŪTŲ PERDUOTI 600MB INFORMACIJOS?

* 1. **Mokymosi rezultatas. Taikyti algoritmų ir logikos mokslo pagrindus** **programuojant.**
     1. **Tema. Logikos mokslo pagrindai.**

*1* *užduotis.*  FORMALIZUOKITE PATARLĘ: „DEGTINEI Į TROBĄ ĮEINANT, PROTAS IŠEINA LAUKAN“ TEIGINIŲ LOGIKOS PRIEMONĖMIS.

(užduotis iš Raškinis, A., Karoblis, G. knygos „Logikos užduočių pratybos”).

*2 užduotis*. FORMALIZUOKITE PATARLĘ: „DEGTINĖ IR ALUS GIMDO VARGUS“ TEIGINIŲ LOGIKOS PRIEMONĖMIS.

(užduotis iš Raškinis, A., Karoblis, G. knygos „Logikos užduočių pratybos”).

*3 užduotis.* UŽRAŠYKITE DE MORGANO TAISYKLĖS KONJUNKCIJAI SIMBOLINĘ IŠRAIŠKĄ.

Kaip ta išraiška turi skambėti skaitoma? Pateikite savo sugalvotą interpretacijos pavyzdį (užduotis iš Raškinis, A., Karoblis, G. knygos „Logikos užduočių pratybos”).

*4 užduotis*. UŽRAŠYKITE DE MORGANO TAISYKLĖS KONJUNKCIJAI SIMBOLINĘ IŠRAIŠKĄ.



Kaip ta išraiška turi skambėti skaitoma? Pateikite savo sugalvotą interpretacijos pavyzdį. (užduotis iš Raškinis, A., Karoblis, G. knygos „Logikos užduočių pratybos”.

* + 1. **Tema. Algoritmai ir algoritmavimas.**

*5 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS APSKAIČIUOJA SKAIČIŲ MASYVO SUMĄ IR VIDURKĮ.

Papildoma sąlyga: turi būti panaudotas iteratyvus algoritmas.

*6 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURANDA MAŽIAUSIĄ MASYVO SKAIČIŲ.

Papildoma sąlyga: turi būti panaudotas rekursyvus algoritmas.

*7 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SKAIČIUOJA 120 FIBONAČIO SKAIČIŲ.

Papildoma sąlyga: turi būti naudojamas dinaminis programavimas.

*8 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS NUSTATO AR RINKINYJE (SET) YRA ELEMENTŲ RINKINYS (SUBSET) LYGUS NURODYTAI SUMAI.

Papildoma sąlyga: turi būti naudojamas dinaminis programavimas.

*9 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURŪŠIUOJA SVEIKŲJŲ SKAIČIŲ MASYVĄ.

Papildoma sąlyga: turi būti panaudotas greito rūšiavimo algoritmas.

*10 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURŪŠIUOJA SVEIKŲJŲ SKAIČIŲ MASYVĄ.

Papildoma sąlyga: turi būti panaudotas burbulo algoritmas.

*11 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SIMBOLIŲ EILUTĘ ATSPAUSDINA ATVIRKŠČIAI.

Papildoma sąlyga: turi būti panaudota lifo duomenų struktūra (stack).

*12 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS RANDA MAŽIAUSIĄ SKAIČIŲ AIBĖJE.

Papildoma sąlyga: turi būti panaudota dvejetainio medžio (binary tree) struktūra.

2. **Mokymosi rezultatas. Taikyti programinio kodo dizaino modelius programuojant.**

**2.3.1. Tema. Dizaino šablonai.**

*1 užduotis*. UŽPILDYKITE PATEIKTĄ LENTELĘ, NURODYDAMI JUMS ŽINOMUS DIZAINO MODELIUS/ŠABLONUS (ANGL. *DESIGN PATTERNS*) IR PRISKIRDAMI VIENAI IŠ KATEGORIJŲ.

|  |  |  |
| --- | --- | --- |
| **Dizaino šablono rūšys** | | |
| Creational | Structural | Behavioral |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

*2 užduotis.* APRAŠYKITE PASIRINKTINAI TRIS DIZAINO ŠABLONUS.

Šablonai turi būti skirtingų rūšių. Pateikite pavyzdžių kur šablonas gali būti taikomas.

**2.3.2. Tema. Dizaino šablonų taikymas.**

*3 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ „ABSTRAKČIOS GAMYKLOS“ ŠABLONUI (ABSTRACT FACTORY PATTERN).

*4 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ „DEKORATORIAUS“ ŠABLONUI (DECORATOR PATTERN).

*5 užduotis*. NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ „STRATEGIJOS“ ŠABLONUI (STRATEGY PATTERN).

*6 užduotis.* PARINKITE DIZAINO ŠABLONĄ PATEIKTAM GYVENIMO PAVYZDŽIUI: GAMYKLA GAMINA TRIJŲ MODELIŲ AUTOMOBILIUS.

Kiekvienas automobilis turi savo unikalius parametrus. Kiekvienas automobilis gaminamas pagal savo brėžinį. Visi automobiliai gaminami vienoje gamykloje.

*7 užduotis.* PARINKITE DIZAINO ŠABLONĄ PATEIKTAM GYVENIMO PAVYZDŽIUI: TURIME PAŠTO SERVERĮ.

Serveris moka siųsti elektroninius laiškus. Laiškai būna skirti įvairiems adresatams. Priklausomai nuo adresato prie laiško yra pridėdama tam tikra tarnybinė informacija.

**2.4. Mokymosi rezultatas. Naudoti informacinių sistemų kūrimo principus ir metodus programinės įrangos projektavime ir programavime.**

**2.4.1. Tema. Daugiasluoksnė programų architektūra ir MVC struktūra.**

*1 užduotis*. APRAŠYKITE KLIENTO-SERVERIO (CLIENT SERVER) PROGRAMINĖS ĮRANGOS MODELĮ.

Pateikite pavyzdžių kur tokia architektūra yra naudojama.

*2 užduotis*. APRAŠYKITE MODELIO-VAIZDO-KONTROLERIO (MODEL VIEW CONTROLLER, MVC) PROGRAMINĖS ĮRANGOS MODELĮ.

Pateikite pavyzdžių kur tokia architektūra yra naudojama.

*3 užduotis*. NAUDOJANT MVC ARCHITEKTŪROS MODELĮ, SUKURKITE PAPRASTĄ DARBUOTOJŲ VALDYMO PROGRAMĄ.

Programoje turi būti panaudoti: modelis, vaizdas ir kontroleris. Darbuotojai turi savo unikalų numerį (id), vardą (name), amžių (age), darbo stažą (experience).

*4 užduotis*. NAUDOJANT PASIRINKTĄ KARKASĄ SUKURKITE PAPRASTĄ WEB VARTOTOJŲ REGISTRACIJOS PROGRAMĄ.

Registracijos formos laukai: vardas, pavardė, vartotojo vardas (slapyvardis), telefonas, el. Paštas, slaptažodis.

**2.4.2. Tema. Pagrindiniai programavimo principai.**

*5 užduotis*. APRAŠYKITE SOLID IR DRY PROGRAMAVIMO PRINCIPUS.

*6 užduotis*. APRAŠYKITE SEPARATION OF CONCERNS IR CODE REUSE PROGRAMAVIMO PRINCIPUS.

*7 užduotis*. NAUDOJANT FUNKCINIO PROGRAMAVIMO PRINCIPUS REALIZUOKITE GREITO RŪŠIAVIMO ALGORITMĄ.

Programa turi mokėti rūšiuoti sveikų skaičių sąrašą.

*8 užduotis*. NAUDOJANT FUNKCINIO PROGRAMAVIMO PRINCIPUS REALIZUOKITE SĄRAŠO GRUPAVIMO, ELEMENTŲ SKAIČIAVIMO IR RŪŠIAVIMO ALGORITMĄ.

Programa turi dirbti su vardų sąrašu.

1. **Kompetencija. Kurti tipinę programinę įrangą.**
2. 1. **Mokymosi rezultatas. Konfigūruoti pagrindinius įrankius, reikalingus projekto paleidimui, ir įdiegti projektui reikalingus paketus.**
      1. **Tema. NodeJS ir Node Package Manager (NPM).**

*1 užduotis*. APIBŪDINKITE „NODE PACKAGE MANAGER“ (NPM) VALDYMO SISTEMOS FUNKCIJAS NODE.JS APLINKOJE.

*2 užduotis*. KOKIUS VEIKSMUS ATLIEKA NPM INIT IR NPM INSTALL KOMANDOS SUKURIANT PROJEKTĄ NAUDOJANT NPM VALDIKLĮ?

*3 užduotis*. PARAŠYKITE KAM YRA SKIRTAS NPM VALDYMO SISTEMOS MODULIS (ANGL. “MODULE”) IR PAKETAS (ANGL. “PACKAGE”).

* + 1. **Tema.NodeJS aplinkos konfigūravimas ir valdymas.**

*4 užduotis*. NURODYTAS PAVYZDINIS KONFIGŪRACINIO PACKAGE.JSON FAILO TURINYS.

{

...

"dependencies": {

"fis": "3.2.1",

"foo": "^1.17.1",

"bar" : " 11.x",

"boo": "\*",

"thr": " ~1.2.3-beta.2",

"pow": "^0.0.x"

},

...

}

Naudodamiesi šiuo konfigūraciniu failu nurodykite kokias versijas galima naudoti paketams "foo" "bar", "boo", "thr". Kiekvienam paketui užrašykite tinkamas naudotinas versijas (pvz: paketui "fis" tik 3.2.1)

*5 užduotis*. KAS YRA ĮRAŠOMA Į PACKAGE-LOCK.JSON FAILĄ?

*6 užduotis*. PAPILDYKITE PACKAGE.JSON KONFIGURACINIO FAILO TURINĮ NPM KOMANDOS TRUMPINIU (ANGL. „SCRIPT“), KURIS PRADĖTŲ NODE.JS PROJEKTO FAILĄ *SERVER.JS.* KOMANDOS TRUMPINYS TURI BŪTI PALEIDŽIAMAS NAUDOJANT.

NPM RUN SERVER-START **ARBA** NPM RUN-SCRIPT SERVER-START

{

"name": "nodejs-project",

"version": "0.2.0",

"description": "test-description",

"main": "index.js",

"dependencies": {

"express": "^4.13.3"

},

"author": "Name Surname",

"license": "MIT"

}

*7 užduotis*. UŽRAŠYKITE KOMANDĄ, KURI ĮDIEGTŲ 0.1.2 VERSIJOS NPM PAKETĄ "FOO-BAR" Į PROJEKTO APLINKĄ BEI ŠĮ PAKEITIMĄ NURODYTŲ REIKALAUJAMŲ SUSIETŲ PAKETŲ KŪRĖJŲ SKILTYJE (ANGL. „DEVELOPER DEPENDENCIES“).

*8 užduotis.* PARSISIŲSKITE IR ĮSIDIEKITE NAUJAUSIĄ NODEJS DARBINĘ APLINKĄ IŠ OFICIALAUS TINKLAPIO.

ESLint, Babel ir Webpack įrankiai yra JavaScript moduliai (arba kitaip – įrankiai), jie turi būti diegiami iš NodeJS modulių repozitorijos (komandinės eilutės panaudojimo atveju), tad reikia įsideigti NodeJS aplinką, kurios pagalba bus diegiami minėti moduliai ir paleidžiamos komandinės eilutės norimam rezultatui pasiekti.

NodeJS aplinka turi pagalbinius įrankius JavaScript moduliams įdiegti ir automatiškai atnaujinti. Vienas jų – NPM (Node package manager). Pateiktų užduočių metu būtent naudosime NPM įrankį moduliams įdiegti ir juos valdyti.

Kad galėtume Windows aplinkoje patogiai naudotis NPM įrankiu, reikia papildomai įsidiegti Windows konsolės valdymo programą/emuliatorių. Yra nemažai alternatyvų (<https://www.slant.co/topics/1552/~best-terminal-emulators-for-windows>), tačiau vienas geriausių ir patogiausių sprendimų yra “Windows Power Shell” programa, kuri suteikia Windows konsolės valdymo galimybę, analogišką Linux terminalui. T.y. “Windows Power Shell” nereikalauja papildomų konfigūracijų ar kitokių papildomų žingsnių norint įsidiegti ir naudoti NPM modulius.

*9 užduotis*. IŠTESTUOKITE ĮDIEGTĄ NODEJS DARBINĘ APLINKĄ NAUDOJANTIS „ WINDOWS POWER SHELL“ KOMANDINĖS EILUTĖS VALDYMO ĮRANKIU, KVIEČIANT VERSIJOS SPAUSDINIMO KOMANDAS.

*10 užduotis*. IŠTESTUOKITE ĮDIEGTĄ NPM JAVASCRIPT MODULIŲ TVARKYKLĘ IŠKVIEČIANT NPM VERSIJOS SPAUSDINIMĄ KOMANDINĖS EILUTĖS VALDYMO ĮRANKYJE.

*11 užduotis*. PATIKRINKITE AR NODEJS TEISINGAI INTERPRETUOJA JAVASCRIPT KODĄ SUKURIANT NAUJĄ DARBINĮ FAILĄ „HELLO.JS“ SU TOKIU JAVASRIPT KODU: „CONSOLE.LOG(‘Hello’);“ IR PALEIDŽIANT NODEJS INTERPRETATORIŲ NURODANT SUKURTĄ HELLO.JS FAILĄ.

* 1. **Mokymosi rezultatas. Diegti ir konfigūruoti įrankius kodo transformavimui, pakavimui ir optimizavimui.**

**3.2.1. Tema. ESLint kodo tikrinimo įrankis.**

*1 užduotis.* ĮSIDIEKITE ESLINT MODULĮ BEI IR INICIJUOKITE JO STANDARTINĘ KONFIGŪRACIJĄ NAUDOJANTIS NPM ĮRANKIU IR TAIKANTIS Į PROJEKTO DARBINĘ APLINKĄ.

ESLint yra įrankis, kurio pagalba galime testuoti JavaScript kodą ir pagal iš anksto numatytas ar savo susikurtas taisykles – nustatyti konkrečias JavaScript kodo vietas (eilutę) kur yra neatitikimai ar galimos kodo klaidos.

ESLint kaip ir visi NPM moduliai gali būti diegiami visuotinai (Global) arba tik projekto darbiniame aplanke (dev), kur jau gali būti naudojamas ir IDE įrankių. Tačiau bet kuriuo atveju, NPM moduliai yra skirti tik darbui darbinėje aplinkoje. Į produkcinę aplinką turi patekti tik ištestuotas kodas be jokių papildomų NPM modulių.

*2 užduotis.* SUKONFIGŪRUOKITE ESLINT DARBUI SU INTERNETINĖMIS NARŠYKLĖMIS, NODEJS IR ES6 APLINKOMIS BEI PRITAIKYKITE PROGRAMAVIMO STILIAUS TAISYKLES: INDENTACIJA – 2 LYGIAI, KABLIATAŠKIS – ISPĖJIMAS, JAVASCRIPT „CONSOLE.LOG“ – LEIDŽIAMA.

*3 užduotis.* IŠTESTUOKITE JAVASCRIPT KODĄ SU ESLINT BEI IŠTAISYKITE PASTEBĖTAS KLAIDAS JEIGU JOS YRA APTINKAMOS ESLINT ĮRANKIO.

UŽDUOTIES JAVASCRIPT KODAS:

function suma(x,y,t) {

    return t + ( x + z );

}

suma(20,30, 'Suma: ')

*4 užduotis.* PARUOŠKITE ESLINT DARBUI SU IDE – UŽTIKRINKITE, KAD EGZISTUOJA KONFIGŪRACINIAI FAILAI (PRIKLAUSOMAI NUO PASIRINKTO IDE) IR ESLINT EGZISTUOJA PROJEKTO DARBINĖJE APLINKOJE.

* + 1. **Tema. Babel įrankio diegimas ir naudojimas.**

*5 užduotis.* ĮDIEKITE IR PARUOŠKITE DARBUI BABEL JS MODULĮ NAUDOJANTIS NPM ĮRANKIU (ĮDIEKITE BABEL, PARUOŠKITE DARBINĖS APLINKOS APLANKUS).

Babel yra JavaScript kodo kompiliatorius/transformatorius. Jo paskirtis – leisti JavaScript programuotojams naudotis naujausiais programavimo kalbų pasiekimais ir naujausiomis kodo bibliotekomis pritaikant juos diegimui į įvairias terpes, kur internetinės naršyklės gali būti pasenusios ir nepalaikyti naujausių programinio kodo pasiekimų. Tai padaroma interpretuojant parašytą modernų JavaScript (pvz. ES6) kodą ir verčiant jį į standartinį JavaScript kodą be jokių papildinių.

Babel užtikrina, kad transformuotas JavaScript kodas veiks visose nurodytose naršyklėse ir jų versijose. Žinoma, reikia nepamiršti, kad tam tikri programavimo kalbos niuansai gali lemti, kad senesnės internetinės naršyklės vis vien nesugebės realizuoti parašyto ir transformuoto kodo.

Daugiau informacijos galima rasti oficialiame Babel puslapyje: <https://babeljs.io/>

*6 užduotis.* IŠTESTUOKITE ĮDIEGTĄ BABEL MODULĮ SU PATEIKTU JAVASCRIPT ES6 KODU.

Paleisti Babel komandą duotam JavaScript kodui ir patikrinti, ar susikuria atitinkami failai darbinėje aplinkoje, be transformacijų atlikimo

export function suma(x, y) {

return x + y;

}

var nums = [1, 2, 3];

var doubleNums = nums.map((e) => e \* 2);

var sum = nums.map((e) => suma(e + 1, e \* 10));

console.log(doubleNums);

console.log(sum);

*7 užduotis*. ĮDIEKITE BABEL TRANSFORMACIJŲ NUSTATYMUS VADOVAUJANTIS OFICIALIA DOKUMENTACIJA (<https://babeljs.io/en/setup/#installation>).

Pakartotinai paleisti babel modulį anksčiau nurodytam JavaScript kodui ir gauti transformuotą rezultatą

"use strict";

Object.defineProperty(exports, "\_\_esModule", {

value: true

});

exports.suma = suma;

function suma(x, y) {

return x + y;

}

var nums = [1, 2, 3];

var doubleNums = nums.map(function (e) {

return e \* 2;

});

var sum = nums.map(function (e) {

return suma(e + 1, e \* 10);

});

console.log(doubleNums);

console.log(sum);

*8 užduotis.* SUKONFIGŪRUOKITE BABEL ESLINT PALAIKYMĄ IR PARUOŠKITE WEBPACK NAUDOJIMUI. IŠTESTUOKITE ES6 KODĄ SU ESLINT PRIEŠ BABEL TRANSFORMACIJĄ.

* + 1. **Tema. Webpack įrankio diegimas ir naudojimas.**

*9 užduotis.* NAUDOJANTIS NPM ĮRANKIU ĮDIEKITE WEBPACK BEI CLI IR SERVER KOMPONENTUS IR PARUOŠKITE DARBUI DARBINĖJE APLINKOJE.

Webpack, kitaip žinomas kaip WebpackJS, yra skirtas apjungti skirtingus JavaScript projekto failus, kurie sudaro JavaScript projektą. Tai gali būti pakopiniai stiliai (CSS), HTML, JavaScript kodas ir kita.

Naudojantis Webpack galimas JavaScript projekto struktūrizavimas (išskirstymas į smulkesnius failus, kurie atitinka tam tikrus JavaScript projekto komponentus) ir projekto dalių atskyrimas. Webpack garantuoja, kad atskiros dalys bus sudėtos į vieną, programinis kodas (jei toks yra) bus patikrintas ir kitais pagalbiniais įrankiais (pvz Babel ar ESLint) bus apdorotas.

Daugiau apie Webpack galima rasti oficialiame puslapyje: <https://webpack.js.org/>

Taip pat, sukurti naujus failus su tokiu turiniu:

**src/hello.js:**

export const hello = subject => console.log(`Hello ${subject}!`);

**src/index.js:**

import { hello } from "./hello";

hello("JS Dev!!!");

**dist/index.html:**

<!DOCTYPE html>

<html>

<head>

<title>Hello Webpack</title>

</head>

<body>

<div>

<h1>Hello Webpack</h1>

</div>

<script src="./bundle.js"></script>

</body>

</html>

*10 užduotis.* SUKONFIGŪRUOKITE WEBPACK TAIP, KAD PRADĖTŲ NAUDOTI ESLINT IR BABEL MODULIUS.

*11 užduotis.* ATLIKUS WEBPACK KONFIGŪRAVIMO DARBUS, PALEISKITE WEBPACK SERVER PROJEKTĄ.

* 1. **Mokymosi rezultatas. Analizuoti ir derinti vykdomą kodą, identifikuoti problemas ir jas taisyti.**
     1. **Tema. Vykdomo kodo analizės įrankiai.**

*1 užduotis*. APRAŠYKITE CHROME DEVTOOLS PANAUDOJIMO GALIMYBES.

*2 užduotis*. IŠVARDINKITE CHROME DEVTOOLS SKYDELIUS IR TRUMPAI APRAŠYKITE JŲ PASKIRTĮ.

*3 užduotis*: IŠVARDINKITE KOKIUS VEIKSMUS GALIMA ATLIKTI ANALIZUOJANT HTML DOM MEDĮ PASINAUDOJANT DEVTOOLS ELEMENTŲ SKYDELIU (ANGL. „ELEMENTS“). IŠVARDINKITE BENT 4 PANAUDOJIMO ATVEJUS.

*4 užduotis*. APRAŠYKITE KAIP GALIME REDAGUOTI CSS KODĄ PASINAUDOJANT CHROME DEVTOOLS? PATEIKITE REDAGAVIMO GALIMYBIŲ PAVYZDŽIŲ. (2-3 PVZ.).

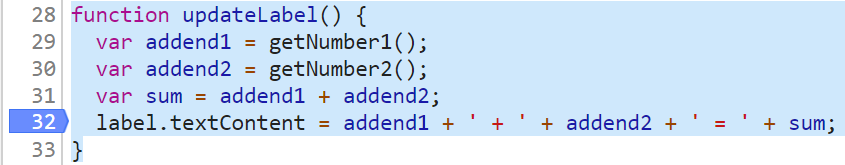
*5 užduotis*. KAM REIKALINGAS NAŠUMO (ANGL. „PERFORMANCE“) SKYDELYJE ESANTIS GREITAVEIKOS ANALIZĖS PROFILIAVIMO ĮRANKIS (ANGL. „PROFILING“)?

*6 užduotis*. APRAŠYKITE ŽINGSNIUS KAIP NAUDODAMIESI ANALIZĖS ĮRANKIO AUDITO SKYDELIU (ANGL. „AUDITS“) IŠANALIZUOTUMĖTE PUSLAPIO NAŠUMĄ.

**3.3.2. Tema*.* Programos derinimas.**

*7 užduotis*. APRAŠYKITE IR PALYGINKITE KODO DERINIMO BŪDUS: CONSOLE.LOG() NAUDOJIMĄ IR STABDOS TAŠKUS (ANGL. BREAKPOINTS).

*8 užduotis*. KAS ĮVYKS ĮVYKDŽIUS JAVASCRIPT KODĄ, JEI 32 EILUTĖJE PADĖSIME STABDOS TAŠKĄ (ANGL. BREAKPOINT)?



1 paveikslėlis. Kodo dalis iš <https://googlechrome.github.io/devtools-samples/debug-js/get-started>

*9 užduotis.* PATEIKITE STABDOS TAŠKŲ (ANGL. BREAKPOINTS) TIPUS IR JUOS APRAŠYKITE.

* 1. **Mokymosi rezultatas. Kurti JavaScript programinį kodą ir naudoti jo sintaksę.**

**3.4.1. Tema. Sintaksė, gramatika, duomenų tipai.**

*1 užduotis.* IŠVESKITE PRANEŠIMĄ: „MAN PATINKA WEB TECHNOLOGIJOS“ NARŠYKLĖS DIALOGO LANGE IR KONSOLĖJE.

*2 užduotis.* PANAUDODAMI JS PROMPT DIALOGO LANGĄ NARŠYKLĖJE PAKLAUSKITE VARTOTOJO VARDO IR JĮ IŠVESKITE NARŠYKLĖS KONSOLĖJE.

*3 užduotis.*  PANAUDODAMI TYPEOF FUNKCIJĄ NUSTATYKITE IR IŠVESKITE NARŠYKLĖS KONSOLĖJE IŠVARDINŲ DUOMENŲ TIPUS:

1. Studentas
2. 12
3. true

**Pavyzdys:**

typeof("JavaScript puiki programavimo kalba")

// Konsolė išves: string

*4 užduotis.* SUKURKITE KINTAMUOSIUS: KNYGOS PAVADINIMAS, AUTORIUS, PUSLAPIŲ SKAIČIUS, KIEKIS, KAINA.

Sukurtiems kintamiesiems priskirkite tinkamas reikšmes pagal kintamojo prasmę.

Kintamųjų reikšmes išveskite naršyklės konsolėje, paga žemiau pateiktą pavyzdį.

**Pavyzdys:**

Knygos pavadinimas: Logikos pagrindai

Autorius: Vardenis Pavardenis

Puslapių skaičius: 500

Kiekis: 500

Kaina: 20.56

*5 užduotis. 4 užduotyje* SUKURTŲ KINTAMŲJŲ REIKŠMES PATALPINKITE Į MASYVĄ KNYGA IR IŠVESKITE NARŠYKLĖS KONSOLĖJE.

*6 užduotis.* SUKURKITE KNYGOS MASYVĄ, KURIS TALPINA INFORMACIJĄ APIE 5 KNYGAS.

Knygos duomenų struktūrą turi būti tokia pati, kaip ankstesnėje užduotyje. Naršyklės konsolėje išveskite trečią ir penktą knygas.

*7 užduotis.* SUKURKITE KNYGOS OBJEKTĄ, PAGAL *6 užduotyje* PATEIKTĄ KNYGOS DUOMENŲ STRUKTŪRĄ. NARŠYKLĖS KONSOLĖJE IŠVESKITE KNYGOS PAVADINIMĄ IR PUSLAPIŲ SKAIČIŲ.

*8 užduotis.*SUKURKITE MASYVĄ, KURIS TALPINA 5 KNYGŲ OBJEKTUS PAGAL *6* *užduotyje* SUKURTĄ OBJEKTO STRUKTŪRĄ.

Naršyklės konsolėje išveskite visų knygų pavadinimus.

*9 užduotis.* SUKURKITE PROGRAMĄ, KURI, NAUDODAMA PROMPT DIALOGO LANGĄ, SURINKTĄ INFORMACIJĄ (KNYGOS PAVADINIMAS, KIEKIS, NUOLAIDOS DYDIS PROCENTAIS) APSKAIČIUOJA KNYGOS PARDAVIMO KAINĄ IR IŠVEDA INFORMACIJĄ NARŠYKLĖS KONSOLĖJE.

**3.4.2. Tema. Operatoriai, salygos sakiniai.**

*10 užduotis.* SUKURKITE PROGRAMĄ, KURI NAUDODAMA PALYGINIMO OPERATORIUS IŠVEDA *5 užduotyje* SUKURTAS KNYGAS, KURIŲ PUSLAPIŲ SKAIČIUS YRA MAŽESNIS UŽ 200, O PAVADINIMĄ SUDARO 15 IR DAUGIAU SIMBOLIŲ.

*11 užduotis.* SUKURKITE PROGRAMĄ, KURI NAUDODAMA LOGINIUS OPERATORIUS IŠVEDA *5 užduotyje* KNYGAS, KURIŲ PUSLAPIŲ SKAIČIUS YRA MAŽESNIS UŽ 200 ARBA DIDESNIS UŽ 500.

*12 užduotis.*SUKURKITE PROGRAMĄ, KURI KINTAMAJAM VIRŠELIO SPALVA PRISKIRTĄ ŠEŠIOLIKTAINĘ SPALVOS REIKŠMĘ KONVERTUOJA Į RGB (SPALVOS REIKŠMĘ).

Naršyklės konsolėje reikia išvesti raudonos, žalios ir mėlynos spalvos skaičių pagal pateiktą pavyzdį.

**Pavyzdys:**

var spalvosKodas = '#ffaadd';

Raudona: 255

Žalia: 170

Mėlyna: 221

*13 užduotis.*SUKURKITE SKAIČIUOTUVO PROGRAMĄ, KURI PER FUNKCIJĄ PROMPT IŠ VARTOTOJO PRIIMA SKAIČIUS, OPERATORIUS(+, -, /, \*, %) IR IŠVEDA REZULTATĄ.

Jeigu vartotojas įrašo netinkamo duomenų tipo reikšmę, turi būti išvedama klaida.

**3.4.3. Tema. Ciklai.**

*14 užduotis.*NAUDODAMI FOR IR WHILE CIKLUS IŠVESKITE *6 užduotyje* SUKURTĄ KNYGŲ MASYVĄ NARŠYKLĖS KONSOLĖJE.

*15 užduotis.* NAUDODAMI FOR CIKLĄ Į *6 užduotyje* SUKURTĄ KNYGŲ MASYVĄ PRIDĖKITE TRIS NAUJAS KNYGAS, KURIŲ PAVADINIMUS VARTOTOJAS NURODYS PROMPT DIALOGO LANGO PAGALBA NARŠYKLĖS KONSOLĖJE.

*16 užduotis.*NAUDODAMI WHILE ARBA FOR CIKLĄ *6 užduotyje* SUKURTAME MASYVE KNYGŲ KAINAS PADIDINKITE 10 % IR PAŠALINKITE PASKUTINĘ KNYGĄ IŠ MASYVO.

Knygų sąrašą išveskite naršyklėje.

**3.4.4. Tema. Objektai.**

*17 užduotis.*SUKURKITE KNYGŲ KATALOGĄ, KURIAME BŪTŲ 3 KATEGORIJOS IR 5 KNYGOS KIEKVIENOJE KATEGORIJOJE.

Visi katalogo duomenys turi būti aprašyti, kaip objektai. Kategorijos objektas, turi turėti kategorijai priklausančius objektus viduje.

Knyga turi turėti šias savybes: ISBN, leidimo metai, pavadinimas, puslapių skaičius, kiekis, kaina.

*18 užduotis.*NARŠYKLĖS KONSOLĖJE IŠVESKITE *17 užduotyje* SUKURTĄ KNYGŲ KATALOGĄ PAGAL ŽEMIAU PATEIKTĄ PAVYZDĮ.

**Pavyzdys:**

Kategorijos pavadinimas: Informacinės technologijos

ISBN: IT-12345

Leidimo metais: 2019 (PASTABA: jeigu knyga nauja, šalia leidimo metų turi būti išvestas pranešimas, kad knyga nauja)

Pavadinimas: “Programavimas linksmai”

Puslapių skaičius: 250

Kiekis: 12

Kaina: 25 EUR

-----------------------------------------------------------------------------------------------------------------------------------------

ISBN: IT-12346

Leidimo metais: 2018 (PASTABA: jeigu knyga nauja, šalia leidimo metų turi būti išvestas pranešimas, kad knyga nauja)

Pavadinimas: “WEB technologijos”

Puslapių skaičius: 350

Kiekis: 50

Kaina: 30 EUR

-----------------------------------------------------------------------------------------------------------------------------------------

..........

Kategorijos pavadinimas: Fantastika

..........

*19 užduotis.* SUKURKITE KNYGŲ PIRKĖJO KORTELĖS OBJEKTĄ PAGAL STRUKTŪRĄ ŽEMIAU PATEIKTAME PAVYZDYJE.

Savybių reikšmes nustatykite objekto išorėje (set metodas).

**Pavyzdys:**

Kortelės ID: TR-CS-5689

Vardas: Vardenis

Pavardė: Pavardenis

Tipas: VIP klientas

Paskutinės įsigytos knygos: (masyvas)

Nr. 1Įdomioji istorija

Nr. 2Nerėgėta lietuva

Nr. 3Gyvenimas

*20 užduotis.* NARŠYKLĖS KONSOLĖJE IŠVESKITE *19 užduotyje* SUKURTO OBJEKTO SAVYBES (VARDAS, PAVARDĖ, PASKUTINIŲ ĮSIGYTŲ KNYGŲ KIEKĮ).

*21 užduotis. 19 užduotyje* SUKURTO OBJEKTO DUOMENIS PAPILDYKITE. PRIDĖKITE DAR VIENĄ ĮSIGYTĄ KNYGĄ NEREDAGUOJANT PAČIO OBJEKTO (SET METODAS).

*22 užduotis.*PAGAL *19 užduotyje* PATEIKTĄ OBJEKTO STRUKTŪRĄ SUKURKITE KONSTRUKTORIŲ OBJEKTŲ KŪRIMUI.

*23 užduotis.*PANAUDODAMI *22 užduotyje* SUKURTĄ KONSTRUKTORIŲ SUKURKITE 3 KNYGŲ PIRKĖJUS.

Pirkėjų informaciją išveskite naršyklės konsolėje.

*24 užduotis.*SUKURKITE PIRKĖJO OBJEKTO PROTOTIPĄ SU METODU SPAUSDINTI.

Panaudokite metodą informacijai apie pirkėjus išvedimui naršyklės konsolėje.

*25 užduotis.* IŠPLĖSKITE *22 užduotyje* SUKURTĄ PIRKĖJO KONSTRUKTORIŲ SUKURDAMI NAUJĄ PAVADINIMU KLUBO NARYS.

Naujam konstruktoriui pridėkite papildomas savybes: lojalumo taškai ir reitingas.

*26 užduotis.*SUKURKITE 3 KNYGŲ PIRKĖJUS PAGAL IŠPLĖSTA KONSTRUKTORIŲ *25 užduotyje.*

Sukurtus objektus išsaugokite lokalioje saugykloje. Duomenis apie pirkėjus nuskaitykite iš saugyklos ir išveskite iš kompiuterio atminties – naršyklės konsolėje.

*27 užduotis. 19 užduotyje* SUKURTAM PIRKĖJO KORTELĖS OBJEKTUI ATLIKITE PAKEITIMUS UŽDRAUDŽIANT Į ĮSIGYTŲ KNYGŲ MASYVE PRIDĖTI NAUJĄ KNYGĄ.

*28 užduotis.*NEKEISDAMI ITERATORIAUS (KINTAMOJO) PAVADINIMO CIKLUOSE PATAISYKITE KODĄ, KAD BŪTŲ IŠVESTOS VISOS KATEGORIJOS IR JOMS PRIKLAUSANČIOS KNYGOS.

var knygos = ['Fantastika', 'IT', 'Technologijos'];

knygos['Fantastika']= ['Misijos į marsą istorija ', 'Ateiviai', 'Vanduo menulyje'];

knygos['IT']= ['Programuok linskmai', 'Naujos WEB technologijos'];

knygos['Technologijos'] = ['Elektromobiliai', 'Nauji kosminiai laiavai'];

for(var i =0; i<knygos.length; i++){

var kategorija =knygos[i];

console.log(kategorija);

for(var i=0; i<knygos[kategorija].length; i++){

console.log(knygos[kategorija][i])

}

}

*29 užduotis.*PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD BŪTŲ IŠVESTAS TEISINGAS REZULTATAS.

const a = [];

for (var i = 0; i < 10; i += 1) {

a[i] = () => i;

}

a[3]()

a[7]()

*30 užduotis.* PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS.

const knygos = ['Fantastika', 'IT', 'Technologijos'];

knygos['Fantastika']= ['Misijos į marsą istorija ', 'Ateiviai', 'Vanduo menulyje'];

knygos['IT']= ['Programuok linskmai', 'Naujos WEB technologijos'];

knygos['Technologijos'] = ['Elektromobiliai', 'Nauji kosminiai laiavai'];

knygos = ['Romanai','Grožinė literatūra','Tikslieji mokslai'];

*31 užduotis.**17 užduotyje* SUKURTO KNYGŲ KATALOGO IŠVEDIMUI SUKURKITE AROW FUNKCIJĄ, KURI PRIE KNYGOS KAINOS PRIDEDA PVM MOKESTĮ (21%) IR NARŠYKLĖS KONSOLĖJE IŠVEDA KAINĄ SU PVM.

*32 užduotis.**17 užduotyje* SUKURTO KNYGŲ KATALOGĄ SUKURKITE KAIP MODULĮ VIENAME FAILE, O IŠVEDIMĄ REALIZUOKITE KITAME FAILE JĮ IMPORTUODAMI.

*33 užduotis.*ŽEMIAU PATEIKTAM SKAITYTOJŲ MASYVUI PANAUDOKITE DESTRUKTORIŲ. IŠVESKITE SKAITYTOJŲ VARDĄ IR KUO DOMISI.

const skaitytojai = [

{

vardas: 'Vardenis Pavardenis',

papildomaInformacija: {

domisi: 'Aviacija',

mokosi: 'Kauno informaicniu technologiju mokykloje',

gyvena: 'Kaune'

},

amzius: 35

},

{

vardas: 'Jonas Jonaitis',

papildomaInformacija: {

domisi: 'Programavimu',

mokosi: 'Kauno informaicniu technologiju mokykloje',

gyvena: 'Kaune'

},

amzius: 25

}

];

**3.4.5. Tema. ES6 standarto ypatumai.**

*34 užduotis.* IŠVESKITE INFORMACIJĄ PAGAL DETALESNĮ APRAŠYMĄ APIE KNYGAS (17 *užduotyje* aprašytas katalogas) PAGAL ŽEMIAU NURODYTĄ PAVYZDĮ, NAUDODAMI ES6 TEMPLATE LITERALS.

Reikšmes tarp riestinių skliaustų pakeiskite duomenimis iš katalogo.

**Pavyzdys**

Labai gera knyga {knygos pavadinimas}, kurią paraštė nuotabus autorius. Džiaugiamės savo klientams galėdami pasiūlyti net {kiekis} knygų. Norėdami įsigyti šią knygą mūsų knygynuose, ieškokite skyriaus pavadinimu {kategorija}.

Iki susitikimo mūsų knygynuose.

* 1. **Mokymosi rezultatas. Vykdyti programinio kodo versijavimą naudojant versijavimo įrankius.**

**3.5.1. Tema. Išeities kodo saugyklos.**

*1 užduotis*. UŽRAŠYKITE TRIS KODO VERSIJŲ KONTROLĖS SISTEMAS, APRAŠYKITE PAGRINDINES VERSIJAVIMO SISTEMŲ FUNKCIJAS.

Papildomai parašykite pavyzdžių kaip versijavimo įrankiai gali būti naudojamos programuotojo praktinėje veikloje.

*2 užduotis.* PARSISIŲSKITE, ĮSIDIEKITE GIT KODO VERSIJAVIMO ĮRANKĮ.

*3 užduotis.* SUKONFIGŪRUOKITE GIT SAUGYKLĄ: NUSTATYKITE GIT VARTOTOJO VARDĄ, ELEKTRONINĮ PAŠTĄ, NAUDOJAMĄ TEKSTINĮ REDAKTORIŲ.

*4 užduotis.* PAPILDYKITE SAVO GIT KONFIGŪRACIJĄ (žiūrėti *3 užduotį*) PARAMETRAIS: COMMIT.TEMPLATE, CORE.PAGER, CORE.EXCLUDEFILE, HELP.AUTOCORRECT, COLOR.\*

*5 užduotis*. APRAŠYKITE BAZINES GIT KOMANDAS: GIT INIT, GIT CLONE, GIT STATUS.

*6 užduotis*. APRAŠYKITE BAZINES GIT KOMANDAS: GIT ADD, GIT COMMIT, GIT RESET, GIT CHECKOUT.

*7 užduotis.* PARAŠYKITE GIT KOMANDAS, KURIOS LEIS SUKURTI NAUJAS PROJEKTO ŠAKAS: TESTING, NEW-FEATURE.

Papildoma sąlyga: turi būti panaudotos Git komandos git branch ir git checkout.

*8 užduotis.* PARAŠYKITE GIT KOMANDAS, KURIOS LEIS APJUNGTI *7 užduotyje* SUKURTAS ŠAKAS SU PAGRINDINE SAUGYKLOS ŠAKA MASTER.

Papildoma sąlyga: po apjungimo papildomos šakos turi būti ištrintos.

**3.5.2. Tema. Programinio kodo versijavimo vykdymas.**

*9 užduotis.* NUKOPIJUOKITE KODO PAVYZDĮ IŠ NUOTOLINĖS GITHUB KODO SAUGYKLOS: <https://github.com/PacktPublishing/.NET-Design-Patterns>.

Lokaliai atlikite kelis pakeitimus šioje kodo saugykloje ir juos įrašykite į saugyklą.

Papildoma sąlyga: turi būti panaudotos komandos git clone, git add, git commit.

*10 užduotis*. SUKURKITE LOKALIĄ GIT KODO SAUGYKLĄ APLANKE VARDU CALCULATOR.

Sukurkite naują .NET projektą ir realizuokite paprasto komandinės eilutės skaičiuotuvo funkcionalumą (+, -, \*, /). Kiekvienas svarbesnis kodo pakeitimas turi būti įrašomas į kodo saugyklą su prasmingu komentaru.

Papildoma sąlyga: turi būti panaudotos komandos git init, git add, git commit, git status, git log.

*11 užduotis*. SAVO SKAIČIUOTUVUI (*10 užduotis*) PADARYKITE KODO PATAISYMUS IR JUOS EKSPORTUOKITE SU GIT KOMANDĄ GIT FORMAT-PATCH.

*12 užduotis*. GITHUB KODO TALPINIMO PLATFORMOJE YRA PATALPINTAS REACT KODAS.

Jo repozitorija yra čia: <https://github.com/facebook/react>. Naudojant Git įrankį išanalizuokite kiek kodo eilučių buvo parašyta šiame projekte? Kiek programuotojų dirba prie pateikto projekto?

* 1. **Mokymosi rezultatas. Naudoti JavaScript funkcijas.** 
     1. **Tema. Funkcijos.**

*1 užduotis.*PATOBULINKITE 3.4. KOMPETENCIJOS *17 užduotyje* SUKURTO KNYGŲ KATALOGO IŠVEDIMĄ SUKURDAMI FUKCIJĄ, KURI SUSKAIČIUOJA KATEGORIJOJE ESANČIŲ KNYGŲ SKAIČIŲ.

Išvedimo rezultate, knygų skaičius kategorijoje turi būti šalia kategorijos pavadinimo. Taip, kaip nurodyta žemiau pateiktame pavyzdyje.

**Pavyzdys:**

Kategorijos pavadinimas: Informacinės technologijos **(knygų skaičius: 5)**

ISBN: IT-12345

Leidimo metais: 2019 (PASTABA: jeigu knyga nauja, šalia leidimo metų turi būti išvestas pranešimas, kad knyga nauja)

Pavadinimas: “Programavimas linksmai”

Puslapių skaičius: 250

Kiekis: 12

Kaina: 25 EUR

-----------------------------------------------------------------------------------------------------------------------------------------

ISBN: IT-12346

Leidimo metais: 2018 (PASTABA: jeigu knyga nauja, šalia leidimo metų turi būti išvestas pranešimas, kad knyga nauja)

Pavadinimas: “WEB technologijos”

Puslapių skaičius: 350

Kiekis: 50

Kaina: 30 EUR

-----------------------------------------------------------------------------------------------------------------------------------------

..........

Kategorijos pavadinimas: Fantastika **(knygų skaičius: 3)**

..........

*2 užduotis.* SUKURKITE FUNKCIJĄ, KURIAI NURODŽIUS DUOMENŲ ŠALTINĮ IR KATEGORIJŲ KIEKĮ NARŠYKLĖS KONSOLĖJE BŪTŲ IŠVESTAS TIK NURODYTŲ KATEGORIJŲ KIEKIS IR JOMS PRIKLAUSANČIŲ KNYGŲ SĄRAŠAS.

*3 užduotis.*PERRAŠYKITE 3.4. KOMPETENCIJOS *37 užduotyje* SUKURTĄ FUNKCIJĄ Į ARROW FUNKCIJĄ.

*4 užduotis.* PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS (NEKEIČIANT PASKUTINĖS KODO EILUTĖS).

function pasisveikinti () {

const pirkejas = 'Vardenis Pavardenis'

console.log(`Labas:'${pirkejas}`);

}

console.log(PIRKEJAS)

**3.6.2. Tema. Standartinės JavaScript funkcijos.**

*5 užduotis.*PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD IŠKVIETUS OBJEKTO SPAUSDINK METODĄ RODYTIX BŪTŲ IŠVESTA x REIKŠMĖ (OBJEKTO KOREGUOTI NEGALIMA).

var spausdink = {

x: 54,

rodytiX: function() {

return this.x;

}

}

var xReiksme = spausdink.rodytiX

console.log(xReiksme());

*6 užduotis.*SUKURKITE ANONIMINĘ FUNKCIJĄ, KURI KONVERTUOJA GAUTĄ MASYVĄ Į EILUTĘ (STRING).

*7 užduotis.* SUKURKITE FUNKCIJĄ, KURI IŠVEDA 3.4. KOMPETENCIJOS *17 užduotyje* SUKURTĄ KNYGŲ KATALOGĄ.

Funkcija turi būti vykdoma automatiškai (Immediately Invoked Function Expression) papildomai nekviečiant naršyklėje įvykdžius .js failą.

*8 užduotis.*ŽEMIAU PATEIKTAME MASYVE PATEIKTI KNYGŲ PARDAVIMAI PAGAL METUS. PANAUDODAMI MINIMALIOS PAIEŠKOS FUNKCIJĄ, SURASKITE MAŽIAUSIĄ PARDAVIMŲ SUMĄ 2019 M.

**Pavyzdys**

const pardavimai = ['2017', '2018', '2019','2020'];

pardavimai['2017'] = [1500,12005,2893, 35000,17000,1223898.12,56784,2455,124588,56597,125658,124788];

pardavimai['2018'] = [200,152005,2563, 35600,15000,125698.12,56856,23455,124788,5647,12548,125688];

pardavimai['2019'] = [5600,18005,2233, 45000,15000,127898.12,56544,23475,1255698,5497,12478,122588];

pardavimai['2020'] = [5600,19005,2233, 65000,14230,127898.12,5647,2347,12478,5627,123258,12474588];

*9 užduotis.*PAGAL *8 užduotyje* APRAŠYTĄ PARDAVIMŲ MASYVĄ, PANAUDODAMI MAKSIMALIOS PAIEŠKOS FUNKCIJĄ, SURASKITE 2018 M. DIDŽIAUSIĄ PARDAVIMŲ SUMĄ.

*10 užduotis.* PAGAL *8 užduotyje* APRAŠYTĄ PARDAVIMŲ MASYVĄ, PANAUDODAMI APVALINIMO FUNKCIJĄ, SUAPVALINKITE 2018 M. PARDAVIMŲ SUMAS.

*11 užduotis.*PARAŠYKITE PROGRAMĄ, KURI PAŠALINTŲ SKAIČIUS PO KABLELIO IŠ 3.4. KOMPETENCIJOS *17 užduotyje* APRAŠYTO KATALOGO KNYGŲ KAINŲ (NEAPVALINANT) PANAUDOJUS MATEMATINĘ FUNKCIJĄ. KNYGAS IŠVESKITE NARŠYKLĖS KONSOLĖJE.

*12 užduotis.*PRAPLĖSKITE MATH OBJEKTĄ SUKURDAMI SAVO METODĄ.

Jūsų sukurtas metodas turi grąžinti didžiausią daliklį iš pateiktų metodui argumentų sąrašo. Rezultatą išvesti naršyklės konsolėje.

*13 užduotis.*IŠVESKITE 3.4 KOMPETENCIJOS. *17 užduotyje* SUKURTO KNYGŲ KATALOGO OBJEKTŲ SAVYBES PANAUDODAMI OBJEKTŲ FUNKCIJĄ, KURI GRĄŽINA OBJEKTO SAVYBES. GRĄŽINTOS SAVYBĖS TURI BŪTI PATALPINTOS MASYVE IR IŠVESTOS NARŠYKLĖS KONSOLĖJE.

*14 užduotis.* SUKURKITE NAUJĄ – TUŠČIĄ OBJEKTĄ PAVADINIMU NAUJAS KATALOGAS. NUKOPIJUOKITE 3.4. KOMPETENCIJOS *17 užduotyje* ESANČIŲ OBJEKTŲ SAVYBES SU REIKŠMĖMIS IR PRISKIRKIT NAUJAI SUKURTAM OBJEKTUI.

Tikslui pasiekti panaudokite tam skirtą objektų funkciją. Naujo objekto duomenis išveskite naršyklės konsolėje.

**3.6.3. Tema. Pagrindinės masyvų funkcijos.**

*15 užduotis.*PANAUDODAMI MASYVŲ FUNKCIJĄ FIND() IŠ *8 užduotyje* SUKURTO PARDAVIMŲ MASYVO, SURASKITE 2017M. PARDAVIMO SUMĄ, KURI DIDESNĖ UŽ 15000.

*16 užduotis.*IŠ ŽEMIAU PATEIKTO MASYVO PANAUDODAMI MASYVŲ FUNKCIJĄ FILTER, SURASKITE KNYGAS, KURIŲ PAVADINIMO ILGIS YRA DIDESNIS UŽ 15 SIMBOLIŲ.

Surastas reikšmes išveskite naršyklės konsolėje.

**Pavyzdys**

const knyguPavadinimai = ['Programavimas linksmai', 'C++', 'SASS', 'PHP', 'JavaScript'];

*17 užduotis.*PANAUDODAMI MASYVŲ FUNKCIJĄ SORT() SURIKIUOKITE KNYGAS IŠ *18 užduotyje* SUKURTO KNYGŲ KATALOGO PAGAL PAVADINIMĄ ABĖCELĖS TVARKA.

Surikiuotas knygas išveskite naršyklės konsolėje.

*18 užduotis.*PANAUDODAMI MASYVŲ FUNKCIJĄ REDUCE() APSKAIČIUOKITE *8* *užduotyje* SUKURTO MASYVO 2019 M. APRAŠYTŲ PARDAVIMŲ BENDRĄ SUMĄ. REZULTATĄ IŠVESKITE NARŠYKLĖS KONSOLĖJE.

*19 užduotis.* PANAUDODAMI MASYVŲ FUNKCIJĄ MAP() APSKAIČIUOKITE *8 užduotyje* SUKURTO MASYVO 2019 M. SUMAS. PADAUGINKITE IŠ 2.

* 1. **Mokymosi rezultatas. Taikyti objektinio programavimo principus programuojant.**
     1. **Tema. Objektinis programavimas.**

*1 užduotis.* SUKURKITE KLASĘ KNYGA.

Klasės savybės: isbn, autorius, leidimo metai, pavadinimas, aprašymas, puslapių skaičius. Klasė turi turėti konstruktorių naujos knygos objekto kūrimui. Tai pat turi būti sukurtas vienas *get* metodas, kuris grąžina masyvą su informacija apie knygą. Užtikrinkite, kad objekto savybės nebūtų pasiekiamos už klasės ribų (reikšmės išvedimas, naujos reikšmės priskyrimas).

*2 užduotis. 1 užduotyje* SUKURTOS KLASĖS PAGRINDU SUKURKITE 3 KNYGŲ OBJEKTUS.

Naudodami *get* metodą išveskite informaciją apie knygas naršyklės konsolėje.

* + 1. **Tema. Klasių instancijavimas ir metodai.**

*3 užduotis.* PRAPLĖSKITE *1 užduotyje* SUKURTĄ KLASĘ SUKURDAMI NAUJĄ KLASĘ PAVADINIMU ELKNYGA.

Naujai sukurta klasė turi paveldėti savybes ir metodus iš klasės *knyga*. Tai pat turi turėti naują savybę formatas ir metodą t*oArray*, kuris visą gražinamą informaciją apie knygą sudeda į masyvą.

*4 užduotis.**3 užduotyje* SUKURTĄ KLASĘ PRAPLĖSKITE STATINIU METODU PAVADINTU PALYGINIMAS.

Sukurtas metodas turi į masyvą patalpinti du sukurtus knygų objektus (kuriuos metodui paduodame kaip argumentus) ir juos išvesti naršyklės konsolėje.

* 1. **Mokymosi rezultatas. Keisti ir manipuliuoti HTML elementų medį (DOM).**
     1. **Tema*.* Darbas su JavaScript Document Object Model (DOM).**

*1 užduotis*. SUKURKITE PROGRAMĄ, KURI SUGENERUOJA HTML LENTELĘ.

Sukurtoje formoje įvedus stulpelių ir eilučių skaičių ir paspaudus mygtuką „Kurti“, sugeneruojama HTML lentelė.

*2 užduotis***.** SUKURKITE PROGRAMĄ, KURI HTML BODY ELEMENTUI PER NUSTATYTĄ LAIKĄ PAKEIČIA ATSITIKTINĘ FONO SPALVĄ.

*3 užduotis.*SUKURKITE PAVEIKSLĖLIŲ ATVAIZDAVIMO IR MAIŠYMO PROGRAMĄ.

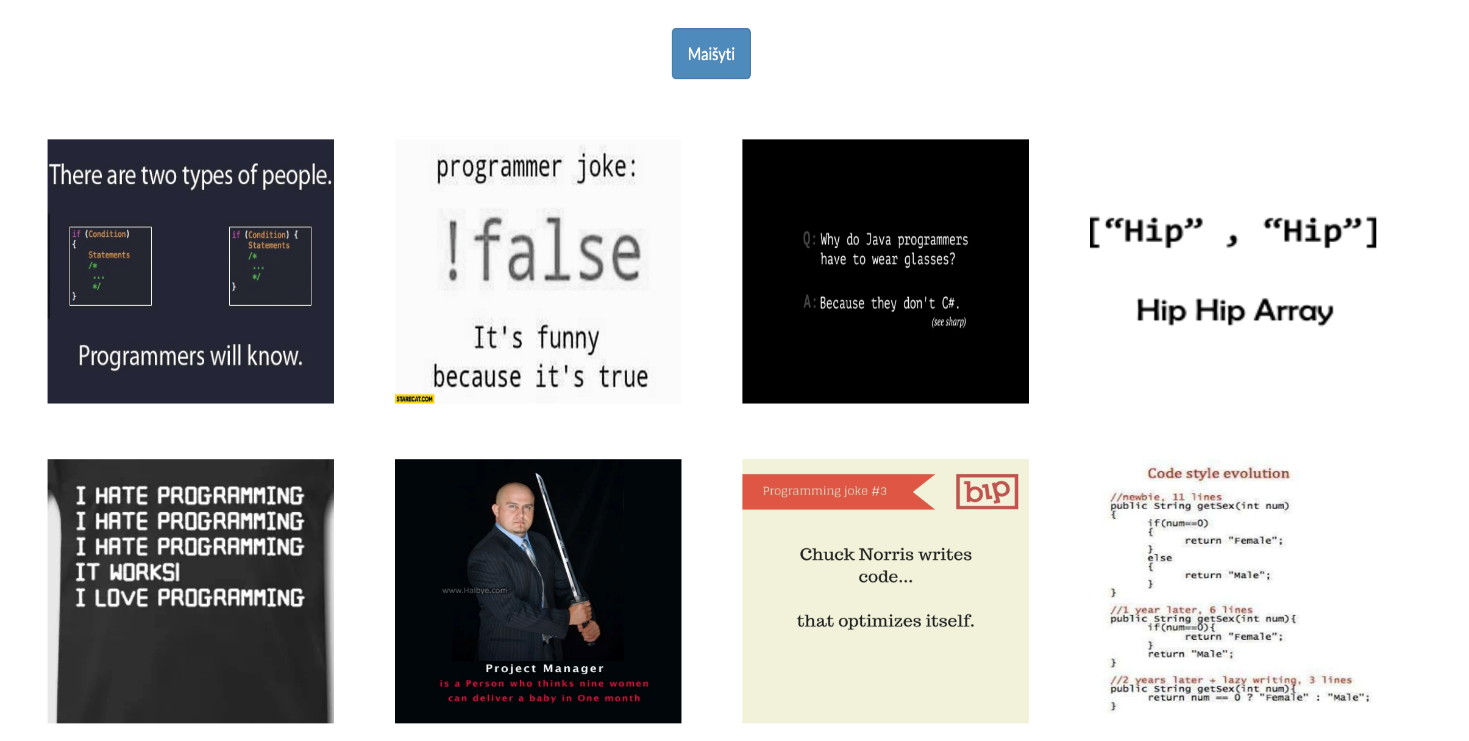
Paspaudus mygtuką rodyti į DOM įdedama 10 paveikslėlių (img elementai), kurie tvarkingai atvaizduojami.

Paspaudus mygtuką „Maišyti“, paveikslėliai išmaišomi atsitiktiniu būdu.

Paspaudus ant paveikslėlio du kartus, jis pakeičiamas numatytuoju paveikslėliu.

**Pavyzdys:**





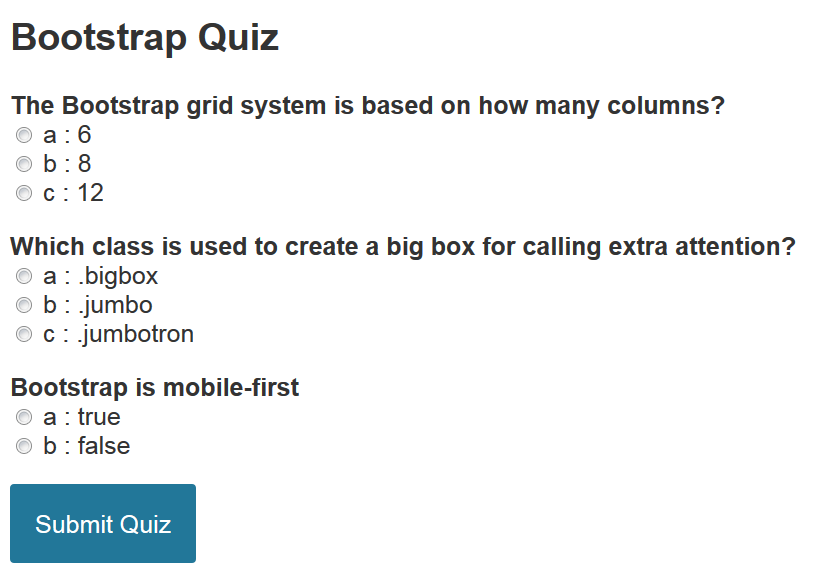
*4 užduotis.* SUKURKITE TESTĄ, TIKRINANTĮ IR PATEIKIANTĮ TEISINGUS ATSAKYMUS.

UŽDUOTIES SĄLYGA: sukurti testą, įgyvendinti testo sprendimą ir rezultatų pateikimą.

**Reikalavimai:**

1. Naudoti objekto duomenų tipą testo klausimams, atsakymų variantams ir teisingam atsakymui aprašyti.
2. Aprašyti funkciją, įkeliančia klausimus į HTML dokumentą iš objekto.
3. Aprašyti funkciją, skaičiuojančią testo rezultatus.

**Pavyzdys:**



*5 užduotis.* SUKURKITE DINAMIŠKĄ SĄVOKŲ SĄRAŠĄ.

UŽDUOTIES SĄLYGA: sukurti dinamišką sąvokų sąrašą pagal pateiktą pavyzdį.

DUOTA:

1. index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>WEB technologies</title>

</head>

<body>

<h1>WEB technologies</h1>

<div class="accordion">

<div class="accordion-header">HTML</div>

<div class="accordion-content">HTML is the standard markup language for creating Web pages.</div>

<div class="accordion-header">CSS</div>

<div class="accordion-content">CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media.</div>

<div class="accordion-header">JavaScript</div>

<div class="accordion-content">JavaScript is a scripting or programming language that allows you to implement complex things on web pages</div>

</div>

</body>

</html>

2. style.css

html {

min-height: 100%;

font-family: 'Nunito', sans-serif;

-webkit-font-smoothing: antialiased;

}

body {

background: linear-gradient(to bottom right, #56ab2f, #a8e063);

line-height: 1.5;

}

h1 {

font-weight: 200;

font-size: 3rem;

color: white;

text-align: center;

}

.accordion {

background: linear-gradient(to bottom right, white, #F8F8F8);

max-width: 400px;

background: white;

margin: 0 auto;

box-shadow: 0 15px 20px -15px rgba(0, 0, 0, 0.3), 0 35px 50px -25px rgba(0, 0, 0, 0.3), 0 85px 60px -25px rgba(0, 0, 0, 0.1);

}

.accordion-header {

border-bottom: 1px solid #DDE0E7;

color: #2a313a;

cursor: pointer;

font-weight: 700;

padding: 1.5rem;

}

.accordion-header:hover {

background: #F6F7F9;

}

.accordion-content {

display: none;

border-bottom: 1px solid #DDE0E7;

background: #F6F7F9;

padding: 1.5rem;

color: #4a5666;

}

.accordion-header::before {

content: '';

vertical-align: middle;

display: inline-block;

width: .75rem;

height: .75rem;

border-radius: 50%;

background-color: #B1B5BE;

margin-right: .75rem;

}

.active.accordion-header::before {

background-color: #d66d75 ;

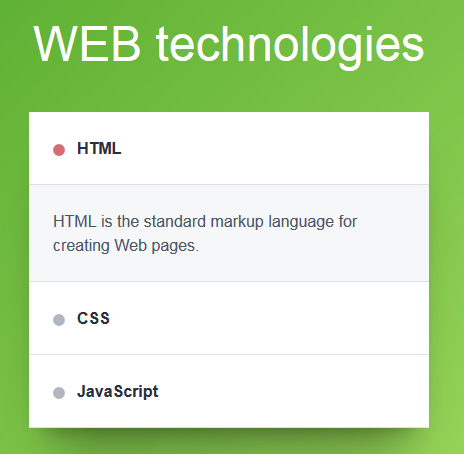
}

**Reikalavimai:**

1. HTML dokumente užrašyti nuorodas į stiliaus ir skripto failus.
2. Naudojant selektorius, įvykius, metodus darbui su CSS klasėmis.

**Pavyzdys:**





*6 užduotis.* SUKURKITE IŠŠOKANTĮ LANGĄ (MODAL BOX).

UŽDUOTIES SĄLYGA: naudojant JavaScript, užrašyti skriptą, kuris atvertų informacinį lauką (Modal Box), kai yra paspaustas mygtukas „open“. Iššokantis langas užsidaro kai paspaudžiamas mygtukas „close“ arba nuspaustus klaviatūroje „Esc“.

DUOTA:

1. index.html

<!DOCTYPE html>

<html lang=“en“>

<head>

<meta charset=“UTF-8“>

<meta name=“viewport“ content=“width=device-width, initial-scale=1.0“>

<meta http-equiv=“X-UA-Compatible“ content=“ie=edge“>

<title>Modal jQuery</title>

</head>

<body>

<h1>Modal</h1>

<div class=“overlay“></div>

<div class=“modal“>

<h2>Modal Box</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti perspiciatis magnam eum sed similique rerum laboriosam amet adipisci explicabo ipsa.</p>

<button class=“close“>Close</button>

</div>

<button class=“open“>Open</button>

</body>

</html>

2. style.css

html {

min-height: 100%;

font-family: ‚Nunito‘, sans-serif;

-webkit-font-smoothing: antialiased;

}

body {

background: linear-gradient(to bottom right, #30cfd0, #f6d365);

line-height: 1.5;

text-align: entre;

}

h1 {

font-weight: 200;

font-size: 3rem;

color: white;

text-align: entre;

}

.overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.5);

opacity: 0;

transition: .5s ease;

z-index: -1;

}

.overlay.active {

opacity: 1;

z-index: 2;

}

.modal {

max-height: calc(100% - 100px);

position: fixed;

top: 50%;

left: 50%;

max-width: 450px;

transform: translate(-50%, -50%);

color: #4a5666;

background: linear-gradient(to bottom right, white, #F8F8F8) !important;

box-shadow: 0 15px 20px -15px rgba(0, 0, 0, 0.3), 0 35px 50px -25px rgba(0, 0, 0, 0.3), 0 85px 60px -25px rgba(0, 0, 0, 0.1);

z-index: 3;

visibility: hidden;

opacity: 0;

transition: .5s ease;

}

.modal h2 {

margin: 0;

font-weight: 400;

padding: 1rem;

border-bottom: 1px solid #DDE0E7;

}

.modal p {

padding: 1rem;

}

.modal.active {

visibility: visible;

opacity: 1;

}

button {

display: inline-block;

background: rgba(10, 20, 30, .3);

border: 1px solid transparent;

color: white;

text-decoration: none;

font-size: 1.2rem;

padding: 1rem 2rem;

border-radius: 45px;

margin: .25rem 0;

vertical-align: middle;

line-height: 1;

overflow: visible;

white-space: nowrap;

cursor: pointer;

}

button:hover {

border: 1px solid rgba(255, 255, 255, .8);

color: white;

background: rgba(255, 255, 255, .1);

}

button.close {

margin: 0 0 1rem;

background: #FC6468;

}

button.close:hover {

border: 1px solid #FC6468;

color: #FC6468;

background: transparent;

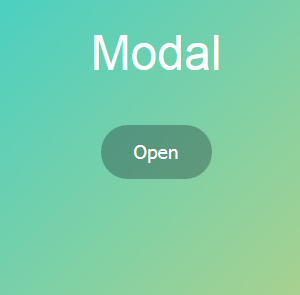
}

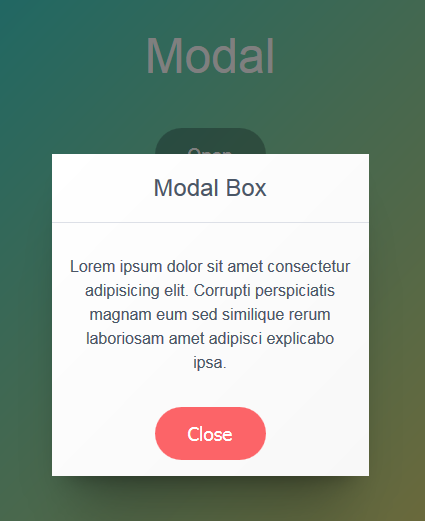
**Reikalavimai:**

1. HTML dokumente užrašyti nuorodas į stiliaus ir skripto failus.

2. Naudojant selektorius, įvykius, metodus darbui su CSS klasėmis.

**Pavyzdys:**





*7 užduotis.* SUKURKITE DINAMIŠKĄ LENTELĘ SU GALIMYBE PRIDĖTI IR PAŠALINTI LENTELĖS EILUTES.

UŽDUOTIES SĄLYGA: užrašyti skriptą, kuris įgyvendintų tokį funkcionalumą: duomenis, esančius formos laukeliuose įrašytų į lentelę, o mygtukas po lentele leistų ištrinti pažymėtas lentelės eilutes.

DUOTA:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS"

crossorigin="anonymous">

<title>Add/Remove Table Rows Dynamically</title>

</head>

<body>

<div class="container">

<div class="row mt-5">

<div class="col-10 offset-1">

<form>

<div class="form-group">

<input type="text" id="name" placeholder="Name">

<input type="text" id="email" placeholder="Email Address">

<input type="button" class="add-row btn btn-info" value="Add Row">

</div>

</form>

<table class="table table-striped">

<thead class="bg-info">

<tr>

<th>Select</th>

<th>Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td><input type="checkbox" name="record"></td>

<td>Name </td>

<td>name.surname@mail.com</td>

</tr>

</tbody>

</table>

<button type="button" class="delete-row btn btn-danger">Delete Row</button>

</div>

</div>

</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"

crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"

crossorigin="anonymous"></script>

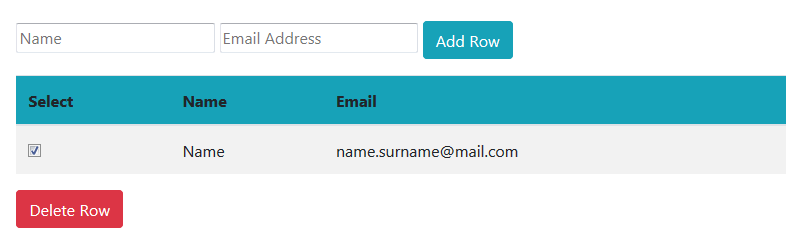
</body>

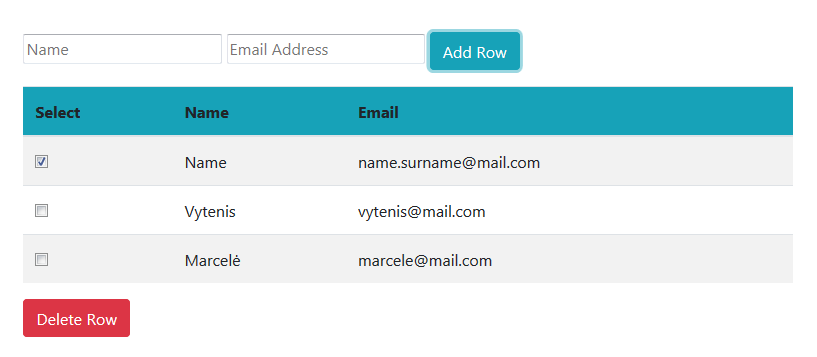
</html>

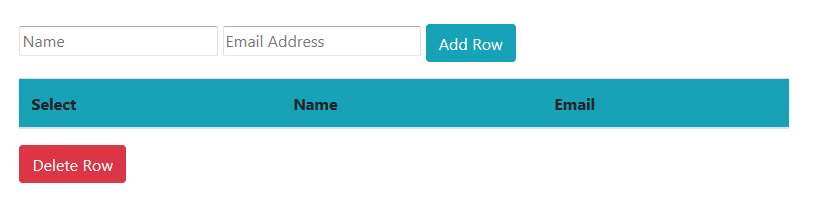
**Reikalavimai:**

1. HTML dokumente užrašyti nuorodas į skripto failus.
2. Naudojant selektorius, įvykius, metodus darbui su CSS klasėmis.

**Pavyzdys:**







* 1. **Mokymosi rezultatas. Kurti paprastas serveryje vykdomas aplikacijas naudojant JavaScript.**
     1. **Tema. NodeJS vykdymo aplinka.**

*1 užduotis*. KOKIE YRA NODE.JS APLINKOS NAUDOJIMO PRIVALUMAI IR SAVYBĖS?

*2 užduotis*. PARAŠYKITE KOMANDĄ, KURI IMPORTUOTŲ IŠORINĘ „HTTP“ BIBLIOTEKĄ NODE.JS PROJEKTE.

*3 užduotis*. PARAŠYKITE, KOKIA BUS ŠIO NODE.JS PROGRAMINIO KODO IŠVESTIS.

console.log("pirmas");

setTimeout(() => {

console.log("antras");

}, 0);

console.log("trecias ");

*4 užduotis*. PARAŠYKITE PROGRAMĄ NAUDODAMI ŠĮ FUNCKIONALUMO APRAŠYMĄ.

* Programos veikimo pradžoje vartotojas turi būti informuotas apie norimą įvestį (pvz: „Įveskite savo amžių:“ ).
* Vartotojui įrašius savo amžių, programa turi pakrinti ar šis žmogus yra pilnametis (t.y 18 metų ).
* Jeigu vartotojas yra nepilnametis - programa turi paskaičiuoti kiek metų liko iki pilnametystės.
* Jeigu įvedamas netinkamas simbolis – vartotojas informuojamas apie netinkamą įvestį.

**Programos veikimo pavyzdys:**

|  |
| --- |
| Įveskite savo amžių:  INPUT: -22  OUTPUT: -22 yra netinkama įvestis  INPUT: vienas  OUTPUT: vienas yra netinkama įvestis  INPUT: 10  OUTPUT: Jums nėra 18 metų, sugrįžkite po 8 metų  INPUT: 19  OUTPUT: Sveiki atvykę! Jums yra 19 metų |

*5 užduotis*: SUKURKITE NODE.JS HTTP WEB SERVERĮ, KURIS ATITIKTŲ ŠIUOS REIKALAVIMUS.

* Naudotų 3000 prievadą (angl. „port“)
* Nurodžius šakninį egzistuojantį serverio kelią (t.y: <http://127.0.0.1>:3*000* arba *http://localhost:3000*) vartotojas turi matyti *index.html* faile sudarytą turinį *(index.html išsaugotas esamoje direktorijoje)*
* Nurodžius neegzistuojantį serverio kelią (pvz: <http://127.0.0.1:3000/sveiki> arba *http://localhost:3000/sveiki*), turi būti iššaukta funkcija informuojanti vartotoją apie neegzistuojantį kelią: „Šis kelias neegzistuoja“.
* Paleižiant serverį konsolėje turi būti informuojama: „HTTP serveris veikia!“

***index.html*:**

<html>

<head>

<title>Sveiki</title>

</head>

<body>

Sveikas Pasauli!

</body>

</html>

**3.9.2. Tema. Express karkasas.**

*6 užduotis*. KAS YRA TARPINĖ PROGRAMINĖ ĮRANGA (ANGL. „MIDDLEWARE“) „EXPRESS JS“ KARKASE? KOKIOS YRA TARPINĖS PROGRAMINĖS ĮRANGOS PAGRINDINĖS FUNKCIJOS?

*7 užduotis*: UŽRAŠYKITE KELIŲ SUKŪRIMO (ANGL. „ROUTING“) STRUKTŪRĄ NAUDOJANT „EXPRESS JS“ KARKASĄ. PAAIŠKINKITE STRUKTŪROJE KIEKVIENĄ PANAUDOTĄ ELEMENTĄ.

*8 užduotis*. SUPROGRAMUOKITE ŽINIATINKLIO SERVERĮ NAUDODAMI „EXPRESS JS“ KARKASĄ. ŠIS SERVERIS TURI ATITIKTI ŠIUOS REIKALAVIMUS.

* Naudoti 3000 prievadą (angl. „port“)
* Nurodžius egzistuojantį serverio kelią vartotojas turi matyti registracijos formą su trimis įvesties elementais bei registracijos formos patvirtinimo mygtuką. Nurodžius vardą, elektroninį paštą ir metus, vartotojas turi turėti galimybę patvirtinti registracijos formą.
* Visi įvesti duomenys turi būti validūs, nustatant įvesties standartus naudojant „express-validator“ biblioteką:
  1. Vardo įvesties vietoje turi būti daugiau nei 3 simboliai.
  2. Elektroninis paštas turi atitikti standartinius reikalavimus.
  3. Amžius turi būti užrašytas skaitine forma.
* Jeigu įvestis yra netinkamo formato, vartotojas turi būti informuotas, kuri dalis yra neteisinga.
* Patvirtinus teisingą registracijos formą vartotojas turi būti informuojamas apie sėkmingą registraciją išvedant įvestus duomenis. Naudokite „ejs“ biblioteką, kuriant klaidų informacinius pranešimus (galima naudoti pavyzdinį index.ejs failo turinį).

index.ejs: (patalpintas kataloge “views”)

<form method="post" action="/registracija">

Vardas:

<br>

<input type="text" name="vardas" value="">

<br>

El. paštas:

<br>

<input type="text" name="pastas" value="">

<br>

Amzius:

<br>

<input type="text" name="amzius" value="">

<br>

<br>

<input type="submit" value="Pateikti">

</form>

<% for (var i = 0; i < klaida.length; i++) { %>

<p style="color: red">

<%= klaida[i]%>

</p>

**3.9.3. Tema. Duomenų bazės.**

*9 užduotis*. SURAŠYKITE Į REIKIAMĄ LENTELĖS STULPELĮ IŠVARDINTAS DUOMENŲ BAZIŲ SAVYBES PRISKIRDAMI JAS ARBA REALIACINĖMS (SQL) ARBA NOSQL DUOMENŲ BAZĖMS.

Savybės: Dinaminė schema; Iš anksto nustatyta schema; Duomenys saugomi: rakto-reikšmės porose (angl. key-value pairs); Duomenys saugomi lentelėse; Duomenys saugomi dokumentų saugyklose (angl. document based); Duomenys saugomi grafų duomenų bazėse; Duomenys saugomi kaip stulpelių saugykla (angl. wide-column stores); Yra optimizuotos darbui su sudėtingomis užklausomis; Gali dirbti su labai dideliais duomenų kiekiais realiu laiku; Užtikrina duomenų vientisumą ir neprieštaringumą po kiekvienos transakcijos; Užtikrina duomenų vientisumą ir neprieštaringumą tik praėjus tam tikram laikui.

|  |  |
| --- | --- |
| **Reliacinės (SQL) duomenų bazės** | **NoSQL duomenų bazės** |
|  |  |
|  |  |
|  |  |
|  |  |

*10 užduotis*. ĮDIEKITE MONGODB DUOMENŲ BAZĘ. ĮDIEKITE NODE.JS MODULĮ, SKIRTĄ DARBUI SU MONGO DB.

Parašykite NODE.JS programą, kuri sukurtų duomenų bazę vardu „MYDB“, SUKURTŲ duomenų rinkinį vardu „CARS“ ir įkelkite į rinkinę 3 automobilius, sutekdami jiems gamintoją „BRAND“, modelį „MODEL“, SPALVĄ „COLOR“ ir numerį „NUMBER“, iš kurių bent du turėtų juodą spalvą.

*11 užduotis.* Parašykite NODE.JS programą, kuri surastų sukurtoje „MYDB“ duomenų bazėje automobilius, kurie turi juodą spalvą. Kaip rezultatą programa turi atspausdinti visus surastus įrašus su visais duomenų laukais.

*12 užduotis*. ĮDIEKITE NODE.JS MONGOOSE MODULĮ.

Sukurkite NODE.JS programą, kuri sukurtų modelį CARS, turintį dvi savybes „BRAND“ IR „GAS“, bei sukurkite šio modelio įrašą ir jį išsaugokite į duomenų bazę „CARS“. Taip pat modeliui „CARS“ sukurkite metodą „PRINT“, kuris atspausdintų automobilio gamintoją „BRAND“ ir jo kuro kiekį „GAS“.

*13 užduotis*. NAUDODAMI NODE.JS MONGOOSE MODULĮ, SUKURKITE PROGRAMĄ, KURI PRISIJUNGTŲ PRIE SUKURTOS DUOMENŲ BAZĖS „CARS“ BEI SURASTŲ ĮRAŠĄ IŠ ŠIOS DUOMENŲ BAZĖS.

* 1. **Mokymosi rezultatas. Kurti aplikacijas naudojant React karkasą.**
     1. **Tema. Populiarių karkasų palyginimas.**

*1 užduotis*. PASIRINKITE TEISINGĄ ATSAKYMĄ:

1. Kuris iš išvardintų karkasų yra paremtas komponentais ?

a) Vue

b) React

c) Angular

d) Visi aukščiau išvardinti

2. Kuris iš karkasų palaiko JSX sintaksę ?

a) Vue

b) Angular

c) React

d) Visi aukščiau išvardinti

3. Kuris iš karkasų turi native puslapiavimą(routing)?

a) React

b) Angular

c) Vue

d) Visi aukščiau išvardinti

4. Kurie karkasai palaiko Virtual DOM technologiją ?

a) React ir Angular

b) Angular ir Vue

c) React ir Vue

5. Kuris karkasas palaiko Two-way Data Binding technologiją ?

a) React

b) Angular

c) Vue

d) Visi aukščiau išvardinti

6. Kuris karkasas pasižymi didžiausiu lankstumu ?

a) React

b) Angular

c) Vue

d) Visi aukščiau išvardinti

7. Kuris karkasas pasižymi „švariu“ ir lengvai suprantamu kodu ?

a) React

b) Angular

c) Vue

d) Visi aukščiau išvardinti

8. Kuris karkasas naudoja MVC architektūrą ?

a) React

b) Angular

c) Vue

d) Visi aukščiau išvardinti

**3.10.2. Tema. React karkasas.**

*2 užduotis.*ĮDIEKITE NODE JS SAVO KOMPIUTERYJE.

Patikrinkite ar korektiškai veikia. Pabandykite terminale įvykdyti JavaScript kodą

*3 užduotis.*<https://reactjs.org/community/examples.html> TINKLAPYJE YRA PATEIKTI DEMO PROJEKTAI SU NUORODA Į GITHUB.

Išsirinkite vieną projektą ir nusiklonuokite. Paanalizuokite kodą. Paleiskite projekto aplikaciją savo kompiuteryje naudojant npm.

*4 užduotis.* SUKURKITE NAUJĄ REACT APLIKACIJĄ PAVADINIMU ‘KNYGOS‘.

Aplikacijos paskirtis: atlikti paiešką knygų sąraše.

Privalomi komponentai:

* Knygų sąrašas su paieškos laukeliu
* Apie projektą

Kuriant komponentų architektūrą rekomenduojama naudotis React „stateless functional components“ ir „higher-order components“ (HOC) technikomis.

Navigacijai tarp skirtingų puslapių privaloma naudoti *react-router.*

Duomenys tarp komponentų turi būti perduodami, kaip *props*.

Privaloma naudoti reikalingus *lifecycle* metodus.

1. **Kompetencija. Analizuoti skirtingų tipų reikalavimus, apibūdinančius kompiuterinę programą.**
2. 1. **Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo reikalavimų programinei įrangai formatą.**

**4.1.1. Tema. Reikalavimų programinei įrangai formatai *.***

*1* *užduotis.* REIKALAVIMŲ PROGRAMINEI ĮRANGAI FORMATAI.

Pateiktam reikalavimui: a) nustatyti reikalavimo formatą; b) sudaryti testavimo atvejų sąrašą (detalių žingsnių kurti nereikia).

Užduoties tikslas - sugebėti atskirti funkcinių reikalavimų formatus bei mokėti apibrėžti testavimo apimtis remiantis funkcinių reikalavimų dokumentais (nepriklausomai nuo formato).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Reikalavimas #:** | R1 | **Reikalavimo tipas:** |  | **Įvykis/panaudojimo atvejis #:** | 1 |
|  |  |  |  |  |  |
| **Aprašymas:** | Sistema turi leisti registruotis bendrovės darbuotojui | | | | |
|  |  |  |  |  |  |
| **Pagrindimas:** | Darbuotojas gali suklysti įvedinėjant registravimosi duomenis, todėl reikalinga galimybė juos pakoreguoti. | | | | |
|  |  |  |  |  |  |
| **Šaltinis:** | Bendrovės darbuotojas. | | | | |
| **Tinkamumo kriterijus:** | Atidaromas pagrindinis sistemos langas su priregistravimo kortele. | | | | |
|  |  |  |  |  |  |
| **Priklausomybės:** | Nėra | | **Konfliktai:** | | Nėra |
| **Papildoma medžiaga:** |  | | | | |
| **Istorija:** | Užregistruotas 2008m. sausio 15d. | | | |  |

**Panaudojimo atvejis Nr. 1**

|  |  |
| --- | --- |
| Nr. | 1 |
| Pavadinimas: | Prisijungimas prie sistemos. |
| Vartotojas/aktorius: | Bendrovės darbuotojas |
| Aprašas: | Apima prisijungimo prie sistemos procesą. |
| Prieš sąlyga: | Vartotojas yra registruotas sistemoje ir turi priskirtą slaptažodį. |
| Sužadinimo sąlyga: | Slaptažodžio teisingas įvedimas. |
| Po sąlyga: | Atidaromas pagrindinis sistemos langas. |

*2 užduotis.* UŽRAŠYKITE VARTOTOJO PASAKOJIMO (USER STORY) FORMATĄ.

Pateikite pavyzdžių.

*3 užduotis.* YRA PATEIKTI FUNKCINIAI PROGRAMINĖS ĮRANGOS REIKALAVIMAI.

Suformuokite tris vartotojo pasakojimus (user stories)

|  |
| --- |
| **FR1. Sistemoje egzistuoja vartotojai**  Sistemoje egzistuoja vartotojai, kurie gali dirbti su sistema prisijungę. Prisijungimas vykdomas pateikiant vartotojo vardą ir slaptažodį.  **FR2. Sistemoje egzistuoja administratoriaus rolė**  Sistemoje egzistuoja speciali vartotojų rolė - administratoriai. Administratoriai gali:   * kurti vartotojų grupes (žiūrėti FR3), priskirti joms vartotojus * kurti dokumentų tipus ir priskirti grupes galinčias kurti dokumentus bei juos priimti * registruoti naujus vartotojus sistemoje * keisti esamų vartotojų slaptažodžius |

*4 užduotis.* APRAŠYKITE GERŲ VARTOTOJO PASAKOJIMO FORMATĄ (INVEST).

**4.1.2. Tema. Reikalavimų peržiūros procesas.**

*4 užduotis.*  REIKALAVIMŲ PERŽIŪROS PROCESAS.

Žemiau pateiktą panaudojimo atvejį užrašykite Vartotojo istorijos (vartotojo pasakojimo) formate bei nustatykite reikalavimo paruošimo kriterijus (“Definition of Ready”) bei užtikrinkite aiškius priėmimo kriterijus (“Acceptance criteria”).

Užduoties tikslas - suprasti, kam reikalinga reikalavimų peržiūra, kada ji yra vykdoma, kokie yra peržiūros privalumai ir naudos komandai.

**Panaudojimo atvejis Nr. 2**

|  |  |
| --- | --- |
| Nr. | 2 |
| Pavadinimas: | Išeiti iš sistemos. |
| Vartotojas/aktorius: | Bendrovės darbuotojas. |
| Aprašas: | Apima atsijungimo nuo sistemos ir programos uždarymo procesą. |
| Prieš sąlyga: | Bet kuris darbo etapas sistemoje. |
| Sužadinimo sąlyga: | Pele sužadinamas mygtukas “Atsijungti”. |
| Po sąlyga: | Programa uždaryta. |

* 1. **Mokymosi rezultatas. Naudoti funkcinius, nefunkcinius ir techninius kompiuterinės programos reikalavimus.**

**4.2.1. Tema. Funkciniai reikalavimai.**

*1 užduotis*. FUNKCINIAI REIKALAVIMAI.

Patikrinkite, ar veikianti žiniatinklio programa atitinka atitinka žemiau apibrėžtus funkcinius reikalavimus. Pateikti neatitikimų sąrašą, jei tokių yra.

Užduoties tikslas - išmokti naudoti funkcinius reikalavimus testuojant programinę įrangą. Išmokti nustatyti neatitikimus tarp reikalavimuose apibrėžtų ir realizuotų sistemos funkcijų.

Eiga: OpenCart sistemoje (teirautis nuorodos iš dėstytojo), atidarykite paieškos langą paspausdami ant paieškos mygtuko, arba prisijunkite su pateikiamais ‘demo’ paskyros rekvizitais prie https://demo.opencart.com/admin/ ir vietoje produkto paieškos naudokite filtravimą skiltyje ‘Catalog’. Įvesdami įvairius paieškos kriterijus, įsitikinkite, kad Sistema veikia taip, kaip nurodo žemiau pateikti reikalavimai. Pateikite skirtumų sąrašą, jei tokių pastebėsite.

Opencart – Paieška (supaprastinti paieškos funkcionalumo reikalavimai)

- Jei į paieškos lauką įvedamas produkto tikslus pavadinimas ir spaudžiamas "Paieška" mygtukas, sistema pateikia sąrašą visų šio produkto atmainų, turimų pardavime.

- Jei įvedamas raktažodis pagal produkto kategoriją ir spaudžiamas "Paieška" mygtukas, sistema pateikia atitinkamos kategorijos produktų saraš.

- Jei į paieškos laukelį nieko neįvedame ir spaudžiame "Paieška" mygtuką, sistema grąžina visų turimų prekių sąrašą.

- Jei į paieškos laukelį įvedame neegzistuojačio produkto pavadinimą ir spaudžiame "Paieška" mygtuką, sistema pateikia pranešimą, apie tokio produkto nebuvimą.

- Jei paieškos rezultatų yra daug, jie yra skirstomi į puslapius po 10 rezultatų kiekviename puslapyje.

- Paieškos rezultatai yra surūšiuoti abėcėlės tvarka.

**4.2.2. Tema. Nefunkciniai reikalavimai.**

*2* *užduotis.* NEFUNKCINIAI (TECHNINIAI, SAUGUMO) REIKALAVIMAI.

Atsakykite į žemiau pateiktus klausimus. Kiekvienas klausimas gali turėti vieną arba daugiau teisingų atsakymų.

Užduoties tikslas - gebėti skirti funkcinius sistemos reikalavimus nuo nefunkcinių. Suprasti ir gebėti paaiškinti nefunkcinių reikalavimų rūšis.

1. Kokie yra pagrindiniai skirtumai tarp funkcinių ir nefunkcinių sistemos reikalavimų?

|  |
| --- |
| Atsakymas: |

1. Pateikite bent penkis nefunkcinių reikalavimų testavimo tipus.

|  |
| --- |
| Atsakymas: |

1. Kodėl svarbu žiniatinklio programas testuoti su skirtingomis naršyklėmis?

|  |
| --- |
| Atsakymas: |

1. Pateikite bent tris pavyzdžius, kaip naršyklių teikiamas„developer tools“ įrankių rinkinys gali būti taikomas žiniatinklio programos testavimui.

|  |
| --- |
| Atsakymas: |

1. Žemiau pateiktus reikalavimus sugrupuokite į atitinkamas reikalavimų grupes:
   1. Sistema turi gebėti aptarnauti vidutiniškai 500 vartotojų vienu metu;
   2. Turi būti galimybė vartotojui pačiam susikurti/redaguoti/šalinti savo paskyrą;
   3. Vartotojas gali įkelti paskyros nuotrauką, kuri neviršija 3MB;
   4. Esant poreikiui, sistema turi pateikti vartotojui ataskaitą apie jo aktyvumą;
   5. Vartotojo duomenų pakeitimų išsaugojimas turi užtrukti ne ilgiau nei 0,15 sekundės;
   6. Sistema turi palaikyti vartotojo identifikavimą mobiliuoju parašu;
   7. Sistema turi būti pritaikyta neįgaliesiems;
   8. Vartotojas gali parsisiųsti ataskaitą apie savo aktyvumą .csv formatu;
   9. Sistema turi palaikyti dvi kalbas (LT, EN);
   10. Sistemos sutrikimo atveju turi būti galimybė atstatyti prarastus duomenis.

|  |  |
| --- | --- |
| Funkciniai reikalavimai: | Nefunkciniai reikalavimai: |

1. Ar reikalavimų peržiūrą priskirtumėte funkciniam ar nefunkciniam testavimui? Kodėl?

|  |
| --- |
| Atsakymas: |

1. Kas yra atsakingas už nefunkcinių reikalavimų atitikimą kuriant programą?
   1. Programuotojai
   2. Testuotojai
   3. Devops
   4. Inžinieriai tinkle
2. Kas yra lokalizacijos testavimas?
   1. Kai sistema (programa) tikrinama ar tinkamai palaiko kelias kalbas
   2. Kai sistema yra tikrinama ar teisingai verčiamas jos turinys į kitą kalbą dinamiškai
   3. Kai tikrinama, ar sistema tinkamai prisitaiko prie regiono, laiko juostų pokyčių ir atitinkamai pritaiko specifinius jos komponentus bei verčia turinį
   4. Kai tikrinama, ar sistema geba atpažinti šalį bei atitinkamai pritaikyti turinio vertimus
3. Kaip galime apibūdinti sistemos panaudojamumą?

|  |
| --- |
| Atsakymas: |

Panaudojamumas (angl. *usability*) – tai naudotojo veiklos veiksmingumas, našumas ir jaučiamas pasitenkinimas, su kuriuo konkretus naudotojas gali pasiekti konkrečių tikslų konkrečiose aplinkose, šiuo atveju, naudodamasis mūsų sistema.

1. Ką tikrina suderinamumo testavimas? (pasirinkite geriausiai tinkanti apibūdinimą)
   1. Ar kuriama programa yra suderinama su duomenų baze, operacine sistema, kita programine bei technine įranga
   2. Ar kuriama programa gali dirbi sinchronizuotai su keliomis duomenų bazėmis vienu metu
   3. Ar kuriama programa gali pilnavertiškai veikti tiek su Windows, tiek su Linux operacinėmis sistemomis
   4. Ar kuriama programa nelūžta po atnaujinimo
   5. **Mokymosi rezultatas. Testuoti programinę įrangą naudojant su JavaScript programavimo kalba suderinamus testavimo įrankius ir metodus.**

**4.3.1. Tema. Testavimo principai.**

*1* *užduotis.*  DUOTA FUNKCIJA. KIEK REIKĖS TESTAVIMO SCENARIJŲ, KAD PADENGTI VISUS ATVEJUS? IŠVARDINKITE JUOS.

*If a > b*

*b = 12*

*if a > c*

*c = a*

*else*

*c = b*

*else*

*c = 0*

*end if*

2 *užduotis.*  IŠVARDINKITE PAGRINDINES RIZIKAS/ NAUDAS, KODĖL SVARBU PROGRAMOS KLAIDĄ SURASTI KUO ANKSČIAU.

**4.3.2. Tema. Modulių (unit) testavimas.**

*3* *užduotis.*  ATSAKYKITE Į ŽEMIAU PATEIKTUS KLAUSIMUS.

Kiekvienas klausimas gali turėti vieną arba daugiau teisingų atsakymų.

1. Kaip vadinamas testavimo lygmuo, kuriame testus kuria dažniausiai tik programuotojai?
   1. Komponentų testavimas
   2. Integracijų testavimas
   3. Sistemos testavimas
   4. Našumo testavimas
2. Kurio lygmens testai dažniausiai yra skirti testuoti sąsajas tarp komponentų arba sistemų?
   1. Komponentų
   2. Integracijų
   3. Sistemos
   4. Vartotojų priėmimo
3. Kuris iš žemiau pateiktų tikslų priskiriamas komponentų lygmens testavimui?
   1. Rasti komponentų defektus
   2. Nustatyti jei funkcinis ir nefunkcinis sistemos sąsajų darbas atitinka specifikaciją
   3. Nustatyti jei funkcinis ir nefunkcinis sistemos darbas atitinka specifikaciją
   4. Nustatyti, ar sistema yra išbaigta ir tinkamai funkcionuoja
4. Kokiame testavimo lygmenyje dažniausiai testuojame: vartotojo procedūras, konfigūracijas, formas ir ataskaitas, verslo procesus?
   1. Komponentų
   2. Integracijų
   3. Sistemos
   4. Vartotojų priėmimo
5. Kas yra dažniausiai atsakingas už testavimo vykdymą priklausomai nuo testavimo lygmens? Sužymėkite “X” atitinkamuose langeliuose.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atsakinga grupė | Komponentų testavimas | Integracijos testavimas | Sistemos testavimas | Priėmimo testavimas |
| Programuotojai |  |  |  |  |
| Testuotojai |  |  |  |  |
| Verslo atstovai / galutiniai vartotojai |  |  |  |  |

1. Kaip vadinama testavimo veikla, kai testavimas atliekamas gamintojo organizacijoje, tačiau: arba nepriklausomos testavimo komandos, arba potencialių klientų?
   1. Beta testavimas
   2. Regresinis testavimas
   3. Alfa testavimas
   4. Nepriklausomas (Objektyvus) testavimas
2. Kas yra Beta testavimas?
   1. Testavimo veikla, atliekama testuotojų gamintojo organizacijoje
   2. Testavimo veikla, atliekama užsakovo, gamintojo aplinkoje
   3. Testavimo veikla atliekama potencialių klientų jų patalpose/vietoje
   4. Testavimo veikla, organizuojama nepriklausomos orgamizacijos su tikslu priimti pagamintą IS.
3. Ką testuojame sistemos lygmenyje?
   1. Programas /sistemas
   2. Sistemos reikalavimus
   3. Sistemos sąsajas
   4. Verslo procesus
4. Kokios yra tipinės integracinio lygmens klaidos?
   1. Skaičiavimų klaidos
   2. Duomenų srautų klaidos
   3. Duomenų surišimo klaidos (data mapping)
   4. Logikos klaidos
5. Kuo remiamės atlikdami testavimą component lygmenyje?
   1. Sąsajų specifikacijomis
   2. Rizikos vertinimo ataskaitomis
   3. Sistemos reikalavimais
   4. Programos kodu
6. Kas negerai su šiuo metodu: public void EnterInputName() { inputName.sendKeys("My Name"); }?

a. Viskas gerai

b. Jis turi įkoduotus duomenis, geriau perduoti testo duomenis kaip parametrą

c. Metodas turėtų būti privatus, o ne viešas

d. Jis negrąžina jokios reikšmės, būtų geriau jei grąžintų vardą, kuris buvo įvestas

**4.3.3. Tema. Pilnas (End-to-end) testavimas.**

*4* *užduotis.*  ATSAKYKITE Į ŽEMIAU PATEIKTUS KLAUSIMUS.

Kiekvienas klausimas gali turėti vieną arba daugiau teisingų atsakymų.

Užduoties tikslas - mokėti testavimo tipus; gebėti atskirti skirtingus testavimo tipus; suprasti, kokie testavimo metodai būdingi kiekvienam testavimo tipui; skirti bazinių testų ir regresijos sąvokas.

1. Žemiau pateiktus testavimo tipus suskirstykite į atitinkamas kategorijas ir įrašykite į lentelę.

Komponentų testavimas, Našumo testavimas, Priėmimo testavimas, Apkrovos testavimas, Saugumo testavimas, Palaikymo testavimas, Prieinamumo testavimas, Lokalizavimo testavimas.

|  |  |  |
| --- | --- | --- |
| Funkcinis testavimas | Nefunkcinis testavimas | Palaikymo testaviams |

1. Ką vadiname juodos dėžės testavimu?
   1. Programos testavimas kai žinome programos kodą
   2. Programos testavimas kai nežinome programos kodo
   3. Testavimas, kuris yra paremtas vidine struktūros analize
   4. Testavimas iš vartotojo perspektyvos, bet tikrinant duomenis duomenų bazėje
2. Ką vadiname baltos dėžės testavimu?
   1. Programos testavimas kai žinome programos kodą
   2. Programos testavimas kai nežinome programos kodo
   3. Testavimas, kuris yra paremtas vidine struktūros analize
   4. Testavimas iš vartotojo perspektyvos, tikrinant duomenis duomenų bazėje
3. Kokie yra juodos dėžės testavimo privalumai?
   1. Testuotojas puikiai žino programos vidinę struktūrą
   2. Testuotojas ir programuotojas nepriklauso vienas nuo kito
   3. Testuojama iš naudotojo perspektyvos
   4. Gali būti ištestuoti tik tam tikri įvesties poaibiai
4. Kokie yra baltos dėžės trūkumai?
   1. Testuotojas puikiai žino programos vidinę struktūrą
   2. Testuotojas ir programuotojas nepriklauso vienas nuo kito
   3. Parenkami duomenys, kad būtų ištestuoti visi galimi įvesties poaibiai
   4. Padeda optimizuoti kodą
5. Kokie yra juodos dėžės trūkumai?
   1. Sunku sukurti testavimo atvejus, neturint tikslių specifikacijų
   2. Testavimo atvejai gali kartotis, testuotojas nežino, ką programuotojas ištestavo
   3. Gali likti neištestuota daug kelių per programą
   4. Testuotojas jaučiasi izoliuotas nuo komandos
6. Suskirstykite testavimo metodus pagal testavimo tipus ir įrašykite į lentelę:

testavimas pagal reikalavimų specifikaciją; našumo testavimas; sakinių testavimas; apkrovos testavimas; atšakų testavimas; scenarijų testavimas; kelių testavimas; modifikuotas sprendimų testavimas; atsitiktinis testavimas; panaudos atvejų testavimas

|  |  |
| --- | --- |
| Juodosios dėžės testavimo metodai: | Baltosios dėžės testavimo metodai: |

1. Kas yra regresijos testavimas?
   1. Testavimas atliekamas su tikslu įsitikinti, kad defektas yra ištaisytas
   2. Testavimas atliekamas su tikslu įsitikinti, kad visos sistemos funkcijos veikia po sistemos ar jos dalies pakeitimo
   3. Testavimas atliekamas su tikslu įsitikinti, kad sistema atlieką funkcijas, kurias turi atlikti
   4. Testavimas atliekamas su tikslu įsitikinti, kad sistema neatlieka to, ko atlikti neturi
2. Kas yra baziniai (smoke) testai?
   1. Sudėtingi verslo scenarijai, patikrinantys visą sistemą
   2. Baziniai visos sistemos testai, tam kad įsitikinti, jog verta testuoti toliau
   3. Vartotojo priėmimo testų poaibis
   4. Automatiniai integraciniai testai
3. Kas yra patvirtinimo testavimas?
   1. Regresijos testų dalis, vykdoma sistemos surinkimo metu
   2. Pakartotinis, anksčiau nesėkmingai įvykdyto testavimo atvejo vykdymas, siekiant įsitikinti, kad defektas ištaisytas
   3. Pakartotinis testavimas po atliktų pakeitimų, siekiant patvirtinti, jog neatsirado naujų defektų nekeistose sistemos vietose
4. Koks pavadinimas web sąsajos elementui teste (pvz. "Save" mygtukui) geriausiai tinkamas?
   1. clickSave
   2. buttonSave
   3. optionSave
   4. elementSave
   5. webElementSave

**Modulis „Nesudėtingų duomenų bazių projektavimas ir kūrimas (JavaScript)“**

1. **Kompetencija. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes.**
   1. **Mokymosi rezultatas. Pateikti reliacinės duomenų bazės schemą.**

**1.1.2. Tema. Duomenų bazių projektavimas (CREATE TABLE sakinys, duomenų normalizavimas).**

*1 užduotis*. SUPROJEKTUOTI RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LIETUVOS BANKAS).

Lietuvos bankas paprašo jūsų padaryti jiems DB. Jie nori, kad toje DB būtų sudėti duomenys apie visus Lietuvos žmones (jų asmens kodas, v**a**rdas, pavardė, telefono numeris), tam, kad jie galėtų išsiaiškinti, kokiuose bankuose jie turi sąskaitas ir koks jų pinigų balansas-likutis jose yra. Suprojektuokite tokią duomenų bazę, atitinkančią 3 normalinės formos reikalavimus.

Projekte turi būti nurodyti pirminiai ir antriniai raktai, ryšiai tarp lentelių ir jų tipai. (P.S. ne mažiau 3 lentelės).

Kontrolinis klausimas: kuriuose bankuose Petro balansas didesnis už 1000?

*2 užduotis*. SUPROJEKTUOTI RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LEIDYBOS STEBĖJIMO SISTEMA).

„Poezijos būrelis“ – tai leidykla, leidžianti poeziją ir poezijos rinkinius. Reikalinga sistema, kurioje būtų galima registruoti poetus, eilėraščius, jų rinkinius ir pardavimą.

* Poetu gali būti visi norintieji, ne vien tik užfiksavę savo poeziją sistemoje arba jau sukūrė poezijos kūrinį.
* Poezijos kūrinius galima pateikti internetu, el. paštu arba popieriuje.
* Visi sistemoje užfiksuoti poezijos kūriniai parašyti poetų, kurių duomenys jau užfiksuoti sistemoje. Nepateikęs visos reikalaujamos informacijos, poetas negalės pateikti ir išsaugoti savo poezijos sistemoje.
* Leidinį gali sudaryti ir vienas eilėraštis, ir poezijos rinkinys, ir literatūros kritikos kūrinys.
* Leidiniai parduodami klientams, apie kuriuos sistemoje yra informacijos. Anonimiški pirkėjai neaptarnaujami.
* Vieno pardavimo metu galima parduoti ir vieną leidinį, ir kelis. Tačiau jei parduodama keliems klientams, leidykla tai traktuoja kaip pardavimą kelis kartus. Kiekvienas klientas aptarnaujamas atskirai.
* Parduodami ne visi leidiniai. Sunkiau parduoti, pavyzdžiui, kokius nors ypatingus leidimus. O kai kurių leidinių nepavyksta parduoti nei egzemplioriaus.

Suprojektuokite tokią duomenų bazę, atitinkančią 3 normalinės formos reikalavimus. Projekte turi būti nurodyti pirminiai ir antriniai raktai, ryšiai tarp lentelių ir jų tipai. (P.S. ne mažiau 7 lentelės, įmanoma ir su 6). Ne visa informacija yra skirta projektavimo aspektams.

Kontrolinis klausimas: kuris autorius populiariausias?

**1.2. Mokymosi rezultatas. Pateikti nereliacinės (NoSQL) duomenų bazės schemą.**

**1.2.2. Tema. NoSQL duomenų bazių tipai ir jų savybės.**

Nereliacinių (NoSQL) duomenų bazių užduotims atlikti rekomenduojamos duomenų bazės: Redis, Cassandra, MongoDB ir Neo4J. Galimi sprendimai pateikiami naudojant būtent šias duomenų bazes.

**Scenarijus**

Sumodeliuokite duomenų bazę parduotuvei. Parduotuvės duomenų bazėje saugomos pačios parduotuvės ir jų prekės.

Parduotuvės. Kiekviena parduotuvė identifikuojama jos kodu (pavyzdžiui „V1“, „K2“ar pan.). Kiekvienai parduotuvei saugoma jos plotas kvadratiniais metrais, adresas.

Prekės. Kiekvienoje parduotuvėje saugomas joje esančių prekių sąrašas. Kiekviena prekė turi unikalų skaitinį kodą (pavyzdžiui „11559245“). Kiekvienai prekei saugomas jos pavadinimas, vieneto svoris, vieneto kaina, kiekis.

*1 užduotis*. PATEIKITE *KEY-VALUE* DUOMENŲ BAZĖS MODELĮ PARDUOTUVEI. ĮVERTINKITE, KAIP SAUGOTI SUDĖTINIUS RAKTUS, KAIP SAUGOTI REIKŠMES, RYŠIUS TARP PREKIŲ IR PARDUOTUVĖS.

Operacijos su duomenų baze:

* Gauti parduotuvės informaciją
* Gauti konkretaus tipo prekės, konkrečioje parduotuvėje informaciją
* Bonus: gauti visas prekes esančias parduotuvėje

*2 užduotis*. PATEIKITE COLUMN FAMILY DUOMENŲ MODELĮ PARDUOTUVĖS SCENARIJUI.

Duomenų bazėje turi būti galimybė vykdyti šias užklausas:

1. Išrinkti parduotuvės informaciją pagal jos kodą.
2. Išrinkti visas prekes, esančias pasirinktoje parduotuvėje.
3. Sužinoti, kiek vienetų prekių su pasirinktu kodu yra parduotuvėje.

**1.2.3. Tema. NoSQL duomenų bazių valdymas.**

*3 užduotis.* PARDUOTUVĖJE ATSIRADO NAUJAS REIKALAVIMAS – PAGAL PREKĖS KODĄ GAUTI PARDUOTUVIŲ, KURIOSE YRA ŠI PREKĖ, SĄRAŠĄ SU PREKĖS VIENETŲ SKAIČIUMI.

Kaip pakeisti duomenų bazės schemą ir darbo su ja užklausas šiam reikalavimui įgyvendinti?

*4 užduotis*. SUMODELIUOKITE PARDUOTUVĖS SCENARIJŲ DOKUMENTŲ DUOMENŲ BAZĖJE.

*5 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS.

1. Išrinkite parduotuvę pagal jos kodą.
2. Išrinkite visas prekes, esančias pasirinktoje parduotuvėje.
3. Išrinkite visas parduotuves, kuriose yra prekė pasirinktu kodu.
4. Gaukite kiekvienoje parduotuvėje esančių prekių bendrą kiekį.

*6 užduotis*. SUMODELIUOKITE PARDUOTUVĖS SCENARIJŲ GRAFŲ DUOMENŲ BAZĖJE.

*7 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS.

1. Išrinkite parduotuvę pagal jos kodą.
2. Išrinkite visas prekes, esančias pasirinktoje parduotuvėje.
3. Išrinkite visas parduotuves, kuriose yra prekė pasirinktu kodu.
4. Išrinkite kiekvienoje parduotuvėje esančių prekių bendrą vienetų kiekį.

*8 užduotis*. ATSIRADO PAPILDOMAS REIKALAVIMAS SUSIETI PREKES SU GAMINTOJO INFORMACIJA.

Kiekvienas gamintojas identifikuojamas pagal įmonės pavadinimą, kartu saugoma gamintojo valstybė. Prekės, gaminamos parduotuvėje, nėra susiejamos su gamintoju.

Papildykite duomenų bazės modelį ir parašykite šias užklausas:

1. Išrinkite visas parduotuves, kuriose yra Vokietijos gamintojų prekės.
2. Išrinkite visas parduotuves, kuriose yra brangesnių nei 1 euras nurodyto gamintojo prekių.

**2. Kompetencija. Programiškai įgyvendinti ir administruoti duomenų bazes.**

**2.1. Mokymosi rezultatas. Diegti ir valdyti duomenų bazių valdymo sistemą.**

**2.1.1. Tema. DBVS diegimas.**

*1 užduotis*. PRISIJUNKITE PRIE DUOMENŲ BAZIŲ VALDYMO SISTEMOS NAUDODAMI TELNET .

*2 užduotis*. PAKARTOKITE TUOS PAČIUS VEIKSMUS NAUDODAMI PHPMYADMIN.

**2.2. Mokymosi rezultatas. Naudoti SQL kalbą duomenų bazės užpildymui ir informacijos išrinkimui.**

* 1. 1. **Tema. Duomenų išrinkimas naudojant SQL select sakinį ir pagrindinius select elementus.**

*1 užduotis*. PANAUDOKITE TINKAMAS PAGRINDINES SQL KOMANDAS.

Duota:

darbuotojai.sql

DROP TABLE IF EXISTS `DARBUOTOJAI`;

CREATE TABLE `DARBUOTOJAI` (

`ASMENSKODAS` bigint(20) DEFAULT NULL,

`VARDAS` varchar(20) DEFAULT NULL,

`PAVARDE` varchar(30) DEFAULT NULL,

`DIRBANUO` date DEFAULT NULL,

`GIMIMOMETAI` date DEFAULT NULL,

`PAREIGOS` varchar(30) DEFAULT NULL,

`SKYRIAUSPAVADINIMAS` varchar(30) DEFAULT NULL,

`PROJEKTONUMERIS` int(11) DEFAULT NULL

);

INSERT INTO `DARBUOTOJAI` VALUES (32541036850,'Mantas','Bananas','2010-08-01','1999-04-21','Programuotojas','Amzinai atostogose',2),(35206891026,'Justas','Zmogauskas','2008-02-04','1990-08-22','Programuotojas','Daug dirbantys',2),(35261458702,'Petrius','Kanusauskas','1996-04-18','1976-10-11','Programuotojas','Mazai dirbantieji',1),(36510284592,'Antanas','Smeliauskas','2006-05-04','1986-10-14','Testuotojas','Mazai dirbantys',1),(38962504820,'Zilvinas','Morkinis','2012-11-20','1988-06-15','Programuotojas','Mazai dirbantys',3),(39520146780,'Jonas','Jonauskas','1985-05-25','1960-05-04','Testuotojas','Daug dirbantieji',3),(42056548920,'Toma','Antanaitiene','2004-09-11','1979-05-11','Testuotoja','Daug dirbantys',2),(44205967260,'Juste','Karnisoviene','1988-09-06','1950-07-30','Projektu vadove','Vadovybe',1),(45289645130,'Zinaida','Zidane','1999-04-18','1980-04-01','Programuotoja','Amzinai atostogose',2),(49853148205,'Toma','Zidane','2010-10-06','1990-07-14','Projektu vadove','Vadovybe',2);

Atlikite užklausas importuotoje DB (naudojamos priemonės/aplinkos – SQL DB aplinka):

1. Išrinkite visus duomenis iš lentelės „DARBUOTOJAI“.
2. Išrinkite visus duomenis iš stulpelio „ASMENS KODAS“ lentelėje „DARBUOTOJAI“.
3. Išrinkite visus duomenis iš stulpelių „VARDAS“, „PAVARDĖ“, „PAREIGOS“lentelėje „DARBUOTOJAI“.
4. Išrinkite skirtingas reikšmes iš stulpelio „SKYRIAUSPAVADINIMAS“ lentelėje „DARBUOTOJAI“.
5. Išrinkite visus duomenis apie darbuotojus, kurie dirba „Daug dirbantys“ skyriuje.
6. Išrinkite duomenis, kokias pareigas užima Toma.
7. Išrinkite visus duomenis apie darbuotojus, kurių gimimo data - 1960-05-04.
8. Išrinkite darbuotojų vardus, kurių pavardės yra Morkinis.
9. Išrinkite duomenis (vardą ir pavardę) apie programuotojus iš „Daug dirbantys“ skyriaus.
10. Įterpkite į lentelę „DARBUOTOJAI“ naują darbuotoją, užpildydami visus reikiamus laukus (asmens kodą, vardą, pavardę, nuo kada pradėjo dirbti, gimimo metus, pareigas, skyriaus pavadinimą ir projekto numerį).
11. Įterpkite į lentelę „DARBUOTOJAI“ naują darbuotoją, užpildydami tik laukus: asmens kodą, vardą, pavardę, nuo kada pradėjo dirbti, gimimo metus. Pareigas, skyriaus pavadinimą ir projekto numerį palikite neužpildytus.
12. Užpildykite likusius tuščius laukus „DARBUOTOJAI“ lentelėje, jūsų prieš tai įterptame įraše. Priskirkite darbuotojui pareigas, skyrių ir projektą.
13. Ištrinkite lentelės „DARBUOTOJAI“ įrašą, kurio asmens kodas yra toks, kurį jūs sukūrėte.
14. Įterpkite du darbuotojus pavarde Antanaitis, kurių pareigos būtų „PROGRAMUOTOJAS“
15. Pakeiskite abiejų Antanaičių pareigas į „TESTUOTOJAS“ vienu sakiniu.
16. Suskaičiuokite, kiek įmonėje dirba Testuotojų.
    * 1. **Tema. Duomenų išrinkimas naudojant sąryšius (SQL select su join).**

*2 užduotis*. PANAUDOKITE TINKAMAS SĄLYGŲ IR GRUPAVIMO SQL KOMANDAS.

Duota:

DARBUOTOJAI.sql

DROP TABLE IF EXISTS `DARBUOTOJAI`;

CREATE TABLE `DARBUOTOJAI` (

`ASMENSKODAS` bigint(20) DEFAULT NULL,

`VARDAS` varchar(20) DEFAULT NULL,

`PAVARDE` varchar(30) DEFAULT NULL,

`DIRBANUO` date DEFAULT NULL,

`GIMIMOMETAI` date DEFAULT NULL,

`PAREIGOS` varchar(30) DEFAULT NULL,

`SKYRIAUSPAVADINIMAS` varchar(30) DEFAULT NULL,

`PROJEKTONUMERIS` int(11) DEFAULT NULL

);

INSERT INTO `DARBUOTOJAI` VALUES (32541036850,'Mantas','Bananas','2010-08-01','1999-04-21','Programuotojas','Amzinai atostogose',2),(35206891026,'Justas','Zmogauskas','2008-02-04','1990-08-22','Programuotojas','Daug dirbantys',2),(35261458702,'Petrius','Kanusauskas','1996-04-18','1976-10-11','Programuotojas','Mazai dirbantieji',1),(36510284592,'Antanas','Smeliauskas','2006-05-04','1986-10-14','Testuotojas','Mazai dirbantys',1),(38962504820,'Zilvinas','Morkinis','2012-11-20','1988-06-15','Programuotojas','Mazai dirbantys',3),(39520146780,'Jonas','Jonauskas','1985-05-25','1960-05-04','Testuotojas','Daug dirbantieji',3),(42056548920,'Toma','Antanaitiene','2004-09-11','1979-05-11','Testuotoja','Daug dirbantys',2),(44205967260,'Juste','Karnisoviene','1988-09-06','1950-07-30','Projektu vadove','Vadovybe',1),(45289645130,'Zinaida','Zidane','1999-04-18','1980-04-01','Programuotoja','Amzinai atostogose',2),(49853148205,'Toma','Zidane','2010-10-06','1990-07-14','Projektu vadove','Vadovybe',2);

ATLIKITE UŽKLAUSAS IMPORTUOTOJE DB (NAUDOJAMOS PRIEMONĖS/APLINKOS – SQL DB APLINKA):

1. Išrinkite duomenis apie darbuotoją (asmens kodą, vardą ir pavardę) iš lentelės DARBUOTOJAI, kurie būtų gimę 1988m. birželio 15d.
2. Išrinkite visus duomenis apie darbuotojus iš lentelės DARBUOTOJAI, kurie yra gimę iki 1988m. liepos 29d.
3. Išrinkite duomenis apie darbuotojus (dirba nuo kada ir gimimo metus) iš lentelės DARBUOTOJAI, kurie būtų įsidarbinę nuo 2000m. spalio 30d. iki 2012m. lapkričio 11d.
4. Išrinkite duomenis apie darbuotojus (vardą, skyrių ir projekto numerį) iš lentelės DARBUOTOJAI kurie dirba 2 ir 3 projektuose. (Panaudoti IN operatorių).
5. Išrinkite duomenis (vardą, pavardę ir asmens kodą) apie visas moteris iš lentelės DARBUOTOJAI (panaudojant operatorių LIKE).
6. Išrinkite visus duomenis apie visus darbuotojus iš lentelės DARBUOTOJAI, kurie yra gimę 12 dieną (panaudojant operatorių LIKE).
7. Išrinkite visus duomenis iš lentelės DARBUOTOJAI, kad skyriaus pavadinime 3 raidė būtų ‘u’.
8. Išrinkite visus darbuotojus iš lentelės DARBUOTOJAI, kuriems nepaskirtos jokios pareigos.
9. Išrinkite duomenis apie darbuotoją (vardą, pavardę, nuo kada dirba ir pareigas) kad tenkintų sąlygas: (dirba nuo 2010-08-01 ir jų pareigos yra Programuotojas).
10. Išrinkite duomenis apie darbuotojus (vardą, pavardę, skyriaus pavadinimą ir projekto numerį) iš lentelės DARBUOTOJAI su sąlyga, kad jie būtų iš Mažai dirbantys skyriaus arba 1 projekto.
11. Išrinkite visus darbuotojų vardus, išskyrus tuos, kurių vardai prasideda raide ‚J’ .
12. Išrinkite duomenis (vardą, dirba nuo kada ir gimimo metus) iš lentelės „DARBUOTOJAI” apie visus darbuotojus, tik ne tuos, kurie įsidarbino nuo 2009 m. spalio 30 d. iki 2012 m. lapkričio 11d.
13. Išrinkite duomenis apie darbuotojus (vardą, pavardę ir gimimo metus) iš lentelės DARBUOTOJAI ir išrikiuokite visus duomenis nuo seniausio žmogaus iki jauniausio.
14. Išrinkite duomenis apie darbuotojus (vardą, pavardę ir gimimo metus) iš lentelės DARBUOTOJAI ir išrikiuokite visus duomenis nuo jauniausio žmogaus iki seniausio.
15. Išrinkite iš lentelės DARBUOTOJAI projekto numerį, kuris būtų minimalus skaičius ir maksimalus skaičius.
16. Išrinkite duomenis apie projektą ir kiek tame projekte yra priskirta žmonių iš lentelės DARBUOTOJAI (projekto numeris ir skaičius kiek dalyvauja žmonių).
17. Išrinkite duomens (projekto numeris, pareigos, skaičius) iš lentėlės DARBUOTOJAI kiek dirba programuotojų kiekvienam projekte?

#17 punkto užklausą pataisykite taip, kad rodytų tik tuos projektus, kur dirba bent 3 darbuotojai.

**Modulis „Programavimo aplinkos ir kūrimo proceso valdymas (JavaSript)“**

1. **Kompetencija. Naudoti tarnybinių stočių operacines sistemas.**
   1. **Mokymosi rezultatas. Suprasti serverių veikimo principus ir panaudojimo galimybes.**
      1. **Tema. Serveriai ir jų operacinės sistemos.**

*1 užduotis.* DEBESŲ KOMPIUTERIJA.

UŽDUOTIES SĄLYGA: suraskite informaciją apie debesų kompiuteriją, atsakykite kas yra VDS serveris. Patikrinti, kaip serverių resursai įtakoja serverio kainą.

Naudojamos priemonės (aplinkos): interneto naršyklė (Firefox, Opera ir pan.).

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato: naudojant https://www.google.lt informacijos paieškos įrankį ieškoma tikslinė informacija, raktiniai žodžiai: debesų kompiuterija, VDS serveris.

*2 užduotis.* SERVERIO IŠTEKLIAI.

UŽDUOTIES SĄLYGA: naudojant https://www.google.lt informacijos paieškos įrankį suraskite informaciją apie WordPress turinio valdymo sistemos reikalavimus serveriui, parinkti serverį, kuris galės talpinti WordPress turinio valdymo sistemą.

Naudojamos priemonės (aplinkos): interneto naršyklė (Firefox, Opera ir pan.).

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato: naudojant https://www.google.lt informacijos paieškos įrankį ieškoma tikslinė informacija, raktiniai žodžiai: WordPress requirements.

*3 užduotis.* LINUX OS PARINKIMAS.

UŽDUOTIES SĄLYGA: naudojant https://distrochooser.de/en parinkite sau tinkamą Linux OS.

Naudojamos priemonės (aplinkos): inter.neto naršyklė (Firefox, Opera ir pan.).

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato: tinklapyje https://distrochooser.de/en atlikti testą ir nustatyti kokia Linux OS Jums tinka geriausiai.

* + 1. **Tema. Serverio operacinės sistemos diegimas.**

*4 užduotis.* XUBUNTU OS DIEGIMAS VIRTUALIOJE APLINKOJE.

UŽDUOTIES SĄLYGA: naudojant VirtualBox įrankį ir Xubuntu OS iso atvaizdą virtualizuokite Linux OS.

Naudojamos priemonės (aplinkos): Windows 7, VirtualBox.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

Darbas vykdomas Windows OS aplinkoje. Konfigūruojame VirtualBox įrankį pagal pateiktus parametrus:

1. Virtualios mašinos pavadinimas: VardasPavarde

2. Virtualios mašinos tipas: Linux

3. Virtualios mašinos versija: Ubuntu (64 bit)

4. Virtualios mašinos RAM dydis: 2GB

5. Virtualios mašinos HDD dydis: 10GB

6. Virtualios mašinos tinklo konfigūracija: Bridged Adapter

Diegiame GNU/Linux OS (Xubuntu) su pateiktais parametrais:

1. Vartotojo vardas: VardasPavarde

2. Vartotojo slaptažodis: studentas

3. Serverio vardas studentas

Konfigūruojame GNU/Linux OS (Xubuntu) su pateiktais reikalavimais:

1.Virtualioje sistemoje turi būti įdiegtas VirtualBox Guest Additions

2. Virtualioje OS turi būti aktyvuota administratoriaus paskyra (root).

**1.1.3. Tema. Bazinis serverio operacinės sistemos funkcionalumas.**

*5 užduotis.* XUBUNTU OS KONFIGŪRAVIMAS.

UŽDUOTIES SĄLYGA: naudojant Xubuntu grafinius konfigūracijos įrankius sukonfigūruokite klaviatūros išdėstymą ir Xubuntu grafinės aplinkos išvaizdą.

Naudojamos priemonės (aplinkos): Xubuntu OS.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato: naudojant Xubuntu Settings Manager konfigūruojame Keyboard ir Apperance.

* 1. **Mokymosi rezultatas.Valdyti programinius paketus.**

**1.2.1. Tema. Programiniai paketai.**

* + 1. *užduotis*. LINUX OS PAKETŲ ADMINISTRAVIMAS.

UŽDUOTIES SĄLYGA: naudojant komandų eilutę arba grafinius įrankius atnaujinkiteLinux OS ir įdiekite naujus programinius paketus.

Naudojamos priemonės (aplinkos): komandų eilutė, aptitude, Ubuntu software center.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Su apt-cache įrankiu suraskite programinį paketą xcowsay.

2. Su apt-get įrankių įdiekite paketą į sistemą.

3. Naudojant APT įrankį (GUI arba komandinės eilutės) įdiekite programą figlet.

4. Išbandykite programų veikimą.

5. Nueikite į https://launchpad.net ir suraskite „Oracle Java (JDK) 7 / 8 / 9 Installer PPA“

6. Pridėkite prie sistemos PPA: webupd8team/java

7. Įdiekite į sistemą programinius paketus: eclipse, oracle-java8-installer.

* 1. **Mokymosi rezultatas. Naudoti Web serverio programinę įrangą HTTP bylų viešinimui.**

**1.3.1. Tema. Web serverio aplinka operacinėje sistemoje.**

*1 užduotis.* WEB SERVERIO DIEGIMAS IR KONFIGŪRAVIMAS.

UŽDUOTIES SĄLYGA: naudojant komandų eilutę įdiekite ir sukonfigūruokite WEB serverį.

Naudojamos priemonės (aplinkos): komandų eilutė.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Paruoškite WEB serverį darbui:
   1. Įdiekite WEB serverį Apache.
   2. Paleiskite WEB serverį Apache.
   3. Patikrinkite jo veikimą.
   4. Jei reikia, WEB serverį sukonfigūruokite.
2. Sukonfigūruokite WEB serverį taip, kad jis galėtų talpinti 2 svetaines:
   1. 1-ma svetainė: linux.org. Svetainė turi talpinti tekstą: „Sveiki aš esu Linux OS!“ ir turėti nuorodą į svetainę windows.org (index.html – naudojame tik HTML ir CSS)
   2. 2-ra svetainė: windows.org. Svetainė turi talpinti tekstą: „Sveiki aš esu Windows OS!“ ir turėti nuorodą į svetainę linux.org (index.html – naudojame tik HTML ir CSS)
3. Paleiskite abi svetaines ir patikrinkite jų veikimą.
4. Įjungite savo WEB serveryje PHP palaikymą, parašykite trumpą php testą (<?php phpinfo();?>) ir patikrinkite ar PHP palaikymas veikia.
5. **Kompetencija. Taikyti aktualias programinės įrangos kūrimo metodikas.**
   1. **Mokymosi rezultatas. Suprasti SCRUM proceso dalis ir komandos narių atsakomybes.**
      1. **Tema. Scrum procesas.**

*1 užduotis.*KAS YRA SPRINTO ĮSIPAREIGOJIMŲ SAVININKAS**?**

 Atskiri komandos nariai, paskirti Scrum meistro (ScrumMaster)

 Visa komanda kartu

 Scrum meistras

 Atskiri komandos nariai, pagal susitarimą sprinto planavimo susitikime

*2 užduotis.*DAUG ŽMONIŲ MANO, KAD PROGRAMAVIMAS POROMIS (PAIR PROGRAMMING) MAŽINA KLAIDŲ SKAIČIŲ IR PALENGVINA KODO PRIEŽIŪRĄ.

Programavimas poromis - kas tai?

 Vienas komandos narys pateikia kodą, kad vėliau kitas galėtų jį peržiūrėti

 Du žmonės dalinasi viena darbo vieta (kompiuteriu) paprastai pasikeisdami paeiliui renka kodą ar atlieka veiksmus klaviatūra, kitas stebi, atkreipia dėmesį ir padeda pirmajam

 Kodas vienu metu rašomas dviejose eilutėse kad sumažinti klaidų skaičių

*3 užduotis.*KOKS YRA REKOMENDUOJAMAS SCRUM KOMANDOS DYDIS?

 Nesvarbu, jei tik komandos nariai gali užtikrinti visas reikalingas funkcijas (doesn't matter, as long as the team is cross functional)

 9, plius/ minus 3 nariai

 10, plius/ minus 3 nariai, ar daugiau, jei komanda geografiškai nutolusi

 7 plius/ minus 2 nariai

*4 užduotis.* KOKIA KASDIENIO SCRUM SUSITIKIMO (DAILY SCRUM MEETING) TRUKMĖ (TIME-BOX)?

 5 minutės

 10 minučių

 15 minučių

 Tiek kiek reikės

*5 užduotis.* KASATSAKO UŽ ĮRANKIŲ PASIRINKIMĄ IR KONFIGŪRAVIMĄ ORGANIZACIJOJE, KURI PRIPAŽĮSTA AGILE VERTYBES**?**

 Komandos, kurios turėtų susiderinti tarpusavyje

 Scrum meistrai (ScrumMasters), kurie turėtų susiderinti tarpusavyje

* 1. **Mokymosi rezultatas. Analizuoti pateiktus reikalavimus, nustatant programos atitikimą reikalavimams.**

### Tema. Programinės įrangos reikalavimų analizė.

*1 užduotis.*KAS VYKSTA PROGRAMINĖS ĮRANGOS KŪRIMO CIKLO (angl. SOFTWARE DEVELOPMENT LIFECYCLE, SDLC) REIKALAVIMŲ SURINKIMO IR ANALIZĖS ETAPE?

Klientas sumoka už numatytą darbų apimtį

Klientas išsako savo lūkesčius projekte

Projekto komanda realizuoja kiekvieną reikalavimą programos kode

 Projekto vadovas samdo projekto komandą

*2 užduotis*. REIKALAVIMŲ ANALIZĖ NEAPIMA JŲ ATSEKAMUMO.

  Taip, neapima

  Ne, apima

*3 užduotis.* REIKALAVIMŲ ANALIZĖ YRA ITERATYVUS PROCESAS.

  Taip, yra

  Ne, nėra

*4 užduotis.*KAIP ORGANIZUOJAMAS PRODUKTO DARBŲ SĄRAŠAS (angl. PRODUCT BACKLOG)?

  Darbai sąraše išdėstomi atsitiktine tvarka

  Kategorijomis pagal prioritetą - P1, P2, P3 ir t.t.

 Dideli darbai sąrašo pradžioje, smulkūs gale

 Svarbiausi darbai sąrašo pradžioje, mažiausiai svarbūs gale

*5 užduotis*. KADA VYKSTANT SPRINTUI GALIMA PRIDĖTI NAUJAS SPRINTO UŽDUOTIS?

  Kai produkto savininkas (product owner) identifikuoja naują užduotį

  Kai Scrum meistras (Scrum Master) identifikuoja naują užduotį

  Niekada. Sprinto užduotys yra fiksuojamos ir apsprendžiamos sprinto planavimo metu

  Kuo greičiau po to kai jos identifikuojamos, jei jos nekeičia darbo apimčių (scope change) siekiant užsibrėžtų sprinto tikslų (sprint goals)

  Kai Scrum meistras (Scrum Master) jas patvirtina

*6 užduotis.* KAIP VERTINTUMĖTE TOKĮ VARTOTOJO PASAKOJIMĄ?

„Kaip pardavimo agentas, aš noriu turėti klientų paieškos galimybę kad galėčiau rasti savo klientus greitai ir lengvai“

  Gerai suformuluotas. Nereikia papildomos informacijos

  Gerai suformuluotas. Reiktų paaiškinimo, ką reiškia “greitai ir lengvai” kad būtų galima testuoti vartotojo sąsają

  Blogai suformuluotas. Pernelyg didelės apimties (per platus)

  Blogai suformuluotas. Pernelyg mažos apimties (per siauras)

* 1. **Mokymosi rezultatas. Naudoti projekto eigos valdymo principus.**
     1. **Tema. Projekto eigos valdymas.**

*1 užduotis.*KOKS SPRINTO UŽDUOTIES DYDIS LAIKOMAS TINKAMU?

  Viena žmogaus diena arba mažiau, kad kiti komandos nariai galėtų lengvai pastebėti, jei užduoties įgyvendinimas užstrigo

  2-3 žmonių 2-3 dienos, kad kiekvienas produkto darbų sąrašo (angl. backlog) įrašas būtų viena sprinto užduotis

*2 užduotis****.*** KAIP DAŽNAI TURI VYKTI PRODUKTO DARBŲ SĄRAŠO PERŽIŪRA (angl. BACKLOG GROOMING)?

 Vieną kartą, projekto pradžioje

 Vieną kartą, projekto pabaigoje

 Kiekvieno sprinto metu

 Kiekvieno kodo atnaujinimo ciklo (angl. release cycle) metu

*3 užduotis.*KĄ SCRUM KOMANDA TURĖTŲ PADARYTI PIRMO SPRINTO METU?(galimi keli teisingi atsakymai)

 Testuoti produktą

 Sukurti detalų projekto planą

 Užfiksuoti produkto architektūrą

 Sukurti nedidelį gabaliuką veikiančio funkcionalumo (angl. potentially shippable functionality)

*4 užduotis.*ĮMONĖS VADOVAS PAPRAŠO KOMANDOS NARIO ATLIKTI DARBĄ, KURIS NEĮEINA Į VYKSTANČIO SPRINTO NUMATYTĄ APIMTĮ.

Ką tokiu atveju turėtų daryti komandos narys?

 Pridėti jį prie sekančio sprinto darbų

 Pridėti jį prie vykstančio sprinto darbų pakeičiant juo dalį jau prisiimto analogiškos apimties darbo

 Informuoti Produkto savininką, kad jis galėtų aptarti tai su įmonės vadovu

 Pridėti jį prie vykstančio sprinto darbų

*5 užduotis.*KADA SPRINTO VYKDYMAS BAIGIAMAS?

 Kai baigiamos visos užduotys

 Kai visos prisiimtų užduočių (all committed product backlog items) rezultatai atitinka baigtumo kriterijus (meet their definition of done)

 Priklauso nuo situacijos

 Kai baigiasi sprintui skirtas laikas (time-box expires)

1. **Kompetencija. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą.**
   1. **Mokymosi rezultatas. Diegti ir valdyti programavimo JavaScript kalba darbo aplinką.**
      1. **Tema. JavaScript programavimo aplinka.**

*1 užduotis*. PAAIŠKINKITE, KOKIE YRA NETBEANS IDE NAUDOJIMO PRIVALUMAI DIRBANT SU JAVASCRIPT FAILAIS?

*2 užduotis*. ĮDIEKITE NETBEANS IDE APLINKĄ.

* + 1. **Tema. IDE naudojimas projektuose.**

*3 užduotis*. SUKURKITE NAUJĄ NETBEANS IDE PROJEKTĄ JAVA PROGRAMAI.

Projekto pavadinimas – „FIRSTAPPLICATION“, pagrindinės klasės pavadinimas – „FIRSTAPPMAIN.FIRSTAPPLICATION“.

*4 užduotis*. KAIP ATSISIŲSTI KOMANDOS PROJEKTĄ IŠ „GITHUB“ Į NETBEANS IDE?

* 1. **Mokymosi rezultatas. Sekti programavimo darbų vykdymą naudojant komandinio darbų planavimo sistemas.**

### 3.2.1. Tema. Komandinio darbų planavimo sistemos.

*1 užduotis*. UŽRAŠYKITE PAGRINDINIUS PROGRAMINĖS ĮRANGOS KŪRIMO ETAPUS.

*2 užduotis.* APRAŠYKITE PROGRAMINĖS ĮRANGOS GYVAVIMO CIKLĄ.

*3 užduotis*. UŽRAŠYKITE TRIS KOMANDINIO DARBŲ PLANAVIMO SISTEMAS.

*4 užduotis.* APRAŠYKITE JIRA FUNKCIONALUMĄ IR PANAUDOJIMO GALIMYBES KURIANT IR TESTUOJANT PROGRAMINĘ ĮRANGĄ.

*5 užduotis.* APRAŠYKITE PAGRINDINIUS JIRA SISTEMOS APLINKOS ELEMENTUS PAPRASTAM VARTOTOJUI.

### Tema. Programavimo darbų vykdymo sekimas.

*6 užduotis.* APRAŠYKITE KAIP YRA VYKDOMAS PROJEKTINIS DARBAS JIRA DARBŲ PLANAVIMO SISTEMOJE.

*7 užduotis*. APRAŠYKITE KAIP YRA PLANUOJAMOS, SUKURIAMOS, APRAŠOMOS UŽDUOTYS JIRA DARBŲ PLANAVIMO SISTEMOJE.

*8 užduotis.* APRAŠYKITE KAIP YRA PRISKIRIAMI DARBAI (UŽDUOTYS) ATSKIRIEMS VARTOTOJAMS.

Papildomai aprašykite kaip užduotys yra komentuojamos.

*9 užduotis.* APRAŠYKITE KAIP YRA UŽBAIGIAMI DARBAI (UŽDUOTYS) JIRA DARBŲ PLANAVIMO SISTEMOJE.

Papildomai aprašykite kaip atrodo TODO sąrašas.

*10 užduotis*. APRAŠYKITE KAIP YRA VYKDOMA PAIEŠKA JIRA DARBŲ PLANAVIMO SISTEMOJE.

**Modulis „Įvadas į darbo rinką“**

## 

## TESTAS ĮSIVERTINTI GEBĖJIMAMS BAIGUS PROGRAMĄ

1. Koks skaičius yra užrašytas dvejetainėje sistemoje 1010?

a) 101

b) 10

c) 11

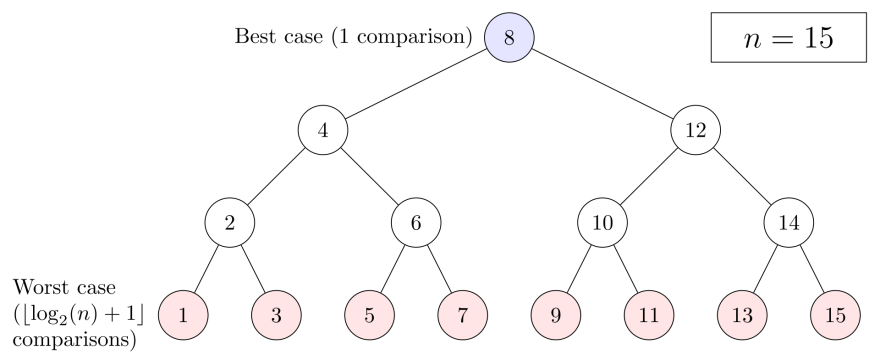
1. Koks bus šio loginio veiksmo rezultatas: 1010 & 1101?

a) 1000

b) 1111

c) 0111

1. Koks algoritmas pavaizduotas?

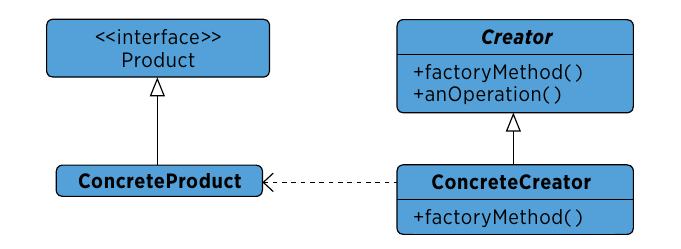


a) nuoseklios paieškos

b) dvejetainės paieškos

c) linijinės paieškos

1. Koks šablonas yra pavaizduotas?

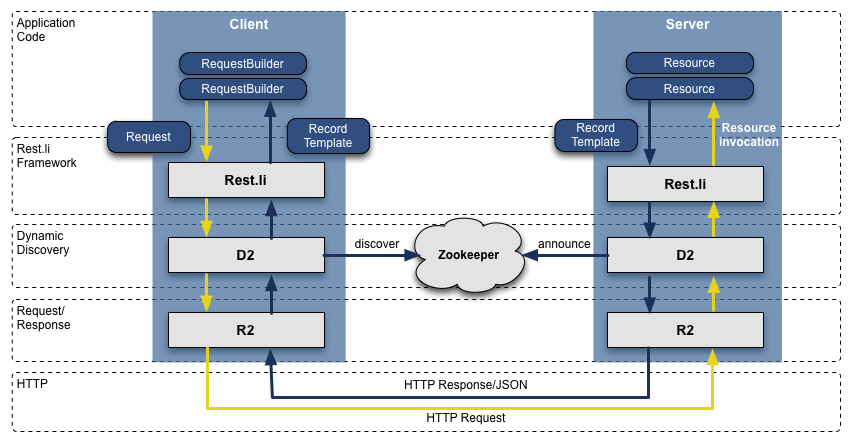


a) Singleton

b) Factory Method

c) Observer

1. Kokia architektūra yra pavaizduota?



a) Kliento - Serverio

b) Modelio - Vaizdo - Kontrolerio

c) Brokerio

1. JavaScript tai:

a) Java programavimo kalbos dalis

b) sala

c) savarankiška programavimo kalba

1. Inkapsuliacija tai:

a) galimybė programuoti nežinant konkrečių detalių

b) informacijos slėpimas

c) objektų organizavimas specializuojant egzistuojančius bendresnius tipus

1. Kokį rezultatą gausime įvykdę žemiau nurodytą kodą:

var x = 0;

function Vartotojas() {

    x++;

    this.x = x;

    return x;

}

var vartotojas = new Vartotojas;

console.log(vartotojas.x);

a) undefined

b) 2

c) TypeError

d) ReferenceError

1. Ar galima naudoti anoniminę funkciją, kaip argumentą kitoje funkcijoje ?

a) taip

b) ne

1. Kurie iš pateiktų kodo fragmentų išves asmens vardą ir pavardę.

1)

let Asmuo = function(vardas, pavarde, amzius)

{

  this.vardas = vardas;

  this.pavarde = pavarde;

  this.amzius = amzius;

}

asmuo = new Asmuo('Vardenis', ' Pavardenis', 26);

Asmuo.prototype.pasisveikink = function() {

  console.log("Labas "+ this.vardas + " " + this.pavarde)

};

asmuo.pasisveikink()

2)

let Asmuo = function(vardas, pavarde, amzius)

{

  this.vardas = vardas;

  this.pavarde = pavarde;

  this.amzius = amzius;

}

asmuo = new Asmuo('Vardenis', ' Pavardenis', 26);

Asmuo.pasisveikink = function() {

  console.log("Labas "+ this.vardas + " " + this.pavarde)

};

asmuo.pasisveikink()

3)

let asmuo = {

  vardas: "Vardenis",

  pavarde: "Pavardenis",

  amzius: 26

}

asmuo.pasisveikink = function() {

  console.log("Labas "+ this.vardas + " " + this.pavarde)

};

asmuo.pasisveikink()

4)

let asmuo = {

  vardas: "Vardenis",

  pavarde: "Pavardenis",

  amzius: 26

}

asmuo.prototype.pasisveikink = function() {

  console.log("Labas "+ this.vardas + " " + this.pavarde)

};

asmuo.pasisveikink()

a) 1

b) 2 ir 3

c) 1 ir 4

d) 1 ir 3

1. Kas bus išvesta įvykdžius žemiau pateiktą kodą ?

(function() {

  return [

    (() => this.x).bind({ x: 'pokytis' })(),

    (() => this.x)()

  ]

}).call({ x: 'poslinkis' });

a) ['pokytis', 'poslinkis']

b) ['poslinkis', 'poslinkis]

c) ['undefined', 'undefined]

1. Ar žemiau pateiktas kodas atitinka  HOC architektūros keliamus reikalavimus ?

import React from "react";

import { render } from "react-dom";

function withHiddenProp(WrappedComponent) {

  return class extends React.Component {

    render() {

      if (this.props.hidden) {

        return null;

      } else {

        return <WrappedComponent {...this.props} />;

      }

    }

  };

}

class HelloWorld extends React.Component {

  render() {

    return <h1>Hello world</h1>;

  }

}

const HelloWorldWithHidden = withHiddenProp(HelloWorld);

render(

  <HelloWorldWithHidden hidden={false} />,

  document.getElementById("root")

);

a) taip

b) ne

1. Paaiškinkite Node.js blokuojančius ir neblokuojančius procesų tipus.
2. Sukurkite Node.js HTTP Web serverį, kuris atitiktų šiuos reikalavimus:

* Naudotų 8080 prievadą (angl. „port“)
* Nurodžius serverio kelią (pvz.: [*http://127.0.0.1*](http://127.0.0.1)*:8080/?action=insert&brand=Volvo&color=white* ) vartotojas turi matyti kaip rezultatą tokį turinį „1 car inserted to database“; Taip pat turi būti sukurtas įrašas duomenų bazėje „Cars“, su savybe „brand“ ir savybe „color“.
* Parametrų „brand“ ir „color“ reikšmes turi būti galima keisti užklausoje;
* Nurodžius serverio kelią, kuriame nėra „action=insert“ parametro arba nėra „action=find“ parametro (pvz: [*http://127.0.0.1:8080*/](http://127.0.0.1:8080/)), turi būti sugražintas rezultatas: „Bad request parameters“.
* Nurodžius serverio kelią su „action=find&color=white“ parametrais (pvz. [*http://127.0.0.1*](http://127.0.0.1)*:8080/?action=find&color=white*), serveris turi gražinti sąrašą įrašų iš duomenų bazės, su automobilių savybėmis „brand“ ir „color“, kur „color“ atitinka užklausoje nurodytą parametrą.

1. Sukurkite Node.js programą (panaudodami Mongoose modulį) tam, kad prisijungtumėte prie MongoDB duomenų bazės ir surastumėte keletą objektų, kurių vieno iš parametrų skaitinė reikšmė yra tarp X ir Y dydžių. Pavyzdžiui, galite prisijungti prie sukurtos duomenų bazės „Cars“, kurioje saugomi objektai turintys savybę „gas“, ir surasti automobilius, kurių kuro kiekis „gas“ yra tarp 20 ir 55.
2. TDD tai:

a) laiko padalijimo dvipusis ryšys (Time Division Duplex)

b) techninio projektavimo direktyva (Technical Design Directive)

c) testais valdomas programinės įrangos kūrimas (Test Driven Development)

1. Kuri GIT komanda parodo visus užfiksuotus pakeitimus?

a) git commit log

b) git log

c) git stash

1. Kiek šakų gali būti Git repozitorijoje?

a) 20

b) priklauso nuo kompiuterio resursų

c) kiek norime

1. Kas yra komandos sparta (Team Velocity)?

a) tai vidutinis darbo apimties kiekis kurį komanda užbaigia pilnai

b) tai maksimalus darbo apimties kiekis kurį komanda užbaigia pilnai

c) tai minimalus darbo apimties kiekis kurį komanda užbaigia pilnai

1. Koks tai yra reikalavimas: „Pardavėjas gali išrašyti išankstinę sąskaitą faktūrą“?

a) nefunkcinis

b) funkcinis

c) sisteminis

# Užduočių atsakymai

**Modulis „Įvadas į profesiją“**

## TESTAS ĮSIVERTINTI GEBĖJIMAMS PRIEŠ PRADEDANT MOKYTIS

Testo atsakymai:

1. b)

2. c)

3. b)

4. c)

5. b)

6. c)

7. b)

8. b)

9. c)

10. b)

11. b)

12. b)

13. c)

14. c)

**Modulis „Informacinių sistemų projektavimas ir kūrimas (JavaScript)“**

* 1. **Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.**
  2. **Mokymosi rezultatas. Pateikti internetinio puslapio turinį naudojant kompiuterinę žymėjimo kalbą.**

**1.1.1. Tema. Pagrindinės HTML kalbos žymės.**

*1 užduotis****.*** PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (STRAIPSNIS).

Rezultatai:

1. index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>HTML praktinė užduotis Nr. 1</title>

</head>

<body>

<h3>boldas, pusjuodis, pastorintasis šriftas</h3>

<h2>Ar vartotinas žodis „boldas“?</h2>

<p><i>Boldas</i> – nevartotina svetimybė. </p>

<p>V. Dagienės, G. Grigo, T. Jevsikovos <a href="#">„Enciklopediniame kompiuterijos žodyne“</a> (Vilnius, 2008, p. 389) teikiamas terminas <b>pusjuodis</b>, angl. <i>bold</i>. Tai šrifto stilius, kai rašmenys, norint juos paryškinti, piešiami pastorintomis linijomis.</p>

<p>Kaip nevartotina svetimybė <i>boldas</i> vertinamas „Kalbos patarimuose“, siūlomi pakaitai: 1. <b>pusjuodis</b>; 2. <b>pastorintasis</b> (šriftas) (žr.: Kalbos patarimai. Kn. 4: Leksika: 1. Skolinių vartojimas, Vilnius, 2013, p. 19; pataisytas ir papildytas leidimas, Vilnius, 2005, p. 33).</p>

<p>Žymės: <a href="#">kompiuterija</a>, <a href="">leksika</a>, <a href="">neteiktina svetimybė</a></p>

</body>

</html>

**1.1.2. Tema. HTML kalbos turinio žymės.**

*2 užduotis.* PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (KELIONIŲ AGENTŪRA).

Rezultatai:

1. Index.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

<meta name="author" content="Kelionės" />

<meta name="keywords" content="HTML, kelionės" />

<meta name="description" content="TOP Kelionės" />

</head>

<body>

<h2>Kelionės</h2>

<ul>

<li><a href="#">Paskutinė minutė</a></li>

<li><a href="page/poilsines.html">Poilsinės kelionės</a></li>

<li><a href="page/datos.html">Kelionių datos</a></li>

</ul>

<h3>Paskutinė minutė</h3>

<h4>Ispanija, Kosta Dorada</h4>

<img src="img/barcelona.jpg" alt="paveiksliukas" width="300">

<p>Paskutinės minutės atostogos Ispanijoje, Salou! Skrydis iš Vilniaus, <br /> bagažas, pervežimai ir 7 n. viešbutyje su pusryčiais ir vakarienėmis - 379 &euro;!</p>

<h3>Išvykimo laikai</h3>

<ul>

<li>Vilnius - Barselona 14:00 - 16:30</li>

<li>Barselona - Vilnius 17:30 - 22:00</li>

</ul>

<h3>Į kelionės kainą įskaičiuota:</h3>

<ol>

<li>Skrydis Vilnius - Barselona - Vilnius;</li>

<li>Pervežimas oro uostas - viešbutis - oro uostas;</li>

<li>20 kg registruojamas ir rankinis bagažas;</li>

<li>7 n. apgyvendinimas pasirinktame viešbutyje su nurodytu <br /> maitinimo tipu.</li>

</ol>

</body>

</html>

2. Aplankas „Pages“

2.1. Datos.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

<meta name="author" content="Kelionės" />

<meta name="keywords" content="HTML, kelionės" />

<meta name="description" content="TOP Kelionės" />

<style>

table,

td,

th {

border: 1px solid grey;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Kelionės</h2>

<ul>

<li><a href="../index.html">Paskutinė minutė</a></li>

<li><a href="poilsines.html">Poilsinės kelionės</a></li>

<li><a href="datos.html">Kelionių datos</a></li>

</ul>

<h3>Kelionių datos</h3>

<table>

<tr>

<th>Data</th>

<th>Kryptis</th>

<th>Kaina</th>

</tr>

<tr>

<td>Rugsėjo 18 - Rugsėjo 28</td>

<td>Turkija</td>

<td>345 &euro;</td>

</tr>

<tr>

<td>Rugsėjo 20 - Rugsėjo 25</td>

<td>Kipras</td>

<td>299 &euro;</td>

</tr>

</table>

</body>

</html>

2.2. Plačiau.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

<meta name="author" content="Kelionės" />

<meta name="keywords" content="HTML, kelionės" />

<meta name="description" content="TOP Kelionės" />

</head>

<body>

<h2>Kelionės</h2>

<ul>

<li><a href="../index.html">Paskutinė minutė</a></li>

<li><a href="poilsines.html">Poilsinės kelionės</a></li>

<li><a href="datos.html">Kelionių datos</a></li>

</ul>

<h3>Polsinės kelionės</h3>

<h4>Ispanija, Kosta Dorada (Plačiau)</h4>

<img src="../img/barcelona.jpg" alt="paveiksliukas" width="300">

<p>Paskutinės minutės atostogos Ispanijoje, Salou! Skrydis iš Vilniaus, <br />

bagažas, pervežimai ir 7 n. viešbutyje su pusryčiais ir vakarienėmis - 379 &euro;!</p>

<h3>Išvykimo laikai</h3>

<ul>

<li>Vilnius - Barselona 14:00 - 16:30</li>

<li>Barselona - Vilnius 17:30 - 22:00</li>

</ul>

<h3>Į kelionės kainą įskaičiuota:</h3>

<ol>

<li>Skrydis Vilnius - Barselona - Vilnius;</li>

<li>Pervežimas oro uostas - viešbutis - oro uostas;</li>

<li>20 kg registruojamas ir rankinis bagažas;</li>

<li>7 n. apgyvendinimas pasirinktame viešbutyje su nurodytu <br />

maitinimo tipu.</li>

</ol>

<a href="poilsines.html">Grįžti</a>

</body>

</html>

2.3. Poilsinės.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

<meta name="author" content="Kelionės" />

<meta name="keywords" content="HTML, kelionės" />

<meta name="description" content="TOP Kelionės" />

</head>

<body>

<h2>Kelionės</h2>

<ul>

<li><a href="../index.html">Paskutinė minutė</a></li>

<li><a href="poilsines.html">Poilsinės kelionės</a></li>

<li><a href="datos.html">Kelionių datos</a></li>

</ul>

<h3>Poilsinės kelionės</h3>

<h4>Ispanija, Kosta Dorada</h4>

<img src="../img/barcelona.jpg" alt="paveiksliukas" width="300">

<p>Paskutinės minutės atostogos Ispanijoje, Salou! Skrydis iš Vilniaus, <br />

bagažas, pervežimai ir 7 n. viešbutyje su pusryčiais ir vakarienėmis - 379 &euro;! <a href="placiau.html">Plačiau...</a></p><br /><br />

<h4>Kreta</h4>

<img src="../img/kreta.jpg" alt="paveiksliukas" width="300">

<p>Rugsėjį keliauk į vaizdingąją Kretą! Skrydis iš Vilniaus, bagažas, pervežimai ir 7 n.<br /> viešbutyje su „viskas įskaičiuota“ – 499 &euro;! <a href="placiau.html">Plačiau...</a></p>

</body>

</html>

*3 užduotis.* PATEIKITE TURINĮ HTML ŽYMĖJIMO KALBA (NAUJIENŲ SRAUTAS).

Rezultatai:

1. Index.html

<!DOCTYPE HTML>

<html lang="lt">

<head>

<meta charset="utf-8" />

<title>Prisiminimų dirbtuvės "Scrap"</title>

<meta name="keywords" content="praktika" />

</head>

<body>

<h1 id="top">Prisiminimų dirbtuvės "Scrap"</h1>

<h2>Pagrindinės temos</h2>

<ul>

<li><a href="#apie">Kas yra skrebinimas ir kokia jo vertė</a></li>

<li><a href="#rusys">Skrebinimo rūšys</a></li>

<li><a href="#irankiai">Skrebinimo įrankiai</a></li>

</ul>

<h2 id="apie">Kas yra skrebinimas ir kokia jo vertė</h2>

<p>Skrebinimas - įdomi veikla. Vienus ji įtraukia, kitiems atrodo beprasmiška. Mėgstantis popieriaus spalvų, faktūrų ir ramaus kūrimo teikiamą džiaugsmą, tikėtina, išbandęs skrebinimą nebegalės sustoti, nes pats procesas įtraukia taip, jog galima paskęsti jame ilgoms valandoms. Skrebinimo tikslas - įamžinti akimirkas, kad jos liktų prisiminimu, o ne išnyktų negrįžtamai mūsų praeityje.</p>

<p>Išgirdę apie skrebinimą nieko apie jį nežinantys žmonės tik klausiamai pakelia antakius, o papasakojus detaliau, kilsteli ir pečius. Tai atrodo tokia paprasta ir jokios apčiuopiamos naudos neduodanti veikla. Bet tereikia išsitraukti trumpam parodyti albumą ir tada skrebinimas įgauna prasmę. Mes turime socialinius tinklus, kuriuos kartas nuo karto, atiduodami savo privatumą, pildome kasdieniais įspūdžiais. Tai taip pat dalis mūsų istorijos. Visgi, vargu, ar kada nors, po penkių ar dešimties metų, atsikėlę nuo Kalėdų stalo, išsitrauksime mobiliuosius, kad kartu prisimintume praėjusius metus. Bet esu tikra, kad vartysime albumus – po penkių, po dešimties ir juo labiau po dvidešimties metų, kai vaikai paliks namus, ir tuščias lizdas atrodys toks didelis. Žinau, kad vartys ir mano vaikai. Kodėl? Todėl, kad jie tai daro jau dabar...<a href="#">Plačiau</a></p>

<img src="https://i.pinimg.com/736x/d0/5c/53/d05c531429eec58c0ce82c4b5562d2dd--travelling-photos-travelling-scrapbook.jpg" alt="foto iš interneto" width="300" height="300" />

<p>Skrebinimas - hobis ne tik suaugusiems. Jis ypatingai naudingas ir vaikams, nes:</p>

<ol>

<li>Lavina emocinį intelektą</li>

<li>Skatina kurti</li>

<li>Ugdo kantrybę</li>

<li>Lavina smulkiąją motoriką</li>

</ol>

<a href="#top">TOP</a>

<h2 id="rusys">Skrebinimo rūšys</h2>

<img src="https://iy.delfi.lt/norm/102523/4410329\_pBMJIq.jpeg" alt="pavyzdys" width="300" height="300">

<p>Galima išskirti keletą skrebinimo rūšių.</p>

<h3>Tradicinis skrebas</h3>

<p>Tradicinis skrebukas kuriamas naudojant nedidelį kiekį nuotraukų (dažniausiai vieną), didelį dėmesį skiriant puošybos elementams ir detalėms. Kaip ir kiti skrebukai, tradicinis skrebukas turi turėti pavadinimą (temą) arba žinutę nešančią frazę, kuri šiuo atveju tampa pagrindiniu akcentu. Tradiciniuose skrebuose ne visada sutiksite aprašymus (angl. journaling), tačiau nuotrauka ir frazė yra būtini. Plačiau skaitykite <a href="https://en.wikipedia.org/wiki/Scrapbooking">čia</a>.</p>

<h3>Project Life skrebas</h3>

<p>Project Life yra specifinė skrebinimo rūšis, kurioje nuotraukos dedamos ir įmautes su nedidelėmis kišenėlėmis. Ši skrebinimo rūšis ypatinga tuo, kad leidžia išsaugoti daug nuotraukų ir mažesnis dėmesys skiriamas puošybai.Plačiau skaitykite <a href="https://beckyhiggins.com/project-life/">čia</a>.</p>

<a href="#top">TOP</a>

<h2 id="irankiai">Skrebinimo įrankiai</h2>

<img src="https://g2.dcdn.lt/images/pix/file54920751\_ea9466a.jpg" alt="Skrebinimo irankiai" width="300" height="300">

<p>Ši rankdarbių rūšis - šiek tiek pavojinga, nes yra begalė įrankių, kurie naudojami kuriant skrebukus. Dažnai naujokui susisuka galva ir išsigąstama, kad skrebas - per prabangus hobis. Visgi, patys svarbiausi įrankiai yra tik keli ir juos galima rasti kiekvienuose namuose:</p>

<ul>

<li>Nuotraukos</li>

<li>Žirklės</li>

<li>Popierius</li>

<li>Klijai</li>

<li>Ir pagrindinis - VAIZDUOTĖ!</li>

</ul>

<a href="#top">TOP</a>

</body>

</html>

**1.2. Mokymosi rezultatas. Apipavidalinti internetinį puslapį naudojant pakopinius stilių**

**šablonus ir karkasus.**

**1.2.1. Tema. CSS pagrindai.**

*1 užduotis.* UŽRAŠYKITE CSS STILIAUS TAISYKLES (CSS SELEKTORIAI).

Rezultatai:

1. indexEx.html

<html>

<head>

<meta charset="UTF-8">

<title>Selectors Exercise</title>

<link rel="stylesheet" type="text/css" href="styleEx.css">

</head>

<body>

<h1>Selectors Exercise</h1>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<div>

<p class="hello">I am a paragraph with a class</p>

<p id="special">I am a paragraph with an ID</p>

<h2>I am an awesome h2 </h2>

<h3>I am an awesome h3 </h3>

<p id="special2">Roof party yr hella synth, Wes Anderson narwhal four dollar toast before they sold out retro lo-fi. Austin iPhone pop-up farm-to-table, PBR McSweeney's ennui messenger bag distillery before they sold out Portland wolf fanny pack YOLO. Locavore slow-carb trust fund farm-to-table. Pinterest gastropub lo-fi, McSweeney's trust fund VHS shabby chic ugh Austin twee. Messenger bag banjo lumbersexual, whatever 3 wolf moon <span>XOXO (red)</span> normcore. Pug fanny pack 3 wolf moon, typewriter organic chia mustache scenester seitan shabby chic Blue Bottle salvia ugh iPhone. Fanny pack Williamsburg direct trade, cold-pressed disrupt flannel listicle health goth asymmetrical freegan mixtape street art pour-over whatever.</p>

</div>

<div>

<h2>Things I need to do </h2>

<ul>

<li>Walk Dog <input type="checkbox" checked> </li>

<li>Feed Dog <input type="checkbox" checked> </li>

<li>Wash Dog <input type="checkbox"></li>

</ul>

</div>

<div>

<h2 class="hello">I am another awesome h2 </h2>

<p>Cardigan Tumblr mlkshk, fap tilde 3 wolf moon Portland. Heirloom health goth taxidermy blog lo-fi selfies, post-ironic master cleanse fingerstache normcore. Kickstarter plaid twee, bespoke single-origin coffee sustainable lo-fi vinyl Pinterest pork belly <em>cronut skateboard</em> 3 wolf moon. Normcore single-origin coffee salvia, bespoke Austin swag Godard before they sold out kogi disrupt locavore. Lumbersexual Shoreditch Vice, artisan American Apparel master cleanse yr salvia vegan. Bespoke letterpress <span>XOXO</span> heirloom kale chips deep v four loko. Lomo sustainable put a bird on it trust fund post-ironic</p>

<p>I'm the second paragraph inside this div!</p>

</div>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<div>

<h2>A less awesome h2 </h2>

<p>Roof party yr hella synth, Wes Anderson narwhal four dollar toast before they sold out retro lo-fi. Austin iPhone pop-up farm-to-table, PBR&B McSweeney's ennui messenger bag distillery before they sold out Portland wolf fanny pack YOLO. Locavore slow-carb trust fund farm-to-table. Pinterest gastropub lo-fi, McSweeney's trust fund VHS shabby chic ugh Austin twee. Messenger bag banjo lumbersexual, whatever 3 wolf moon XOXO normcore. Pug fanny pack 3 wolf moon, typewriter organic chia mustache scenester seitan shabby chic Blue Bottle salvia ugh iPhone. Fanny pack Williamsburg direct trade, cold-pressed disrupt flannel listicle health goth asymmetrical freegan mixtape street art pour-over whatever</p>

<p>One last paragraph here!</p>

<p> List inside a DIV</p>

<ul>

<li>Dog

<ul>

<li>Dog Dog</li>

<li>Dog Dog</li>

</ul>

</li>

<li> Cat </li>

<li> Mouse

<ul>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

</ul>

</li>

</ul>

</div>

<p>PARAGRAPH NOT INSIDE A DIV</p>

<p> List NOT inside a DIV</p>

<ul>

<li>Dog

<ul>

<li>Dog Dog</li>

<li>Dog Dog</li>

</ul>

</li>

<li> Cat </li>

<li> Mouse

<ul>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

<li>Mouse Mouse</li>

</ul>

</li>

</ul>

</body>

</html>

2. styleEx.css

/\*

Style the HTML elements according to the following instructions.

WRITE ONLY CSS!

\*/

/\* Give the <body> element a background of lightgray\*/

\*{

    background-color: lightgray;

}

/\* Make the <h1> element purple \*/

h1{

    color: purple;

}

/\* Make all <h2> and <h3> elements orange \*/

h2, h3{

    color: orange;

}

/\* Make all the <p>'s that are nested inside of divs 25px font(font-size: 25px) \*/

div > p{

    font-size: 25px;

}

/\* Give everything with the class 'hello' a white background\*/

.hello{

    background-color: white;

}

/\* Give the element with id 'special' a 2px solid blue border \*/

#special{

    border: 2px;

    border-style: solid;

    border-color: blue;

}

/\* Make the <span> element in <p> element with id 'special2' a 2px solid red border \*/

#special2 span{

    border: 2px;

    border-style: solid;

    border-color: red;

}

/\* Change the style of multilevel list in <div> element: first level of list darkblue, second - pupple \*/

div ul{

    color: purple;

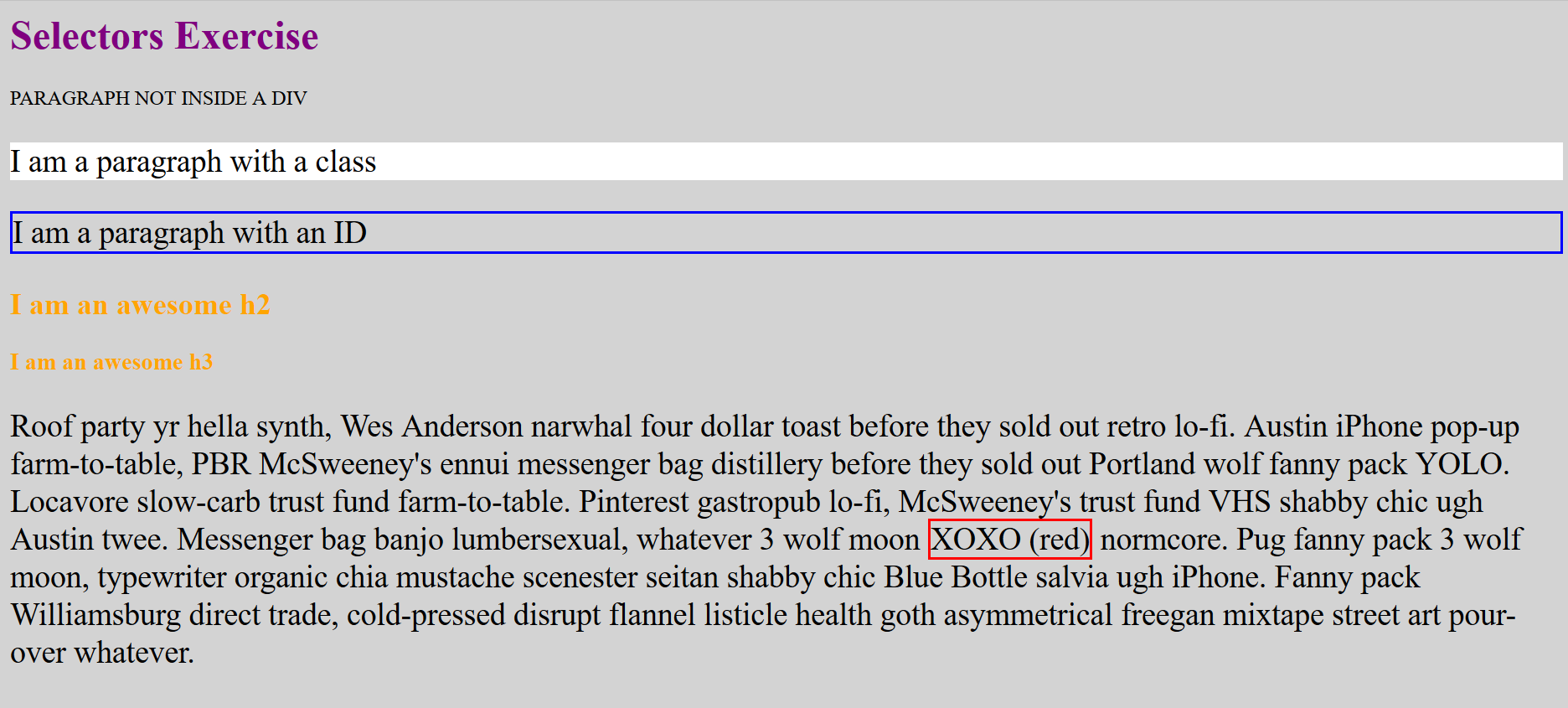
}

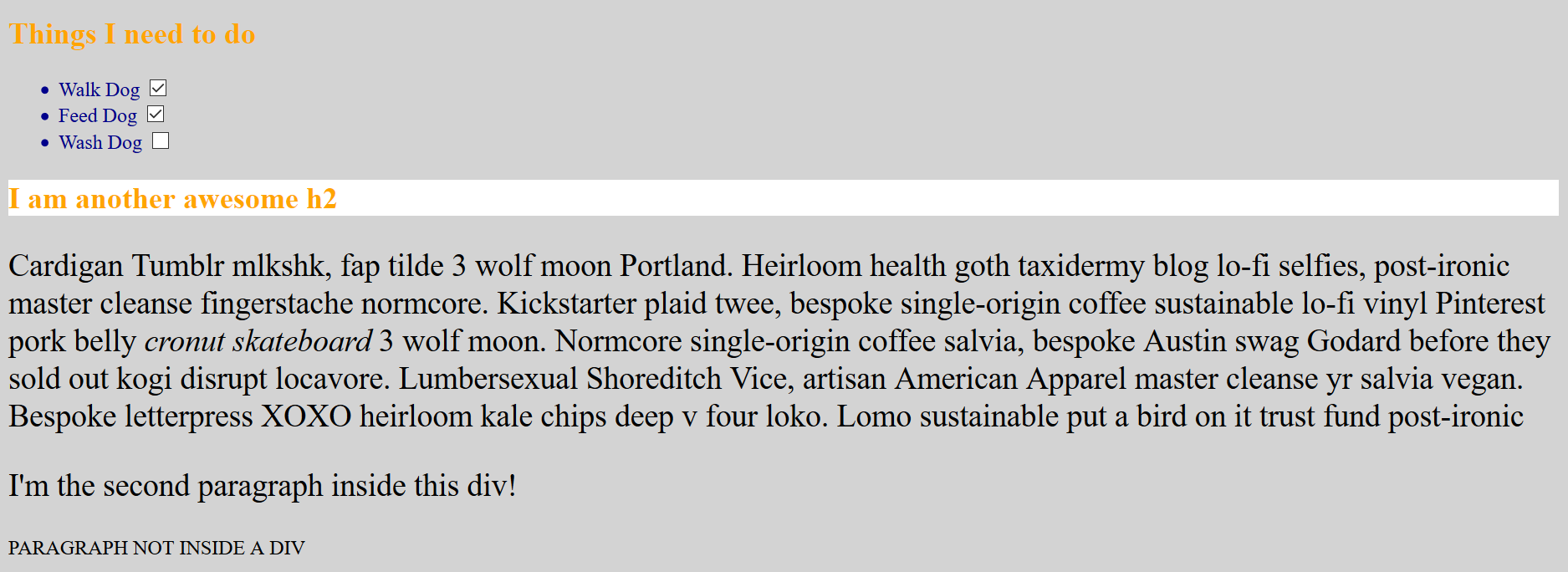
div > ul{

    color: darkblue;

}

Bendras rezultatas naršyklėje turi atrodyti taip:







*2 užduotis*. UŽRAŠYKITE CSS STILIAUS TAISYKLES (RECEPTAI).

Rezultatai:

1. index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header class="container">

<h1>Recipes</h1>

<h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit. </h3>

<nav>

<ul>

<li><a href="#">sandwiches</a></li>

<li><a href="#">cupcakes</a></li>

<li><a href="#">hot drinks</a></li>

</ul>

</nav>

</header>

<main>

<article>

<section class="container description">

<h3>Description</h3>

<p>This peanut butter and jelly sandwich is my favorite sandwich. It has the perfect balance of ingredients and looks great when made right.</p>

</section>

<section class="container">

<img src="https://image.flaticon.com/icons/png/512/63/63035.png" alt="sandwich" width="200" height="200">

<h3>Ingredients</h3>

<ul>

<li>2 slices of white bread</li>

<li>1 jar of grape jelly</li>

<li>1 jar of creamy peanut butter</li>

<li>A butter knife</li>

<li>A sharp knife</li>

<li>A cutting board</li>

</ul>

</section>

<section class="container description">

<h3>Directions</h3>

<ol>

<li>Lay both slices of bread next to each other on a cutting board.</li>

<li>With the butter knife, spread a 1/8 inch layer of peanut butter on the left side</li>

<li>On the right side, spread a 1/8 inch layer of jelly.</li>

<li>Carefully place the two halves together so that the jelly is on top.</li>

<li>With the sharp knife, carefully cut the sandwich in half.</li>

<li>Enjoy the PB&amp;J!</li>

</ol>

</section>

</article>

</main>

<footer class="container">

<nav>

<ul>

<li><a href="#">sandwiches</a></li>

<li><a href="#">cupcakes</a></li>

<li><a href="#">hot drinks</a></li>

</ul>

</nav>

<h5>Copyright &copy; 2008 — 2018 WEBMASTER. </h5>

</footer>

</body>

</html>

1. style.css

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Numatytųjų naršyklės stiliaus taisyklių perrašymas

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Visos išorinės paraštės (margin) lygios 0 \*/

\* {

    margin:0;

}

/\* Šrifto tipo ir dydžio taisyklės visam tinklapiui \*/

body{

    font-family: 'Questrial', sans-serif;

    font-size: 12pt;

}

/\* Kelios klasės \*/

.description{

    background-color: #F3F3F3;

}

.container{

    padding: 30px 10px;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Tinklapio antraštė

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* header:

    fono spalva: #16A085,

    teksto - balta.

\*/

header{

    background-color: #16A085;

    color: white;

}

/\* <h1>, esantis <header> dalyje:

    tekstas: visos didžiosios raidės,

    vidinės paraštės (padding): 20px 0,

\*/

header h1{

    text-transform: uppercase;

    padding: 20px 0;

}

/\* <h3>, esantis <header> dalyje:

    tarpai tarp raidžių: 0,3 em

\*/

header h3{

    letter-spacing: 0.3em;

}

/\* <nav>, esantis <header> dalyje:

    viršutinis rėmelis: 2px dashed white;

    viršutinė vidinė paraštė: 20px;

\*/

header nav {

    border-top: 2px dashed white;

    padding-top: 20px;

}

/\* nuoroda, esanti <header> dalyje:

    spalva: balta;

    teksto pabraukimo nėra;

\*/

header nav a{

    text-decoration: none;

    color: white;

}

/\* sąrašas, esantis <header> dalyje:

    sąrašo ženklinimo nėra;

    visos vidinės paraštės lygios 0;

\*/

header nav ul{

    list-style: none;

    padding: 0;

}

/\*

sąrašas išdėstytas eilutėje:

\*/

header nav li{

    display: inline-block;

    width: 150px;

}

/\* Pseudo klases nuorodoms, <header> dalyje \*/

/\* :link

    teksto spalva: balta;

\*/

header nav a:link {

    color: white;

}

/\* :visited

    teksto spalva: balta;

\*/

header nav a:visited {

    color: white;

}

/\* :hover

    teksto spalva: balta;

    tesktas pabrauktas;

\*/

header nav a:hover {

    color: white;

    text-decoration: underline;

}

/\* :active

    teksto spalva: pilka;

\*/

header nav a:active {

    color: grey;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

         Tinklapio pagrindinė dalis | Main

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* <h3>, esantis <article>:

    tekstas: visos didžiosios raidės;

    teksto spalva: #16A085;

    tarpai tarp raidžių: 0,3em;

\*/

article h3{

    text-transform: uppercase;

    color: #16A085;

    letter-spacing: 0.3em;

}

/\* <ul>, esančio <article>:

    ženklinto sąrašo stilius: kvadratas;

\*/

article ul {

    list-style-type: square;

}

/\* <ol>, esančio <article>:

    numeracijos stilius: mažosios romėniškos raidės;

    eilutės aukštis: 2;

\*/

article ol {

    list-style-type:lower-roman;

    line-height: 2;

}

/\* pirmoji <section>, esanti <main> dalyje:

    teksto lygiavimas: centruotas (center);

\*/

main section:nth-child(1){

    text-align: center;

}

/\* paveiklas:

    rėmelis: 1px solid #16A085;

\*/

img{

    border: 1px solid #16A085;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Tinklapio baigiamoji dalis | footer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* <footer> dalys:

    teksto lygiavimas: centruotas;

    fono spalva: #16A085;

\*/

footer{

    text-align: center;

    background-color: #16A085;

}

/\* nuorodos, esančio <footer>:

    teksto pabraukimo nėra;

    teksto spalva: balta;

\*/

footer ul a{

    text-decoration: none;

    color: white;

}

/\* <h5>, esančio <footer> dalyje:

    viršutinės vidinė paraštė: 20px;

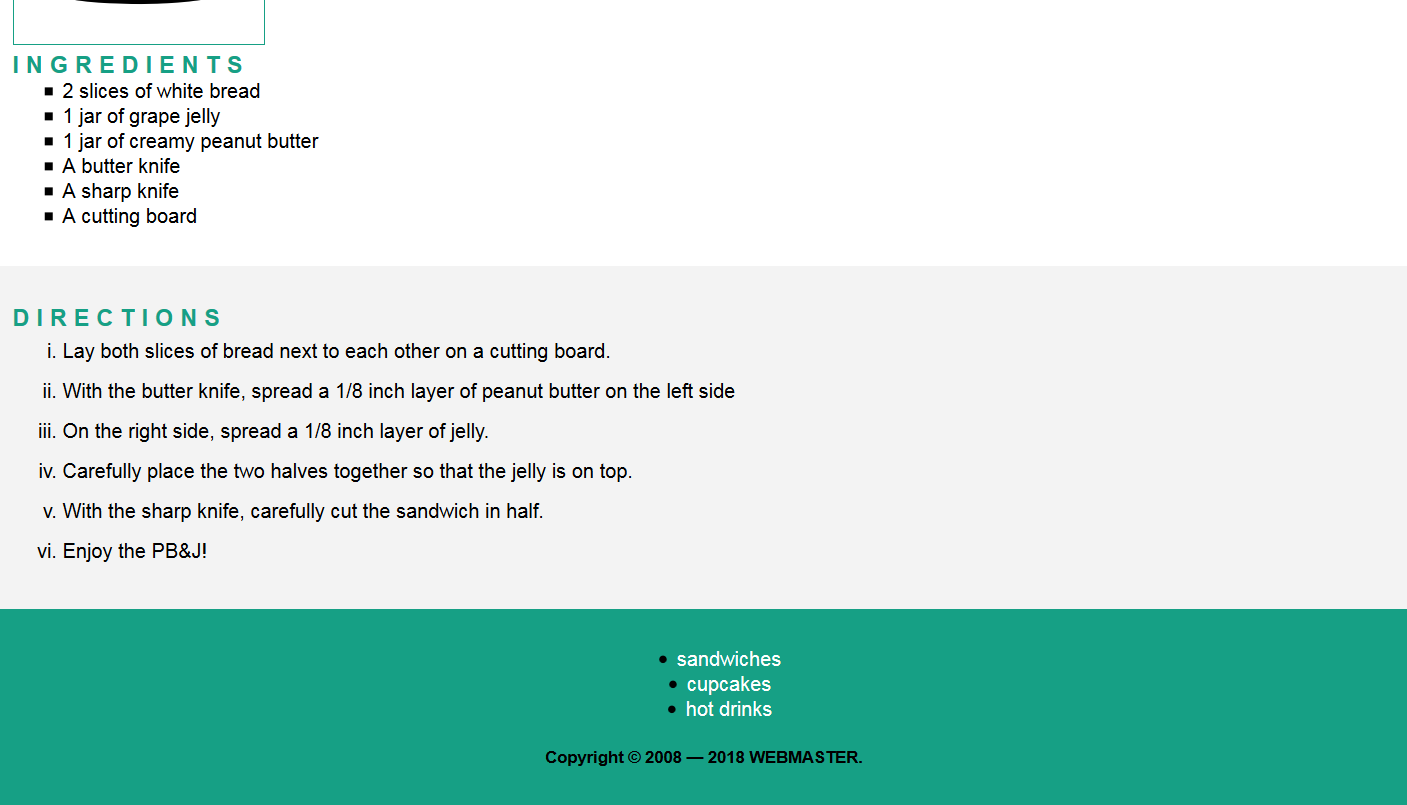
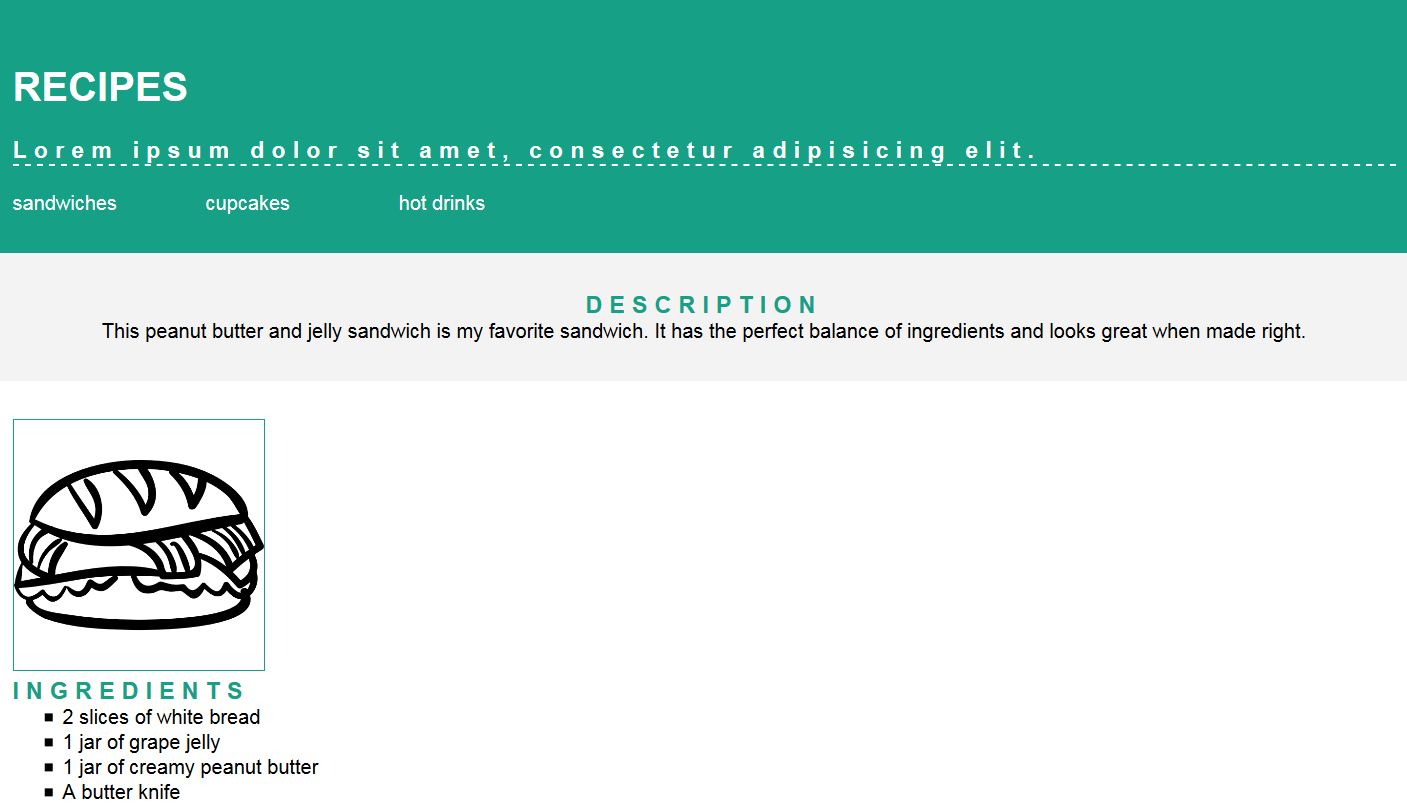
\*/

footer h5{

    padding-top: 20px;

}

Bendras rezultatas naršyklėje turi atrodyti taip:



**1.2.2. Tema. CSS tinklalapio maketo kūrimo technikos.**

*3 užduotis****.*** UŽRAŠYTI CSS KLASES (PARAMETRAI „FLOAT“ IR „CLEAR“).

Bendras rezultatas naršyklėje turi atrodyti taip:







**2.** **Kompetencija. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.**



**2.1.** **Mokymosi rezultatas.** **Suprasti skaičiavimo sistemas.**

**2.1.1. Tema. Skaičiavimo sistemos.**

*1 užduotis.* APRAŠYKITE KAIP NUMERAVO SENOVĖS BABILONIEČIAI, EGIPTIEČIAI, GRAIKAI, SLAVAI.

Informaciją galite rasti čia:

1. Wikipedia (2019). *Numeral system*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Numeral_system>.

2. Vilniaus universitetas (2019). Apie skaičiavimo sistemas. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.mif.vu.lt/ljmm/rasmenys/skaiciavimo_sistemos.htm>.

*2 užduotis.* SURAŠYKITE KOKIOSE ŽMOGAUS VEIKLOS SRITYSE SKAIČIAVIMO SISTEMOS TURI DIDŽIAUSIĄ REIKŠMĘ.

Atsakydami į šį klausimą galite naudotis ekonominės veiklos rūšių klasifikatoriumi:

1. Lietuvos statistikos departamentas (2019). *Ekonominės veiklos rūšių klasifikatorius*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://osp.stat.gov.lt/static/evrk2.htm>

*3 užduotis.* SURAŠYKITE VISAS JUMS ŽINOMAS SKAIČIAVIMO SISTEMAS (POZICINĖS).

Dešimtainė, dvejetainė, aštuntainė, šešioliktainė, N-ainė skaičiavimo sistemos. Papildomai galite paskaityti:

Wikipedia (2019). *List of numeral systems*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/List_of_numeral_systems>

*4 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 125, 256, 2019 DVEJETAINIAIS SKAIČIAIS.

12510 = 11111012

25610 = 1000000002

201910 = 111111000112

Apie skaičių keitimą iš vienos sistemos į kitą galite paskaityti čia:

Tutorials Point (2019). *Number System Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/computer_logical_organization/number_system_conversion.htm>

*5 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS DVEJETAINĖS SISTEMOS SKAIČIUS: 101100110001, 1010101, 001110011.

1011001100012 = 286510

10101012 = 8510

0011100112 = 11510

*6 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 14, 56, 2019 AŠTUNTAINIAIS SKAIČIAIS.

1410 = 168

5610 = 708

201910 = 37438

*7 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS AŠTUNTAINĖS SISTEMOS SKAIČIUS: 77, 66, 32.

778 = 6310

668 = 5410

328 = 2610

*8 užduotis.* UŽRAŠYKITE DEŠIMTAINIUS SKAIČIUS 62, 216, 2019 ŠEŠIOLIKTAINIAIS SKAIČIAIS.

6210 = 3e16

21610 = d816

201910 = 7e316

*9 užduotis.* UŽRAŠYKITE DEŠIMTAINIAIS SKAIČIAIS ŠIUOS ŠEŠIOLIKTAINĖS SISTEMOS SKAIČIUS: 1D5F, A9BC, FF.

1d5f16 = 751910

a9bc16 = 4345210

FF16 = 25510

**2.1.2. Tema. Informacijos matavimo vienetai.**

*10 užduotis.* UŽRAŠYKITE VISUS JUMS ŽINOMUS INFORMACIJOS KIEKIO MATAVIMO VIENETUS.

Bitas, baitas, kilobaitas, megabaitas gigabaitas, terabaitas, petabaitas, eksabaitas.

Papildomai galite paskaityti čia:

Wikipedia (2019). *Units of information*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Units_of_information>.

*11 užduotis.* TURITE 16GB USB ATMINTUKĄ. PASKAIČIUOKITE KIEK TOKS ATMINTUKAS TURI MEGABAITŲ (MB), KILOBAITŲ (KB) IR BAITŲ (B).

Paskaičiuokite, kiek galite patalpinti į tokią laikmeną bitų informacijos

16GB = 16 384 MB (megabaitų)

16GB = 16 777 216 KB (kilobaitų)

16GB = 17 179 869 184 B (baitų)

16GB = 137 438 953 472 b (bitų)

Papildomai galite paskaityti čia:

GbmbOrg (2019). *Data Units Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.gbmb.org.>

*12 užduotis.* JŪSŲ INTERNETO PASLAUGŲ TIEKĖJAS PER MĖNESĮ UŽTIKRINA JUMS 8589934592 BITŲ DUOMENŲ KIEKĮ. KIEK NUOTRAUKŲ GALITE PERSIŲSTI KIEKVIENĄ MĖNESĮ JEI VIENOS NUOTRAUKOS DYDIS BUS 5MB.

Galima persiųsti 204 nuotraukas. 8589934592 bitų tai 1024MB. 1024MB / 5MB = 204,8 nuotraukos.

*13 užduotis.* JŪSŲ INTERNETO GREITAVEIKA 100MBPS. KIEK LAIKO UŽTRUKS 100MB FAILO SIUNTIMAS. PRIIMAME, KAD INTERNETO GREITAVEIKA NESIKEIČIA IR YRA PASTOVI.

100Mbps tai yra 100 megabitų per sekundę arba 12,5 megabaitų (MB) per sekundę. 100MB / 12,5MB/s = 8 sekundės.

Papildomai galite paskaityti čia:

Wikipedia (2019). *Data rate units*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Data-rate_units.>

*14 užduotis.* KOKIA TURI BŪTI JŪSŲ INTERNETO GREITAVEIKA, KAD PER 5 MINUTES GALIMA BŪTŲ PERDUOTI 600MB INFORMACIJOS.

1. minutės tai 300 sekundžių. 600MB / 300 sekundžių = 2MB/s arba 16Mbps.
   1. **Mokymosi rezult****atas. Taikyti algoritmų ir logikos mokslo pagrindus programuojant.**

**2.2.1. Tema. Logikos mokslo pagrindai.**

*1* *užduotis.*  FORMALIZUOKITE PATARLĘ: „DEGTINEI Į TROBĄ ĮEINANT PROTAS IŠEINA LAUKAN“ TEIGINIŲ LOGIKOS PRIEMONĖMIS.

Sakinį sudaro du elementarūs teiginiai:

p – degtinė įeina į trobą,

q – protas išeina laukan.

Sakinio prasmė: “Jeigu degtinė įeina į trobą, tai protas išeina laukan”. Elementarius teiginius sieja implikacijos jungtis. Simboliškai tai užrašoma taip: **p → q**.

*2 užduotis*. FORMALIZUOKITE PATARLĘ: „DEGTINĖ IR ALUS GIMDO VARGUS“ TEIGINIŲ LOGIKOS PRIEMONĖMIS.

Sakinį sudaro du elementarūs teiginiai:

p – degtinė gimdo vargus;

q – alus gimdo vargus.

Sakinio prasmė: “Alus arba degtinė, arba abu drauge gimdo vargus”. Elementarius teiginius sieja silpnosios disjunkcijos jungtis. Simboliškai tai užrašoma taip: **p∨q**.

*3 užduotis.* UŽRAŠYKITE DE MORGANO TAISYKLĖS KONJUNKCIJAI SIMBOLINĘ IŠRAIŠKĄ.

De Morgano taisyklės konjunkcijai simbolinė išraiška yra:

**¬ ( p∧q)∼(¬ p∨¬ q)**

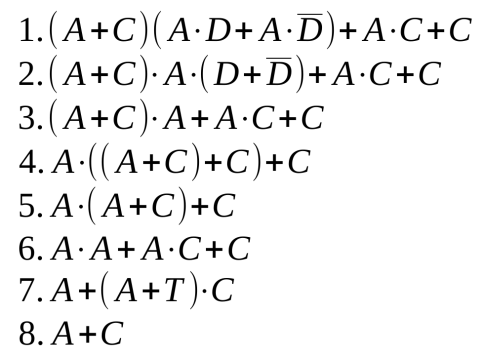
Skaitoma: “Netiesa, kad p ir q ekvivalentiška tam, kad ne-p arba ne-q”. Interpretacija. Pasižymime:

p – asmuo myli žmoną,

q – asmuo myli uošvę.

Tada teiginys „Netiesa, kad asmuo myli ir žmoną, ir uošvę “ekvivalentiškas teiginiui „Asmuo nemyli arba žmonos, arba uošvės, arba abiejų“.

*4 užduotis*. UŽRAŠYKITE DE MORGANO TAISYKLĖS KONJUNKCIJAI SIMBOLINĘ IŠRAIŠKĄ.



Papildomai galite paskaityti čia:

Wikipedia (2019). *Boolean Algebra*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Boolean_algebra.>

**2.2.2. Tema. Algoritmai ir algoritmavimas.**

*5 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS APSKAIČIUOJA SKAIČIŲ MASYVO SUMĄ IR VIDURKĮ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javaalgorithms;

public class IterativeAlgorithm {

public static void main(String[] args) {

int[] intArray = {54, 17, 1, 21, 85, 3, 18, 42};

int result = 0;

for(int i = 0; i < intArray.length; i++){

result += intArray[i];

}

System.out.println("Sum of the array elements: " + result);

System.out.println("Average of the array elements: " + (double) result / intArray.length);

}

}

*6 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURANDA MAŽIAUSIĄ MASYVO SKAIČIŲ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javaalgorithms;

public class RecursiveAlgorithm {

public static void main(String[] args) {

int[] intArray = {27, 32, 19, -7, 41, 38, 99, -36, 54, 17, 1, 21, 85, 3, -18, 42};

int arrayLength = intArray.length;

System.out.println(findMinNumber(intArray, arrayLength));

}

private static int findMinNumber(int[] intArray, int arrayLength) {

if (arrayLength == 1) {

return intArray[0];

} else {

return Math.min(intArray[arrayLength - 1],

}

findMinNumber(intArray, arrayLength - 1));

}

}

*7 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SKAIČIUOJA 120 FIBONAČIO SKAIČIŲ.

**Pavyzdys**. Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.math.BigInteger;

public class JavaDynamicProgrammingFibonacci {

public static void main(String[] args) {

int n = 120;

System.out.println(fibonacciNumberCounting(n));

}

private static BigInteger fibonacciNumberCounting(int n) {

BigInteger a = BigInteger.valueOf(0);

BigInteger b = BigInteger.valueOf(1);

BigInteger c = BigInteger.valueOf(1);

for (int j = 2; j <= n; j++) {

c = a.add(b);

a = b;

b = c;

}

return c;

}

}

*8 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS NUSTATO AR RINKINYJE (SET) YRA ELEMENTŲ RINKINYS (SUBSET) LYGUS NURODYTAI SUMAI.

**Pavyzdys**. Algoritmas Java programavimo kalba:

package javaalgorithms;

public class JavaDynamicProgrammingSubsetSum {

public static void main(String[] args) {

int[] set = {3, 34, 4, 12, 5, 2};

int sum = 170;

int setLength = set.length;

if (isSubsetSum(set, setLength, sum)) {

System.out.println("Found a subset with given sum");

} else {

System.out.println("No subset with given sum");

}

}

private static boolean isSubsetSum(int[] set, int n, int sum) {

boolean[][] subset = new boolean[sum + 1][n + 1];

for (int i = 0; i <= n; i++){

subset[0][i] = true;

}

for (int i = 1; i <= sum; i++) {

subset[i][0] = false;

}

for (int i = 1; i <= sum; i++) {

for (int j = 1; j <= n; j++) {

subset[i][j] = subset[i][j - 1];

if (i >= set[j - 1]) {

subset[i][j] = subset[i][j] || subset[i - set[j - 1]][j - 1];

}

}

}

return subset[sum][n];

}

}

*9 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURŪŠIUOJA SVEIKŲJŲ SKAIČIŲ MASYVĄ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.util.Arrays;

public class JavaQuickSort {

private int[] array;

private int length;

public static void main(String[] args) {

JavaQuickSort sorter = new JavaQuickSort();

int[] arrayOne = {24, -2, 45, 20, 56, -75, 2, 56, 99, 53, -12};

System.out.println("Unsorted array: \n" + Arrays.toString(arrayOne));

System.out.println("Sorted array: ");

sorter.sort(arrayOne);

for (int i : arrayOne) {

System.out.print(i);

System.out.print(", ");

}

}

private void sort(int[] inputArr) {

if (inputArr == null || inputArr.length == 0) {

return;

}

this.array = inputArr;

length = inputArr.length;

quickSort(0, length - 1);

}

private void quickSort(int lowerIndex, int higherIndex) {

int i = lowerIndex;

int j = higherIndex;

int pivot = array[lowerIndex + (higherIndex - lowerIndex) / 2];

while (i <= j) {

while (array[i] < pivot) {

i++;

}

while (array[j] > pivot) {

j--;

}

if (i <= j) {

exchangeNumbers(i, j);

i++;

j--;

}

}

if (lowerIndex < j) {

quickSort(lowerIndex, j);

}

if (i < higherIndex) {

quickSort(i, higherIndex);

}

}

private void exchangeNumbers(int i, int j) {

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

*10 užduotis.* PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SURŪŠIUOJA SVEIKŲJŲ SKAIČIŲ MASYVĄ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javaalgorithms;

public class JavaBubbleSort {

public static void main(String[] args) {

int[] inputNumber = {4, 2, 9, 6, 23, 12, 34, 0, 1};

bubble\_srt(inputNumber);

}

private static void bubble\_srt(int[] array) {

int n = array.length;

int k;

for (int m = n; m >= 0; m--) {

for (int i = 0; i < n - 1; i++) {

k = i + 1;

if (array[i] > array[k]) {

swapNumbers(i, k, array);

}

}

printNumbers(array);

}

}

private static void swapNumbers(int i, int j, int[] array) {

int temp;

temp = array[i];

array[i] = array[j];

array[j] = temp;

}

private static void printNumbers(int[] input) {

for (int inp : input) {

System.out.print(inp + " ");

}

System.out.println("\n");

}

}

*11 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS SIMBOLIŲ EILUTĘ ATSPAUSDINA ATVIRKŠČIAI.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.util.Stack;

public class JavaDataStructuresStack {

public static void main(String[] args) {

String str = "Java for all";

reverseWords(str);

}

private static void reverseWords(String str) {

Stack<Character> characterStack = new Stack<>();

for (int i = 0; i < str.length(); ++i) {

if (str.charAt(i) != ' ') {

characterStack.push(str.charAt(i));

} else {

while (!characterStack.empty()) {

System.out.print(characterStack.pop());

System.out.print(" ");

}

}

while (!characterStack.empty()) {

System.out.print(characterStack.pop());

}

}

}

*12 užduotis*. PARAŠYKITE PROGRAMĄ BET KOKIA JUMS ŽINOMA PROGRAMAVIMO KALBA IR/ARBA PAVAIZDUOKITE ALGORITMĄ, KURIS IEŠKO MAŽIAUSIĄ SKAIČIŲ AIBĖJE.

**Pavyzdys**. Algoritmas Java programavimo kalba:

package javaalgorithms;

class Node {

int data;

Node left, right;

Node(int d) {

data = d;

left = right = null;

}

}

package javaalgorithms;

public class JavaDataStructuresBinaryTree {

public static void main(String[] args) {

JavaDataStructuresBinaryTree binaryTree = new JavaDataStructuresBinaryTree();

Node root = null;

root = binaryTree.insert(root, 4);

binaryTree.insert(root, 23);

binaryTree.insert(root, -12);

binaryTree.insert(root, 3);

binaryTree.insert(root, 65);

binaryTree.insert(root, 54);

System.out.println("Minimum value of binary binaryTree is "

+ binaryTree.minvalue(root));

}

private Node insert(Node node, int data) {

if (node == null) {

return (new Node(data));

} else {

if (data <= node.data) {

node.left = insert(node.left, data);

} else {

node.right = insert(node.right, data);

}

return node;

}

}

private int minvalue(Node node) {

Node current = node;

while (current.left != null) {

current = current.left;

}

return (current.data);

}

}

**2.3. Mokymosi rezultatas. Taikyti programinio kodo dizaino modelius programuojant.**

**2.3.1. Tema. Dizaino šablonai.**

*1 užduotis*. UŽPILDYKITE PATEIKTĄ LENTELĘ.

Dizaino šablonai:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Dizaino šablono rūšys** | | |
|  | **Creational** | **Structural** | **Behavioral** |
| **Šablonas** | Abstract Factory | Adapter | Chain of Responsibility |
| Builder | Bridge | Command |
| Factory Method | Composite | Interpreter |
| Prototype | Decorator | Iterator |
| Singleton | Facade | Mediator |
|  | Flyweight | Memento |
|  | Proxy | Observer |
|  |  | State |
|  |  | Strategy |
|  |  | Template Method |
|  |  | Visitor |

Papildomai galite paskaityti čia:

Wikipedia (2019). *Projektavimo pavyzdys.* [žiūrėta 2019-05-27].Prieiga per internetą[https://lt.wikipedia.org/wiki/Projektavimo\_pavyzdys*.*](https://lt.wikipedia.org/wiki/Projektavimo_pavyzdys.)

*2 užduotis.* APRAŠYKITE PASIRINKTINAI TRIS DIZAINO ŠABLONUS.

**Singleton šablonas (Vienintelis objektas):** užtikrina, kad būtų leidžiama naudoti tik vieną klasės egzempliorių. Naudojamas kai reikia vieno klasės egzempliorius ir būtina kontroliuoti prieigą prie vieno objekto. Pavyzdys: objektas reprezentuoja resursą: ekraną, klaviatūrą, garso posistemę ir pan.

**Decorator šablonas (Dekoratorius):** naudojamas dinamiškai keisti ir pridėti funkcionalumo objektui. Objekto atsakomybė ir elgesys turėtų būti keičiami dinamiškai. Konkretūs įgyvendinimai turėtų būti atsieti nuo atsakomybės ir elgesio. Pavyzdys: daugelis įmonių savo pašto sistemas sukūrė naudodamosi dekoratoriais. Kai pranešimai siunčiami iš įmonės į išorinį adresą, pašto serveris papuošia originalų pranešimą su autorių teisių ir konfidencialumo informacija. Kol pranešimas lieka vidinis, informacija nėra pridedama.

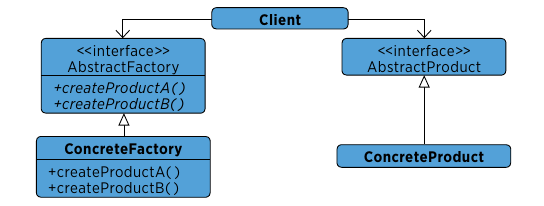
**Strategy šablonas (Strategija):** naudojamas, kuomet bendrą algoritmą reikia varijuoti tam tikru aspektu. Pavyzdžiui apskaitos programoje strategija galėtų būti - mokesčių skaičiavimas skirtingoms šalims. Tai leistų nesunkiai kurti programos versijas skirtingoms valstybėms.

Papildomai galite paskaityti čia:

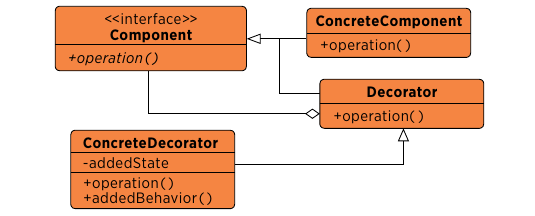
Wikipedia (2019). *Software design pattern.* [žiūrėta 2019-05-27]. Prieiga per internetą<https://en.wikipedia.org/wiki/Software_design_pattern.>

**2.3.2. Tema. Dizaino šablonų taikymas.**

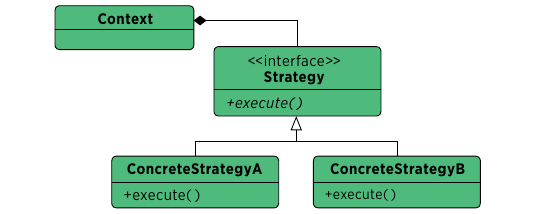
*3 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ ABSTRAKČIOS FABRIKOS ŠABLONUI (ABSTRACT FACTORY PATTERN).



*4 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ DEKORATORIAUS ŠABLONUI (DECORATOR PATTERN).



*5 užduotis*. NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ STRATEGIJOS ŠABLONUI (STRATEGY PATTERN).



Braižant UML klasių diagramas galite naudoti:

JGraph Ltd (2019). *draw.io*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.draw.io.>

*6 užduotis.* PARINKITE DIZAINO ŠABLONĄ PATEIKTAM GYVENIMO PAVYZDŽIUI: GAMYKLA GAMINA TRIJŲ MODELIŲ AUTOMOBILIUS.

Geriausiai šioje situacijoje naudoti fabriko (Factory) šabloną.

*7 užduotis.* PARINKITE DIZAINO ŠABLONĄ PATEIKTAM GYVENIMO PAVYZDŽIUI: TURIME PAŠTO SERVERĮ.

Geriausiai šioje situacijoje naudoti dekoratoriaus šabloną.

**2.4.** **Mokymosi rezultatas. Naudoti informacinių sistemų kūrimo principus ir metodus programinės įrangos projektavime.**

**2.4.1. Tema. Daugiasluoksnė programų architektūra ir MVC struktūra.**

*1 užduotis*. APRAŠYKITE KLIENTO-SERVERIO (CLIENT SERVER) PROGRAMINĖS ĮRANGOS MODELĮ.

Aprašant programinės įrangos modelius naudokitės šia medžiaga:

Maskeliūnas, S. (2007). *Programų sistemų architektūra ir projektavimas.* Mokymo medžiaga parengta vykdant projektą „Programų sistemų magistrantūros įsteigimas“. [žiūrėta 2019-05-27]. Prieiga per internetą <https://klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf>

*2 užduotis*. APRAŠYKITE MODELIO-VAIZDO-KONTROLERIO (MODEL VIEW CONTROLLER, MVC) PROGRAMINĖS ĮRANGOS MODELĮ.

Aprašant programinės įrangos modelius naudokitės šia medžiaga:

1. Maskeliūnas, S. (2007). *Programų sistemų architektūra ir projektavimas. Mokymo medžiaga parengta vykdant projektą „Programų sistemų magistrantūros įsteigimas*“. [žiūrėta 2019-05-27]. Prieiga per internetą <https://klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf>

*3 užduotis*. NAUDOJANT MVC ARCHITEKTŪROS MODELĮ, SUKURKITE PAPRASTĄ DARBUOTOJŲ VALDYMO PROGRAMĄ.

**Pavyzdys.** Modelis realizuotas Java programavimo kalba:

package javamvc;

class EmployeeModel {

private String id;

private String name;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

package javamvc;

public class EmployeeController {

private EmployeeModel model;

private EmployeeView view;

public EmployeeController(EmployeeModel model, EmployeeView view) {

this.model = model;

this.view = view;

}

public void setEmployeeName(String name) {

model.setName(name);

}

public String getEmployeeName() {

return model.getName();

}

public void setEmployeeId(String rollNo) {

model.setId(rollNo);

}

public String getEmployeeId() {

return model.getId();

}

public void updateView() {

view.printEmployeeDetails(model.getName(), model.getId());

}

}

package javamvc;

class EmployeeView {

public void printEmployeeDetails(String name, String no) {

System.out.println("EmployeeModel: ");

System.out.println("Name: " + name);

System.out.println("ID: " + no);

}

}

package javamvc;

public class Main {

public static void main(String[] args) {

EmployeeModel employee = new EmployeeModel();

EmployeeView view = new EmployeeView();

EmployeeController controller = new EmployeeController(employee, view);

employee.setId("1");

employee.setName("Tom");

controller.updateView();

controller.setEmployeeName("Dave");

controller.updateView();

}

}

Papildomai galite paskaityti:

Wikipedia (2019). *Model View Controller*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Model-view-controller.>

*4 užduotis*. NAUDOJANT PASIRINKTĄ KARKASĄ SUKURKITE PAPRASTĄ WEB VARTOTOJŲ REGISTRACIJOS PROGRAMĄ.

Užduoties sprendimui galite naudoti React ar kitą MVC karkasą.

**2.4.2. Tema. Pagrindiniai programavimo principai.**

*5 užduotis*. APRAŠYKITE SOLID IR DRY PROGRAMAVIMO PRINCIPUS.

Užduoties sprendimui naudokite šiuos resursus:

1. Wikipedia (2019). *SOLID*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/SOLID.>

2. Wikipedia (2019). *Don’t repeat yourself*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Don’t_repeat_yourself.>

*6 užduotis*. APRAŠYKITE SEPARATION OF CONCERNS IR CODE REUSE PROGRAMAVIMO PRINCIPUS.

Užduoties sprendimui naudokite šiuos resursus:

1. Wikipedia (2019). *Separation of Concerns*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Separation_of_concerns.>

2. Wikipedia (2019). *Code reuse*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Code_reuse.>

*7 užduotis*. NAUDOJANT FUNKCINIO PROGRAMAVIMO PRINCIPUS REALIZUOKITE GREITO RŪŠIAVIMO ALGORITMĄ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javafunctional;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.function.Function;

import java.util.function.Predicate;

import java.util.stream.Collectors;

import java.util.stream.Stream;

public class JavaQuickSortFunctional {

private static Function<Integer, Predicate<Integer>> smallerThan = x -> y -> y < x;

public static void main(String[] args) {

List<Integer> integerList = Arrays.asList(5, 6, 7, 23, 4, 5645, 6, 1223,

44453, 60182, 2836, 23993, 1);

System.out.println(quickSort(integerList));

}

private static List<Integer> quickSort(List<Integer> integerList) {

if (integerList.isEmpty()) return new ArrayList<>();

return Stream.concat(Stream.concat(quickSort(integerList.stream().skip(1)

.filter(smallerThan.apply(integerList.get(0)))

.collect(Collectors.toList())).stream(),

Stream.of(integerList.get(0))), quickSort(integerList.stream().skip(1)

.filter(smallerThan.apply(integerList.get(0)).negate())

.collect(Collectors.toList())).stream()).collect(Collectors.toList());

}

}

Papildomai apie funkcinį programavimą galite paskaityti čia:

1. Oracle (2019). *Lambda Expressions*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html.>

*8 užduotis*. NAUDOJANT FUNKCINIO PROGRAMAVIMO PRINCIPUS REALIZUOKITE SĄRAŠO GRUPAVIMO, ELEMENTŲ SKAIČIAVIMO IR RŪŠIAVIMO ALGORITMĄ.

**Pavyzdys.** Algoritmas Java programavimo kalba:

package javafunctional;

import java.util.Arrays;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

import java.util.function.Function;

import java.util.stream.Collectors;

public class JavaGroupCountSortFunctional {

public static void main(String[] args) {

List<String> items = Arrays.asList("Tom", "Dave", "John", "Mary",

"Steve", "Tom", "Mary");

Map<String, Long> result = items.stream()

.collect(Collectors.groupingBy(Function

.identity(), Collectors.counting()));

Map<String, Long> finalMap = new LinkedHashMap<>();

result.entrySet().stream().sorted(Map.Entry.<String, Long>comparingByKey())

.forEachOrdered(e -> finalMap.put(e.getKey(), e.getValue()));

System.out.println(finalMap);

}

}

**3.** **Kompetencija. Kurti tipinę programinę įrangą.**

* 1. **Mokymosi rezultatas. Konfigūruoti pagrindinius įrankius, reikalingus projekto paleidimui, ir įdiegti projektui reikalingus paketus.**
     1. **Tema. NodeJS ir Node Package Manager (NPM).**

*1 užduotis*. APIBŪDINKITE „NODE PACKAGE MANAGER“ (NPM) VALDYMO SISTEMOS FUNKCIJAS NODE.JS APLINKOJE.

„Node Package Manager“ (NPM) valdymo sistemos funkcijos Node.js aplinkoje:

* “NPM” veikia kaip „Node.js“ aplinkos paketų / modulių internetinė saugykla, esanti npmjs.com puslapyje.
* Šis įrankis veikia kaip komandų eilutės įrankis paketų, „Node.js“ paketų versijos valdymo ir priklausomybės valdymo įdiegimui. “NPM” yra įdiegiama kartu su „Node.js“.
* NPM pagalba atsisiųsti Node.js moduliai leidžia naudoti jau sukurtus įrankius kuriamame projekte.

Papildomai skaitykite:

NPM documentation. Features of NPM: <https://docs.npmjs.com/about-npm/> , <https://www.npmjs.com/package/feature>

*2 užduotis*. KOKIUS VEIKSMUS ATLIEKA NPM INIT IR NPM INSTALL KOMANDOS SUKURIANT PROJEKTĄ NAUDOJANT NPM VALDIKLĮ?

NPM komandų aprašymas:

* npm init *–* ši komanda sukuria projekto konfiguracinį package.json failą „Node.js“ šakniniame kataloge.
* npminstall *-* ši komanda įdiegia paketą(-us) į lokalų node\_modules katalogą. Privalomi paketai nurodyti konfiguracijos faile package.json (priklausomybės (angl. Dependencies) skiltyje).

Papildomai skaitykite:

NPM documentation. CLI commands: <https://docs.npmjs.com/cli/init> , <https://docs.npmjs.com/cli/install>

*3 užduotis.* PARAŠYKITE KAM YRA SKIRTAS NPM VALDYMO SISTEMOS MODULIS (ANGL. “MODULE”) IR PAKETAS (ANGL. “PACKAGE”).

NPM valdymo sistemos modulio (angl. “module”) ir paketo (angl. “package”) aprašymai:

* Paketas (angl. “package”) yra failas arba katalogas, aprašytas package.json faile. NPM paketą gali sudaryti keli moduliai.
* Modulis (angl. “module”) yra vienas „JavaScript“ failas arba katalogas, turintis nurodytą funkcionalumą. Modulių nereikia aprašyti package.json faile.

Papildomai skaitykite:

NPM documentation. Packages and modules: <https://docs.npmjs.com/about-packages-and-modules>

* + 1. **Tema. NodeJS aplinkos konfigūravimas ir valdymas.**

*4 užduotis*. NURODYTAS PAVYZDINIS KONFIGŪRACINIO PACKAGE.JSON FAILO TURINYS.

Sprendimas:

"foo" – versijos nuo 1.17.1 iki 2.0.0 (neįskaitant)

"bar" - versijos nuo 11.0.0 iki 12.0.0 (neįskaitant)

"boo" – visos galimos versijas

"thr" - versijos nuo 1.2.3-beta.2 iki 1.3.0 (neįskaitant)

"pow" - versijos nuo 0.0.0 iki 0.1.0 (neįskaitant)

Papildomai skaitykite:

NPM documentation. The semantic versioner for NPM: <https://docs.npmjs.com/misc/semver>

*5 užduotis*. KAS YRA ĮRAŠOMA Į PACKAGE-LOCK.JSON FAILĄ?

*Package-lock.json faile yra* aprašomas tikslus paketų medis, kuris buvo sugeneruotas taip, kad vėlesni įdiegimai sugeba generuoti vienodus medžius, nepriklausomai nuo tarpinių priklausomybės atnaujinimų.

Papildomai skaitykite:

NPM documentation. NPM configuration files: <https://docs.npmjs.com/files/package-lock.json>

*6 užduotis*. PAPILDYKITE PACKAGE.JSON KONFIGURACINIO FAILO TURINĮ NPM KOMANDOS TRUMPINIU (ANGL. „SCRIPT“), KURIS PRADĖTŲ NODE.JS PROJEKTO FAILĄ *SERVER.JS.* KOMANDOS TRUMPINYS TURI BŪTI PALEIDŽIAMAS NAUDOJANT.

NPM RUN SERVER-START **ARBA** NPM RUN-SCRIPT SERVER-START

{

"name": "nodejs-project",

"version": "0.2.0",

"description": "test-description",

"main": "index.js",

"scripts": {

" server-start ": "node server.js"

},

"dependencies": {

"express": "^4.13.3"

},

"author": "Name Surname",

"license": "MIT"

}

Papildomai skaitykite:

NPM documentation. Run arbitrary package scripts: <https://docs.npmjs.com/cli/run-script>

*7 užduotis*. UŽRAŠYKITE KOMANDĄ, KURI ĮDIEGTŲ 0.1.2 VERSIJOS NPM PAKETĄ "FOO-BAR" Į PROJEKTO APLINKĄ BEI ŠĮ PAKEITIMĄ NURODYTŲ REIKALAUJAMŲ SUSIETŲ PAKETŲ KŪRĖJŲ SKILTYJE (ANGL. „DEVELOPER DEPENDENCIES“).

Komandų pasiškinimas:

npm install arba npm i arba npm add – komanda įdiegia norimą paketą

foo-bar@0.1.2 – nurodytas paketo pavadinimas bei versijos numeris

--save-devarba-D– komandos nurodymas paketą išsaugoti reikalaujamų susietų paketų kūrėjų skiltyje (angl. „developer dependencies“)

Galimi 6 teisingų atsakymų variantai:

npm install foo-bar@0.1.2 --save-dev

arba

npm i foo-bar@0.1.2 --save-dev

arba

npm add foo-bar@0.1.2 --save-dev

arba

npm install foo-bar@0.1.2 -D

arba

npm i foo-bar@0.1.2 -D

arba

npm add foo-bar@0.1.2 -D

Papildomai skaitykite:

NPM documentation. CLI commands: <https://docs.npmjs.com/cli/install>

*8 užduotis.* PARSISIŲSKITE IR ĮSIDIEKITE NAUJAUSIĄ NODEJS DARBINĘ APLINKĄ IŠ OFICIALAUS TINKLAPIO.

ESLint, Babel ir Webpack įrankiai yra JavaScript moduliai (arba kitaip – įrankiai), jie turi būti diegiami iš NodeJS modulių repozitorijos (komandinės eilutės panaudojimo atveju), tad reikia įsideigti NodeJS aplinką, kurios pagalba bus diegiami minėti moduliai ir paleidžiamos komandinės eilutės norimam rezultatui pasiekti.

NodeJS aplinka turi pagalbinius įrankius JavaScript moduliams įdiegti ir automatiškai atnaujinti. Vienas jų – NPM (Node package manager). Pateiktų užduočių metu būtent naudosime NPM įrankį moduliams įdiegti ir juos valdyti.

Kad galėtume Windows aplinkoje patogiai naudotis NPM įrankiu, reikia papildomai įsidiegti Windows konsolės valdymo programą/emuliatorių. Yra nemažai alternatyvų (<https://www.slant.co/topics/1552/~best-terminal-emulators-for-windows>), tačiau vienas geriausių ir patogiausių sprendimų yra “Windows Power Shell” programa, kuri suteikia Windows konsolės valdymo galimybę, analogišką Linux terminalui. T.y. “Windows Power Shell” nereikalauja papildomų konfigūracijų ar kitokių papildomų žingsnių norint įsidiegti ir naudoti NPM modulius.

*10 užduotis*. IŠTESTUOKITE ĮDIEGTĄ NPM JAVASCRIPT MODULIŲ TVARKYKLĘ IŠKVIEČIANT NPM VERSIJOS SPAUSDINIMĄ „WINDOWS POWER SHELL“ KOMANDINĖS EILUTĖS VALDYMO ĮRANKYJE.

NodeJS darbinė aplinka leidžia ne tik parsisiųsti ir paleisti JavaScript modulius, bet ir kurti *backend* tipo programinę įrangą. Norint parsisiųsti ir įsidiegti NodeJS darbinę aplinką reikia eiti į <https://nodejs.org/en/> ir parsisiųsti siūlomą NodeJS versiją, kuri suderinama su darbine aplinka. Rekomendacija naudoti LTS (Long Term Support) versiją.

**Parsisiuntus NodeJS programinę įrangą, pradedame diegimo žingsnius:**

1. Dukart paspaudus ant NodeJS programinės įrangos ikonėlės pasileis įdiegimo vedlys;
2. Atsidariusiame lange spaudžiame „Next“. Jeigu minėtas mygtukas neaktyvus (pilkas), reikia palaukti kelias akimirkas;
3. Toliau sutinkame su licencijavimo taisyklėmis – pažymime opciją „I accept the terms..“ ir spaudžiame „Next“;
4. Sekančiame lange pasirenkame kur norime įdiegti NodeJS darbinę aplinką. Siūloma numatyta diegimo vieta dažniausiai tinkama, tad spaudžiame „Next“;
5. Sekančiame lange pasitikriname kad visos NodeJS teikiamos galimybės bus įdiegtos (nėra pažymėta raudonu X) ir spaudžiame „Next“;
6. Sekančiame lange pasirenkame „Install“;
7. Pasibaigus įdiegimui, pasirenkame „Finish“.

NodeJS naudojimo vadovą ir dokumentaciją galime rasti čia: <https://nodejs.org/dist/latest-v10.x/docs/api/>

Pasibaigus NodeJS įdiegimui testuojame įdiegtą aplinką:

1. Pasileidžiame „Windows power shell“ komandinės eilutės aplinką;
2. Paleidžiame komandą „$ node -v“, rezultatas „v10.16.0“;
3. Paleidžiame komandą „$ npm -v“, rezultatas „6.9.0“.

*11 užduotis*. PATIKRINKITE AR NODEJS TEISINGAI INTERPRETUOJA JAVASCRIPT KODĄ SUKURIANT NAUJĄ DARBINĮ FAILĄ “hello.js” SU TOKIU JAVASRIPT KODU: “CONSOLE.LOG(‘Hello’);” IR PALEIDŽIANT NODEJS INTERPRETATORIŲ NURODANT SUKURTĄ hello.js FAILĄ.

Sukuriame hello.js failą darbinėje aplinkoje:

1. Pasileidžiame arba naudojame jau iš anksčiau paleistą „Windows power shell“ komandinės eilutės aplinką;
2. Sukuriame naują darbinį failą hello.js;
3. Atsidarome sukurtą hello.js failą su editoriumi;
4. Atsidariusiame lange įvedame programinį kodą: „console.log(`Hello`);“;
5. Išsaugome pakeitimus;
6. Windows Power Shell Paleidžiame NodeJS interpretatorių nurodant sukurtą hello.js failą: „$ node hello.js“.

**3.2. Mokymosi rezultatas. Diegti ir konfigūruoti įrankius kodo transformavimui, pakavimui ir optimizavimui.**

**3.2.1. Tema. ESLint įrankio diegimas ir naudojimas.**

*1 užduotis.* ĮSIDIEKITE ESLINT MODULĮ BEI IR INICIJUOKITE JO STANDARTINĘ KONFIGŪRACIJĄ NAUDOJANTIS NPM ĮRANKIU IR TAIKANTIS Į PROJEKTO DARBINĘ APLINKĄ.

Pasileidžiame „Windows power shell“ komandinės eilutės valdymo aplinką ir atliekame veikmus:

1. Naviguojame iki projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Sukuriame darbinį JavaScript failą: „index.js“;
3. Sukurtame faile įdedame JavaScript kodą: „console.log('Hello');“;
4. Inicijuojame NPM projektą: „$ npm init“:
   1. Atsakome į klausimus ir po kiekvieno pasirinkimo spaudžiame „Enter“;
   2. Pasiekus paskutinį žingsnį „Is this OK (yes)“ – spaudžiame „Enter“.
5. Įvykdome NPM komandą ESLint diegimui: „$ npm install eslint --save-dev“;
6. Inicijuojame ESLint standartinę konfigūraciją: „$ ./node\_modules/.bin/eslint --init“ ir atsakome į klausimus su klaviatūros rodyklių klavišų pagalba:
   1. K: „How would you like to use ESLint“ – A: „To check syntax only“;
   2. K: „What type of modules does your project use?“ – A: „None of these“;
   3. K: „Which framework does your project use?“ – A: „None of these“;
   4. K: „Where does you code run?“ – A: „Node“;
   5. K: „What format do you want your config file to be in?“ – A: „JSON“;
7. Turime gauti tokią failų struktūrą darbiniame aplanke:
   1. .eslintrc.json
   2. index.js
   3. node\_modules (katalogas)
   4. package.json
   5. package-lock.json

*2 užduotis.* SUKONFIGŪRUOKITE ESLINT DARBUI SU INTERNETINĖMIS NARŠYKLĖMIS, NODEJS IR ES6 APLINKOMIS BEI PRITAIKYKITE PROGRAMAVIMO STILIAUS TAISYKLES: INDENTACIJA – 2 LYGIAI, KABLIATAŠKIS – ISPĖJIMAS, JAVASCRIPT „CONSOLE.LOG“ – LEIDŽIAMA.

Vadovaujantis ESLint dokumentacija pateikta adresu: <https://eslint.org/docs/user-guide/configuring> atliekame ESLint konfigūravimo darbus:

1. Naudojantis komandinės eilutės valdymo įrankiu arba IDE įrankiu (vscode, sublime, atom ir pan) atsidarome praeitame žingsnyje sukurtą “.eslintrc.json” failą, kuriame turi būti toks (ar panašus) turinys:

.eslintrc.json

{

"env": {

"es6": true,

"node": true

},

"globals": {

"Atomics": "readonly",

"SharedArrayBuffer": "readonly"

},

"parserOptions": {

"ecmaVersion": 2018

},

"rules": {

}

}

2. Toliau pridedame papildomas taisykles (rules) patenkinti užduoties sąlygas:

**.eslintrc.json**

...

"rules": {

"indent": [

"error",

2

],

"linebreak-style": [

"error",

"unix"

],

"quotes": [

"error",

"double"

],

"semi": [

"warn",

"always"

],

"no-console": "off"

}

...

3. Patestuojame ESLint komandą: „$ npx eslint .\index.js“. Turime gauti panašią žinutę:

1:13 error Strings must use doublequote quotes

✖ 1 problem (1 error, 0 warnings)

1 error and 0 warnings potentially fixable with the `--fix` option.

4. Pataisome „index.js“ faile esantį kodą (iš viengubų kabučių padarome dvigubas): „console.log("Hello");“ ir paleidžiame vėl ESLint, tačiau šiuo metu jau nebeturi būti jokio pranešimo.

*3 užduotis.* IŠTESTUOKITE JAVASCRIPT KODĄ SU ESLINT BEI IŠTAISYKITE PASTEBĖTAS KLAIDAS JEIGU JOS YRA APTINKAMOS ESLINT ĮRANKIO.

Užduoties JAVASCRIPT kodas:

function suma(x,y,t) {

    return t + ( x + z );

}

suma(20,30, 'Suma: ')

Reikalaujant užduoties sąlygoms atliekame veiksmus:

1. Pakeičiame „index.js“ failo turinį į nurodytą užduotyje;
2. Paleidžiame ESLint testavimo įrankį: „$ npx eslint .\index.js“;
3. Turime gauti panašų pranešimą apie aptiktas klaidas:

2:1 error Expected indentation of 2 spaces but found 4 indent

5:14 error Strings must use doublequote quotes

5:23 warning Missing semicolon semi

✖ 3 problems (2 errors, 1 warning)

2 errors and 1 warning potentially fixable with the `--fix` option.

1. Sekantis žingsnis - pašalinti pastebėtas klaidas. Tai galime padaryti dviem būdais:
   1. Rankiniu būdu – su naudojamu kodo editoriumi (IDE) atsidarome „index.js“ failą ir jį redaguojame/pašaliname pastebėtus trūkumus;
   2. Automatiniu būdu: paleidžiame ESLint komandą – „npx eslint --fix .\index.js“
2. Paleidžiame vėl ESLint, tačiau šiuo metu jau nebeturi būti jokio pranešimo.

**Pastaba:** Automatinis klaidų šalinimas veiksmingas tik tuo atveju jeigu tai yra kodavimo standartų klaidos (trūkstamas kabliataškis, neteisingos kabutės ar nenaudojami kintamieji ir pan.), kur kodo logikos problemos jau turi būti sprendžiamos pačio programuotojo – rankiniu būdu.

*4 užduotis.* PARUOŠKITE ESLINT DARBUI SU IDE – UŽTIKRINKITE KAD EGZISTUOJA KONFIGŪRACINIAI FAILAI (PRIKLAUSOMAI NUO PASIRINKTO IDE) IR ESLINT EGZISTUOJA PROJEKTO DARBINĖJE APLINKOJE.

Norint atlikti šią užduotį reikia atlikti pakeitimą ESLint konfigūracijoje ir užtikrinti, kad naudojamas IDE turi ESLint įskiepį (plugin):

1. Redagavimui vėl atsidarome „.eslintrc.json“ failą ir pridedame papildomai keletą konfigūracinių eilučių:

.eslintrc.json

{

"env": {

...

},

"extends": "eslint:recommended",

"globals": {

...

},

"parserOptions": {

...

},

"rules": {

...

"no-inline-comments": "off",

"eqeqeq": "warn"

}

}

1. Toliau vadovaujantis IDE editoriaus nurodymais, reikia susikonfigūruoti ESLint naudojimą:
   1. **JetBrains Idea**: <https://www.jetbrains.com/help/idea/eslint.html>
   2. **SublimeText**: <https://packagecontrol.io/packages/SublimeLinter-eslint>
   3. **VSCode**: <https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>
   4. **Atom**: <http://www.acuriousanimal.com/2016/08/14/configuring-atom-with-eslint.html>
   5. **Apibendrintas ESLint naudojimas:** <https://medium.com/alturasoluciones/eslint-basic-configuration-18b2109d98ec>

**3.2.2. Tema. Babel įrankio diegimas ir naudojimas.**

*5 užduotis.* ĮDIEKITE IR PARUOŠKITE DARBBUI BABEL JS MODULĮ NAUDOJANTIS NPM ĮRANKIU (ĮDIEGTI BABEL, PARUOŠTI DARBINĖS APLINKOS APLANKUS).

Naudodamiesi IDE ar komandinės eilutės interpretatoriumi atliekame tokius veiksmus:

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Paleidžiame Babel ir jo komponentų įdiegimo komandą: „$ npm install --save-dev @babel/cli @babel/core“;
3. Patikriname ar NPM „package.json“ faile atsirado Babel versijos konfigūracija:

package.json

{

"name": "js-learn",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"devDependencies": {

"@babel/cli": "^7.5.5",

"@babel/core": "^7.5.5",

"eslint": "^6.1.0"

}

}

1. Sukuriame 2 naujus katalogus projekto darbinėje aplinkoje: „src“ – naudojamas talpinti pradinį JavaScript kodą ir „dist“ – katalogas skirtas talpinti transformuotą kodą:

darbinis\_katalogas

|- **src**\

|- node\_modules\

|- **dist**\

|- index.js

|...

1. Pasitikriname Babel versiją: „$ npx babel --version“. Turime gauti atsakymą koks matomas „package.json“ versijos konfigūracijoje:

7.5.5 (@babel/core 7.5.5)

*6 užduotis.* IŠTESTUOKITE ĮDIEGTĄ BABEL MODULĮ SU PATEIKTU JAVASCRIPT ES6 KODU (PALEISKITE BABEL KOMANDĄ DUOTAM JAVASCRIPT KODUI IR PATIKRINKITE AR SUSIKURIA ATITINKAMI FAILAI DARBINĖJE APLINKOJE) BE TRANSFORMACIJŲ ATLIKIMO.

export function suma(x, y) {

return x + y;

}

var nums = [1, 2, 3];

var doubleNums = nums.map((e) => e \* 2);

var sum = nums.map((e) => suma(e + 1, e \* 10));

console.log(doubleNums);

console.log(sum);

**Užduoties atlikimo eiga:**

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Naujai sukurtame kataloge „src“ sukuriame naują JavaScript failą „app.js“;
3. Į naujai sukurtą „app.js“ įkeliame užduoties JavaScript kodą;
4. Paleidžiame Babel įrankį: „$ npx babel src -d dist“;
5. Pasitikriname ar naujai sukurtame kataloge „dist“ atsirado naujas failas „app.js“ su identišku turiniu failui kuris yra „src/app.js“;

*7 užduotis.* ĮDIEKITE BABEL TRANSFORMACIJŲ NUSTATYMUS VADOVAUJANTIS OFICIALIA DOKUMENTACIJA (<https://babeljs.io/en/setup/#installation>) IR PAKARTOTINAI PALEISKITE BABEL MODULĮ ANKSČIAU NURODYTAM JAVASCRIPT KODUI GAUTI TRANSFORMUOTĄ REZULTATĄ.

"use strict";

Object.defineProperty(exports, "\_\_esModule", {

value: true

});

exports.suma = suma;

function suma(x, y) {

return x + y;

}

var nums = [1, 2, 3];

var doubleNums = nums.map(function (e) {

return e \* 2;

});

var sum = nums.map(function (e) {

return suma(e + 1, e \* 10);

});

console.log(doubleNums);

console.log(sum);

**Užduoties atlikimo eiga:**

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Įdiegiame papildomus Babel komponentus, kurie padės atlikti transformacijas: „$ npm install @babel/preset-env --save-dev“;
3. Sukuriame naują Babel konfigūracinį failą: „.babelrc“
4. Įterpiame naują Babel konfigūraciją ir aktyvuojame Babel komponentą su tokiu failo turiniu:

.babelrc

{

"presets": ["@babel/preset-env"]

}

1. Pakartotinai paleidžiame Babel kompiliavimo komandą: „$ npx babel src -d dist“;
2. Komandinėje eilutėje turėtume matyti komandos statusą: „Successfully compiled 1 file with Babel.“
3. Patikriname failą esantį „dist/app.js“ ar jis atitinka tai kas nurodyta prie užduoties sąlygos. Galutinis rezultatas turi skirtis nuo „src/app.js“ failo turinio.

*8 užduotis.* SUKONFIGŪRUOKITE BABEL ESLINT PALAIKYMĄ IR PARUOŠKITE WEBPACK NAUDOJIMUI. IŠTESTUOTI ES6 KODĄ SU ESLINT PRIEŠ BABEL TRANSFORMACIJĄ.

**Užduoties atlikimo eiga:**

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Paleidžiame ESLint įrankį kodui kuris yra „src/app.js“: „$ npx eslint .\src\app.js“;
3. Turime gauti žinutę, kad ESLint nesupranta ES6 direktyvų import ir export:

1:1 error Parsing error: 'import' and 'export' may appear only with 'sourceType: module'

✖ 1 problem (1 error, 0 warnings)

1. Įdiegiame Babel ESLint palaikymą: „$ npm install babel-eslint --save-dev“;
2. Sukonfigūruojame ESLint pridėdami papildomą eilutę į „.eslintrc.json“ failą:

.eslintrc.json

{

"env": {

...

},

"globals": {

...

},

"parser": "babel-eslint",

....

}

1. Pakartotinai paleidžiame ESlint komandą: „$ npx eslint .\src\app.js“ ir šį kartą turime gauti kiek kitokią žinutę:

2:1 error Expected indentation of 2 spaces but found 4 indent

✖ 1 problem (1 errors, 0 warnings)

1 error and 0 warnings potentially fixable with the `--fix` option.

1. Paskutinis žingsnis būtų leisti ESLint sutaisyti surastas klaidas automatiškai: „$ npx eslint --fix .\src\app.js“;
2. Iškvietus paskutinę komandą jokių paildomų pranešimų nebeturi rodyti ir pastebėtos klaidos faile „src/app.js“ turi būti ištaisytos.

**3.2.3. Tema. Webpack įrankio diegimas ir naudojimas.**

*9 užduotis.* NAUDOJANTIS NPM ĮRANKIU ĮDIEKITE WEBPACK BEI CLI IR SERVER KOMPONENTUS IR PARUOŠKITE DARBUI DARBINĖJE APLINKOJE.

**Užduoties atlikimo eiga:**

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Įdiegti Webpack ir jo server/cli komponentus: „$ npm install --save-dev webpack webpack-dev-server webpack-cli“;
3. Sukonfigūruoti Webpack serverį. NPM konfigūracijoje „package.json“ reikia papildomai pridėti konfigūracinę eilutę

**package.json**

"scripts": {

"start": "webpack-dev-server --config ./webpack.config.js --mode development",

...

1. Sekantis žingsnis, šakniniame projekto aplanke sukurti naują failą „webpack.config.js“:

darbinis\_katalogas

|- src\

|- node\_modules\

|- dist\

|- **webpack.config.js**

|- package.json

|- ...

1. Naujo failo turinys turi būti toks:

**webpack.config.js**

module.exports = {

entry: './src/index.js',

output: {

path: \_\_dirname + '/dist',

publicPath: '/',

filename: 'bundle.js'

},

devServer: {

contentBase: './dist'

}

};

**Paaiškinimai:**

* Pagrindinis failas kuris bus iškviečiamas (main failas) nurodomas ties „entry“ opcija. Jis bus panaudotas surasti likusius failus kurie turi būti susieti (import);
* Toliau viskas kas bus surasta per „src/index.js“ failą bus sudėta į vientisą naują failą pavadinimu „bundle.js“ ir bus patalpinta „dist“ kataloge;
* Tuo tarpu paleidus Webpack serverį, „dist“ katalogas bus naudojamas kaip šakninis katalogas paleisti aplikacijai.

1. Sukurti naujus failus:

**src/hello.js:**

export const hello = subject => console.log(`Hello ${subject}!`);

**src/index.js:**

import { hello } from "./hello";

hello("JS Dev!!!");

**dist/index.html:**

<!DOCTYPE html>

<html>

<head>

<title>Hello Webpack</title>

</head>

<body>

<div>

<h1>Hello Webpack</h1>

</div>

<script src="./bundle.js"></script>

</body>

</html>

*10 užduotis.* SUKONFIGŪRUOTI WEBPACK TAIP, KAD PRADĖTŲ NAUDOTI ESLINT IR BABEL MODULIUS.

**Užduoties atlikimo eiga:**

1. Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;
2. Papildomai reikia įdiegti Babel-loader: „$ npm install --save-dev babel-loader“;
3. Sekantis žingsnis pridėti papildomą Webpack konfigūraciją Babel:

**webpack.config.js:**

module.exports = {

entry: './src/index.js',

**module: {**

**rules: [{**

**test: /\.(js)$/,**

**exclude: /node\_modules/,**

**use: ['babel-loader']**

**}]**

**},**

**resolve: {**

**extensions: ['\*', '.js']**

**},**

output: {

path: \_\_dirname + '/dist',

publicPath: '/',

filename: 'bundle.js'

},

devServer: {

contentBase: './dist'

}

};

**Paaiškinimai:**

* Pridėtos papildomos konfigūracijos direktyvos sako Webpack, kad bus pradėti naudoti moduliai su papildomais nustatymais;
* Webpack turės ignoruoti visus JavaScript failus esančius „node\_modules/“ kataloge;
* Visiems kitiems projekto failams, kurie turi plėtinį „.js“ pritaikyti babel-loader.

1. Pridedame ESLint komponentus kartu prie Webpack + Babel įdiegiant ESLint loader: „$ npm install --save-dev eslint-loader“;
2. Dar kartą pakoreguojame „webpack.config.js“ failą ir prie „module > rules“ pridedame „eslint-loader“:

**webpack.config.js:**

module.exports = {

entry: './src/index.js',

module: {

rules: [{

test: /\.(js)$/,

exclude: /node\_modules/,

**use: ['babel-loader', 'eslint-loader']**

}]

},

resolve: {

extensions: ['\*', '.js']

},

output: {

path: \_\_dirname + '/dist',

publicPath: '/',

filename: 'bundle.js'

},

devServer: {

contentBase: './dist'

}

};

*11 užduotis.* ATLIKUS WEBPACK KONFIGŪRAVIMO DARBUS, PALEISKITE WEBPACK SERVER PROJEKTĄ:

**Užduoties atlikimo eiga:**

1.Naviguojame iki anksčiau sukurto projekto darbinės aplinkos: „$ cd kelias/iki/darbinės/aplinkos“;

2.Paleidžiame komandą: „$ npm start“ ir turime gauti panašų rezultatą:

> js-learn@1.0.0 start C:\Users\xxx\Desktop\js-learn

> webpack-dev-server --config ./webpack.config.js --mode development

i ｢wds｣: Project is running at **http://localhost:8081/**

i ｢wds｣: webpack output is served from /

i ｢wds｣: Content not from webpack is served from ./dist

i ｢wdm｣: Hash: 938f06c6f34e25ca61ab

Version: webpack 4.38.0

Time: 368ms

Built at: 2019-07-26 5:28:50 PM

Asset Size Chunks Chunk Names

bundle.js 360 KiB main [emitted] main

Entrypoint main = bundle.js

[0] multi (webpack)-dev-server/client?http://localhost ./src/index.js 40 bytes {main} [built]

[./node\_modules/ansi-html/index.js] 4.16 KiB {main} [built]

[./node\_modules/html-entities/index.js] 231 bytes {main} [built]

[./node\_modules/loglevel/lib/loglevel.js] 7.68 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/index.js?http://localhost] (webpack)-dev-server/client?http://localhost 4.29 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.51 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.53 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/createSocketUrl.js] (webpack)-dev-server/client/utils/createSocketUrl.js 2.77 KiB {main} [built][./node\_modules/webpack-dev-server/client/utils/log.js] (webpack)-dev-server/client/utils/log.js 964 bytes {main} [built]

[./node\_modules/webpack-dev-server/client/utils/reloadApp.js] (webpack)-dev-server/client/utils/reloadApp.js 1.63 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/sendMessage.js] (webpack)-dev-server/client/utils/sendMessage.js 402 bytes {main} [built]

[./node\_modules/webpack-dev-server/node\_modules/strip-ansi/index.js] (webpack)-dev-server/node\_modules/strip-ansi/index.js 161 bytes {main} [built]

[./node\_modules/webpack/hot sync ^\.\/log$] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {main} [built]

[./src/hello.js] 65 bytes {main} [built]

[./src/index.js] 52 bytes {main} [built]

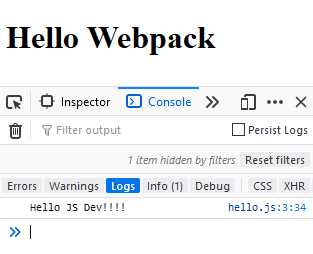
+ 19 hidden modules

**i ｢wdm｣: Compiled successfully.**

3. Paleidus Webpack serverį galime atsidaryti pateiktą nuorodą internetinėje naršyklėje ir pasižiūrėti rezultatą. Adresas matyti komandinės eilutės paleidimo išvestyje (log):

[**http://localhost:8081/**](http://localhost:8081/)

4. Rezultatas:



5. Nuo šiol, keičiant bet kurį projekto failą, Webpack automatiškai atnaujina interneto naršyklės puslapį ir perkompiliuoja visus failus;

6. Norint sustabdyti serverį reikia įvykdyti komandą: „ctrl + c“.

**3.3.** **Mokymosi rezultatas. Analizuoti ir derinti vykdomą kodą, identifikuoti problemas ir jas taisyti.**

**3.3.1. Tema. Vykdomo kodo analizės įrankiai.**

*1 užduotis*. APRAŠYKITE CHROME DEVTOOLS PANAUDOJIMO GALIMYBES.

„Chrome DevTools“ yra žiniatinklio kūrėjų įrankių rinkinys, įdiegtas „Google Chrome“ naršyklėje. „DevTools“ gali padėti redaguoti HTML puslapius, greitai nustatyti HTML, CSS ar JavaScript problemas, suteikia galimybę redaguoti HTML ir CSS kodą realiu laiku, atlikti JavaScript kodo vykdymo testus, analizuoti siunčiamų ar gaunamų užklausų duomenis. Tai padeda kurti geresnes svetaines, kurios veikia greičiau ir naudoja mažiau resursų.

Papildomai skaitykite:

Tools for web Developers. Crome DevTools. <https://developers.google.com/web/tools/chrome-devtools/?utm_source=dcc&utm_medium=redirect&utm_campaign=2018Q2>

*2 užduotis*. IŠVARDINKITE CHROME DEVTOOLS SKYDELIUS IR TRUMPAI APRAŠYKITE JŲ PASKIRTĮ.

Chrome DevTools skydeliai.

* Elementų skydelis (angl. “Elements”) – skirtas peržiūrėti ir pakeisti HTML bei CSS.
* Konsolės skydelis (angl. “Console”) – skirtas rodyti klaidų pranešimus arba JavaScript pranešimus.
* Šaltinių skydelis (angl. “Sources) – skirtas parodyti naudojamus resursus HTML puslapyje, testuoti “JavaScript” kodą, pakeisti ir paleisti “JavaScript” fragmentus, atsisiųsti resursus.
* Tinklo skydelis (angl. “Network”) – skirtas peržiūrėti siunčiamų resursų procesus ir trukmę, patikrinti, kad ištekliai iš tikrųjų būtų įkeliami arba atsisiunčiami. Taip jame galima matyti kiekvieno išteklio informaciją: pvz. HTTP antraščių, turinio, dydžio ir pan.
* Našumo skydelis (angl. “Performace”) – skirtas rasti būdus, kaip pagerinti apkrovos ir vykdymo laiką.
* Atminties skydelis (angl. “Memory”) – skirtas profilio atminties naudojimo analizei.
* Taikomosios programos skydelis (angl. “Application”) – skirtas patikrinimui visų susijusių išteklių, įskaitant „IndexedDB“ arba „Web SQL“ duomenų bazes, vietinę ir sesijos saugyklą, slapukus, „Application Cache“, vaizdus, šriftus ir stilių lenteles.
* Apsaugos skydas (angl. “Security”) – skirtas aptikti ir ištaisyti su mišriu turiniu susijusias problemas, sertifikavimo ir saugumo problemas.
* Audito skydelis (angl. “Audits”) – skirtas atlikti puslapio našumo ir pasiekiamumo testus.

Papildomai skaitykite:

Tools for web Developers. Crome DevTools. <https://developers.google.com/web/tools/chrome-devtools/?utm_source=dcc&utm_medium=redirect&utm_campaign=2018Q2>

*3 užduotis*. IŠVARDINKITE KOKIUS VEIKSMUS GALIMA ATLIKTI ANALIZUOJANT HTML DOM MEDĮ PASINAUDOJANT DEVTOOLS ELEMENTŲ SKYDELIU (ANGL. „ELEMENTS“). IŠVARDINKITE BENT 4 PANAUDOJIMO ATVEJUS.

Analizuojant HTML DOM medį DevTools elementų skydelyje, galima:

* Peržiūrėti HTML DOM medį, jo elementus ir struktūrą;
* Redaguoti DOM elementus ir atributus;
* Redaguoti DOM elemento ir jo vaikų elementų HTML kodą;
* Perkelti ar ištrinti DOM elementą;
* Peržiūrėti elemento įvykių klausytojus (angl. Event Listeners);
* Peržiūrėti ar įrašyti HTML komentarus

Papildomai skaitykite:

Tools for web Developers. Edit the DOM. <https://developers.google.com/web/tools/chrome-devtools/inspect-styles/edit-dom>

*4 užduotis*. APRAŠYKITE KAIP GALIME REDAGUOTI CSS KODĄ PASINAUDOJANT CHROME DEVTOOLS? PATEIKITE REDAGAVIMO GALIMYBIŲ PAVYZDŽIŲ. (2-3 PVZ.).

Viena naudingiausių „DevTools“ įrankių rinkinio funkcijų yra gebėjimas redaguoti HTML ir CSS, keisti kodą ir iškarto matyti pakeitimus puslapyje. Ši funkcija leidžia kūrėjui greitai atlikti ir patikrinti galimus tinklalapio pakeitimus.

* Elementų skydelyje galima rasti puslapio elemento HTML ir CSS tiesiog ant jo paspaudus arba skydelyje arba pasinaudojus funkcija „Ištirti“ (angl. „Inspect“) – dešiniojo pelės klavišo meniu punktas. Šiame skydelyje galima matyti visą elemento padėties, dydžio ir kitą stiliaus informaciją.
* Teksto šrifto, spalvos, pozicijos redagavimas ir rezultato matymas realiu laiku.
* Parinkti, koreguoti fono spalvą.
* Pridėti naują CSS stilių.
* Pakeisti, pridėti elemento pavadinimą arba reikšmę.

Papildomai skaitykite:

* Tools for web Developers. Get Started With Viewing And Changing CSS. <https://developers.google.com/web/tools/chrome-devtools/css/>
* Stephen Rollins. Live editing HTML and CSS with Chrome DevTools. <https://www.lucidchart.com/techblog/2018/05/01/live-editing-html-css-chrome-devtools/>

*5 užduotis*. KAM REIKALINGAS NAŠUMO (ANGL. „PERFORMANCE“) SKYDELYJE ESANTIS GREITAVEIKOS ANALIZĖS PROFILIAVIMO ĮRANKIS (ANGL. „PROFILING“)?

Naudojantis šiuo įrankiu galima sužinoti tikslų puslapio atvaizdavimo laiką. Puslapio dažymas yra pikseliu pildymo procesas, kuris dažnai yra brangiausia atvaizdavimo proceso dalis. Norint sužinoti kiek ilgai trunka šis procesas tereikia paleisti “Profiling” testą ir išanalizavus gautus duomenis galima sužinoti kas užtrunka ilgiausiai bei rasti klaidas ir jas ištaisyti.

Papildomai skaitykite:

Tools for web Developers. Performance Analysis Reference. <https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/reference#top_of_page>

*6 užduotis.* APRAŠYKITE ŽINGSNIUS KAIP NAUDODAMIESI ANALIZĖS ĮRANKIO AUDITO SKYDELIU (ANGL. „AUDITS“) IŠANALIZUOTUMĖTE PUSLAPIO NAŠUMĄ?

Pirmiausia reikia įjungti Chrome DevTools. Iš DevTools meniu pasirinkti Audito skydelį (angl. “Audits”). Audito įrankis turi įvairias parinktis: įrenginio pasirinkimas, norimos analizės tipo pasirinkimas ir tinko sąlygos. Šiuo atveju reikia pasirinkti: kompiuteris, našumas ir simuliacija (angl. „Desktop”, „Performance“ ir „Simulated Fast 3G, 4x CPU slowdown“). Toliau spausti „Atlikti auditą“. Kai baigiasi testas gausite ataskaitą. Ataskaitoje bus galima pamatyti puslapio našumo, pasiekiamumo bei kitą analizės informacija.

Papildomai skaitykite:

Tools for web Developers. Lighthouse. <https://developers.google.com/web/tools/lighthouse/>

**3.3.2. Tema. Programos derinimas.**

*7 užduotis*. APRAŠYKITE IR PALYGINKITE KODO DERINIMO BŪDUS: CONSOLE.LOG() NAUDOJIMĄ IR STABDOS TAŠKUS (ANGL. BREAKPOINTS) .

Abu būdai atlieka savo funkciją – padeda surasti ir pašalinti kodo klaidas, suprasti kas po ko vyksta programiniame kode. JavaScript komanda console.log() yra paprastesnis, bet mažiau informacijos suteikiantis kodo testavimo būdas, tinkamas, kai norime sužinoti vieno ar kelių kintamųjų reikšmes tam tikru programos vykdymo metu. Stabdos taškai gali lėčiau, bet daug išsamiau padėti patikrinti programos veikimo algoritmą. Jie leidžia pristabdyti vykdomą kodą ir ištirti visus duomenis būtent sustabdytu metu, stebint kiekvieną vykdomą kodo eilutę.

Naudojant console.log() komandą reikia JavaScript kode surasti atitinkamą algoritmo vietą ir įterpti console.log() pranešimus. Po to reikia atnaujinti žiniatinklio puslapį, kad pamatyti pranešimus DevTools konsolės skydelyje (angl. Console).

Console.log() komandoje reikia aiškiai nurodyti kiekvieną kintamąjį ar jo vertę norint jį išvesti į konsolės skydelį.

**Pavyzdys:**

function test() {

var test1 = getNumber1();

console.log('test1:', test1);

}

Naudojant stabdos taškus DevTools parodo visų kintamųjų reikšmes tuo metu, kai kodo vykdymas yra sustabdytas. Tam kad sukurti stabdos taškus, reikia atsidaryti DevTools šaltinių skydelį (angl. Sources), surasti kairėje pusėje esančiame failų medyje reikiamą failą su JavaScript kodu ir jame pažymėti eilučių numerius kairės pelės klavišo paspaudimu.

Papildomai skaitykite:

Tools for web Developers .Get Started with Debugging JavaScript in Chrome DevTools. [https://developers.google.com/web/tools/chrome-devtools/JavaScript/](https://developers.google.com/web/tools/chrome-devtools/javascript/)

*8 užduotis*. KAS ĮVYKS ĮVYKDŽIUS JAVASCRIPT KODĄ, JEI 32 EILUTĖJE PADĖSIME STABDOS TAŠKĄ (ANGL. BREAKPOINT)?

Paleidus kodą visos funkcijos bus įvykdytos iki pažymėto stabdos taško. 32 eilutėje esantis kodas nebus įvykdytas. DevTools visada sustabdys kodo vykdymą prieš įvykdant pažymėtą kodo eilutę ir parodys addend1, addend2 ir sum reikšmes. Toliau kodas nebus vykdomas tol kol nebus pasirinkta “Tęsti scenarijaus vykdymą”. Pasirinkus tęsti, kodas bus toliau vykdomas be jokių pakitimų.

Papildomai skaitykite:

Tools for web Developers. Get Started with Debugging JavaScript in Chrome DevTools. [https://developers.google.com/web/tools/chrome-devtools/JavaScript/](https://developers.google.com/web/tools/chrome-devtools/javascript/)

*9 užduotis.* PATEIKITE STABDOS TAŠKŲ (ANGL. BREAKPOINTS) TIPUS IR JUOS APRAŠYKITE).

Stabdos taškų tipai:

* Įprasti stabdos taškai – naudojami kodo vykdymo sustabdymui pasirinktoje kodo eilutėje nepateikiant jokių sąlygų;
* Sąlyginiai ir logikos stabdos taškai (angl. Conditional line-of-code breakpoints) – naudojami pasirinktoje kodo vietoje, bet tik tada, kai sąlyga yra teisinga. Pvz.: x>5;
* DOM stabdos taškai – naudojami HTML kodo dalyje, kai pakeičiamas arba pašalimas konkretus DOM elementas arba jo vaikiniai elementai;
* XHR/fetch stabdos taškai – naudojami sustabdyti vykdymą kai URL užklausoje yra nurodytas tam tikras tekstas. „DevTools“ sustabdo kodo eilutę, kurioje XHR iškviečia send();
* Įvykių klausytojų stabdos taškai (angl. Event Listener Breakpoints) – naudojame įvykio klausytojo stabdos taškus, kai norime sustabdyti kodo vykdymą po konkretaus įvykio. Galite pasirinkti konkrečius įvykius, pvz., pelės klavišo paspaudimą. Taip pat galima pasirinkti įvykių kategorijas, pvz. visus klaviatūros klavišų paspaudimus;
* Išimties stabdos taškai (angl. Exception) – naudojami kai norime sustabdyti kodo eilutę, kuri sukuria sugaunamą arba nesugaunamą išimtį. (angl. caught or uncaught exception);
* Funkcijos stabdos taškai (angl. Function) – kodo vykdymas sustabdomas, kai iškviečiama nurodyta funkcija.

Papildomai skaitykite:

Tools for web Developers. How To Pause Your Code With Breakpoints In Chrome DevTools.

[https://developers.google.com/web/tools/chrome-devtools/JavaScript/breakpoints](https://developers.google.com/web/tools/chrome-devtools/javascript/breakpoints)

**3.4. Mokymosi rezultatas. Kurti JavaScript programinį kodą ir naudoti jo sintaksę.**

**3.4.1. Tema. Sintaksė, gramatika, duomenų tipai.**

*1 užduotis*. IŠVESKITE PRANEŠIMĄ: „MAN PATINKA WEB TECHNOLOGIJOS“NARŠYKLĖS DIALOGO LANGE IR KONSOLĖJE.

console.log("\"Man labai patinka WEB technologijos\"");

*2 užduotis*. PANAUDODAMI JS PROMPT DIALOGO LANGĄ NARŠYKLĖJE PAKLAUSKITE VARTOTOJO VARDO IR JĮ IŠVESKITE NARŠYKLĖS KONSOLĖJE.

var vardas = prompt("Koks Jūsų vardas ?");

console.log('Vardas: '+vardas);

*3 užduotis*. PANAUDODAMI TYPEOF FUNKCIJĄ NUSTATYKITE IR IŠVESKITE NARŠYKLĖS KONSOLĖJE IŠVARDINŲ DUOMENŲ TIPUS:

1. Studentas

2. 12

3. true

console.log(typeof('Studentas'))

console.log(typeof(12))

console.log(typeof(true))mpt("Koks Jūsų vardas ?");

*4 užduotis*. SUKURKITE KINTAMUOSIUS: KNYGOS PAVADINIMAS, AUTORIUS, PUSLAPIŲ SKAIČIUS, KIEKIS, KAINA.

(var knygosPavadinimas = "Naujos WEB technologijos";

var autorius = "Vardenis Pavardenis";

var puslapiuSkaicius = 250;

var kiekis = 25;

var kaina = 36.65

console.log('Knygos pavadinimas: ' + knygosPavadinimas);

console.log('Autorius: ' + autorius);

console.log('Puslapių skaičius: ' + puslapiuSkaicius);

console.log('Kiekis: ' + kiekis);

console.log('Kaina: ' + kaina);

*5 užduotis. 4 užduotyje* SUKURTŲ KINTAMŲJŲ REIKŠMES PATALPINKITE Į MASYVĄ KNYGA IR IŠVESKITE NARŠYKLĖS KONSOLĖJE.

var knyga = ['Naujos WEB technologijos','Vardenis Pavardenis', 250, 25, 36.65];

console.log(knyga);

*6 užduotis*. SUKURKITE KNYGOS MASYVĄ, KURIS TALPINA ***–***INFORMACIJĄ APIE 5 KNYGAS.

var knygos = [

['Naujos WEB technologijos','Vardenis Pavardenis', 250, 25, 46.65],

['Programuok linksmai','Vardenis Pavardenis', 280, 25, 36.65],

['C++ pagrindai','Vardenis Pavardenis', 260, 25, 56.65],

['PHP praktiniai patarimai','Vardenis Pavardenis', 250, 25, 26.65],

['JS praktikumas','Vardenis Pavardenis', 220, 25, 16.65],

['React pagrindai','Vardenis Pavardenis', 320, 15, 66.65]

];

console.log(knygos[2]);

console.log(knygos[4]);

*7 užduotis*. SUKURKITE KNYGOS OBJEKTĄ, PAGAL *6 užduotyje* PATEIKTĄ KNYGOS DUOMENŲ STRUKTŪRĄ. NARŠYKLĖS KONSOLĖJE IŠVESKITE KNYGOS PAVADINIMĄ IR PUSLAPIŲ SKAIČIŲ.

var knyga = {

knygosPavadinimas: "Naujos WEB technologijos",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 250,

kiekis:25,

kaina: 36.5

};

console.log("Knygos pavadinimas: " + knyga.knygosPavadinimas);

console.log("Puslapių skaičius: " + knyga.puslapiuSkaicius);

*8 užduotis*. SUKURKITE MASYVĄ, KURIS TALPINA 5 KNYGŲ OBJEKTUS PAGAL *6* *užduotyje* SUKURTĄ OBJEKTO STRUKTŪRĄ.

var knygos = [

{

knygosPavadinimas: "Naujos WEB technologijos",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 250,

kiekis:25,

kaina: 36.5

},

{

knygosPavadinimas: "Programuok linksmai",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 350,

kiekis:65,

kaina: 26.5

},

{

knygosPavadinimas: "Naujas IT pasaulis",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 200,

kiekis:15,

kaina: 16.5

},

{

knygosPavadinimas: "Tinklas",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 200,

kiekis:35,

kaina: 46.5

},

{

knygosPavadinimas: "TOP technologijos 2019",

autorius: "Vardenis Pavardenis",

puslapiuSkaicius: 160,

kiekis:65,

kaina: 46.5

},

];

console.log(knygos[0].knygosPavadinimas);

console.log(knygos[1].knygosPavadinimas);

console.log(knygos[2].knygosPavadinimas);

console.log(knygos[3].knygosPavadinimas);

console.log(knygos[4].knygosPavadinimas);

*9 užduotis*. SUKURKITE PROGRAMĄ, KURI NAUDODAMA PROMPT DIALOGO LANGĄ, SURINKTĄ INFORMACIJĄ (KNYGOS PAVADINIMAS, KIEKIS, NUOLAIDOS DYDIS PROCENTAIS) APSKAIČIUOJA KNYGOS PARDAVIMO KAINĄ IR IŠVEDA INFORMACIJĄ NARŠYKLĖS KONSOLĖJE.

var knyga = {var knygosPavadinimas = prompt('Knygos pavadinimas');

var kiekis = parseInt(prompt('Kiekis'));

var kaina = parseFloat(prompt('Kaina'));

var nuolaida = parseInt(prompt('Nuolaidos dydis procentais'));

var suma = kiekis \* kaina;

var nuolaidosDydis = (suma/100) \* nuolaida;

console.log("Knygos pavadinimas: " + knygosPavadinimas);

console.log("Kiekis: " + kiekis);

console.log("Nuolaidos dydis %: " + nuolaida);

console.log("Kaina: " + kaina);

console.log("----------");

console.log("Suma be nuolaidos: " + suma);

console.log("Nuolaida(" + nuolaida + "% ): " + nuolaidosDydis );

suma = suma - nuolaidosDydis;

console.log("Mokejimo suma: " + suma);

**3.4.2. Tema. Operatoriai, salygos sakiniai.**

*10 užduotis*. SUKURKITE PROGRAMĄ, KURI NAUDODAMA PALYGINIMO OPERATORIUS IŠVEDA *5 užduotyje* SUKURTAS KNYGAS, KURIŲ PUSLAPIŲ SKAIČIUS YRA MAŽESNIS UŽ 200, O PAVADINIMĄ SUDARO 15 IR DAUGIAU SIMBOLIŲ.

if (knygos[0].puslapiuSkaicius > 200 && knygos[0].knygosPavadinimas.length > 15)

{

console.log(knygos[0].knygosPavadinimas);

}

if(knygos[1].puslapiuSkaicius > 200 && knygos[1].knygosPavadinimas.length > 15){

console.log(knygos[1].knygosPavadinimas);

}

if(knygos[2].puslapiuSkaicius > 200 && knygos[2].knygosPavadinimas.length > 15){

console.log(knygos[2].knygosPavadinimas);

}

if(knygos[3].puslapiuSkaicius > 200 && knygos[3].knygosPavadinimas.length > 15){

console.log(knygos[3].knygosPavadinimas);

}

if(knygos[4].puslapiuSkaicius > 200 && knygos[4].knygosPavadinimas.length > 15){

console.log(knygos[4].knygosPavadinimas);

}

*11 užduotis*. SUKURKITE PROGRAMĄ, KURI NAUDODAMA LOGINIUS OPERATORIUS IŠVEDA *5 užduotyje* KNYGAS, KURIŲ PUSLAPIŲ SKAIČIUS YRA MAŽESNIS UŽ 200, ARBA DIDESNIS UŽ 500.

if(knygos[0].puslapiuSkaicius < 200 || knygos[0].puslapiuSkaicius> 500)

{

console.log(knygos[0].knygosPavadinimas);

}

if(knygos[1].puslapiuSkaicius < 200 || knygos[1].puslapiuSkaicius > 500){

console.log(knygos[1].knygosPavadinimas);

}

if(knygos[2].puslapiuSkaicius < 200 || knygos[2].puslapiuSkaicius > 500){

console.log(knygos[2].knygosPavadinimas);

}

if(knygos[3].puslapiuSkaicius < 200 || knygos[3].puslapiuSkaicius > 500){

console.log(knygos[3].knygosPavadinimas);

}

if(knygos[4].puslapiuSkaicius < 200 || knygos[4].puslapiuSkaicius > 500){

console.log(knygos[4].knygosPavadinimas);

}

*12 užduotis*. SUKURKITE PROGRAMĄ, KURI KINTAMĄJAM VIRŠELIO SPALVA PRISKIRTĄ ŠEŠIOLIKTAINĘ SPALVOS REIKŠMĘ KONVERTUOJA Į RGB (SPALVOS REIKŠMĘ).

var spalvosKodas = '#ffaadd';

spalvosKodas = spalvosKodas.substr(1);

var rgb = parseInt(spalvosKodas, 16);

var raudona = (rgb >> 16) & 0xFF;

var zalia = (rgb >> 8) & 0xFF;

var melyna = rgb & 0xFF;

console.log("Įvesta spalvos hex reikšmė: " + spalvosKodas);

console.log("Konvertavimo į RGB rezultatas: ");

console.log("Raudona: " + raudona);

console.log("Žalia: " + zalia);

console.log("Melyna: " + melyna);

*13 užduotis*. SUKURKITE SKAIČIUOTUVO PROGRAMĄ, KURIS PER FUNKCIJĄ PROMPT IŠ VARTOTOJO PRIIMA SKAIČIUS, OPERATORIUS(+, -, /, \*, %) IR IŠVEDA REZULTATĄ.

var skai1,skai2,rezultatas,operatorius;

skai1=prompt("Įveskite 1 skaičių");

operatorius=prompt("Įveskite operatorių");

skai2=prompt("Įveskitė 2 skaičių");

if(operatorius=="+"){

rezultatas=Number(skai1)+Number(skai2);

}

else if(!operatorius){

rezultatas="Klaida"

}

else if(!skai1){

rezultatas="Klaida"

}

else if(!skai2){

rezultatas="Klaida"

}

else if(operatorius=="/"){

rezultatas=Number(skai1)/Number(skai2);

}

else if(operatorius=="\*"){

rezultatas=Number(skai1)\*Number(skai2);

}

else if(operatorius=="-"){

rezultatas=Number(skai1)-Number(skai2);

}

else if(operatorius=="%"){

rezultatas=(Number(skai1)/100)\*Number(skai2);

}

console.log(rezultatas);

**3.4.3. Tema. Ciklai.**

*14 užduotis*. NAUDODAMI FOR IR WHILE CIKLUS IŠVESKITE *6 užduotyje* SUKURTĄ KNYGŲ MASYVĄ NARŠYKLĖS KONSOLĖJE.

**WHILE**

var i = 0;

while (i < knygos.length) {

console.log(knygos[i].knygosPavadinimas);

i++;

}

**For**

for (var i = 0; i < knygos.length; i++)

{

console.log(knygos[i].knygosPavadinimas);

}

*15 užduotis*. NAUDODAMI FOR CIKLĄ Į *6 užduotyje* SUKURTĄ KNYGŲ MASYVĄ PRIDĖKITE TRIS NAUJAS KNYGAS, KURIŲ PAVADINIMUS VARTOTOJAS NURODYS PROMPT DIALOGO LANGO PAGALBA NARŠYKLĖS KONSOLĖJE.

for (var i = 0; i < 3; i++)

{

var knygosPavadinimas = prompt('Knygos pavadinimas');

var autorius = prompt('Autorius');

var puslapiuSkaicius = prompt('Puslapiu skaicius');

var kiekis = prompt('Kiekis');

var kaina = prompt('Kaina');

knygos.push({

knygosPavadinimas: knygosPavadinimas,

autorius: autorius,

puslapiuSkaicius: puslapiuSkaicius,

kiekis:kiekis,

kaina: kaina

});

}

for(var i=0; i<knygos.length; i++)

{

console.log(knygos[i].knygosPavadinimas);

}

*16 užduotis*. NAUDODAMI WHILE ARBA FOR CIKLĄ *6 užduotyje* SUKURTAME MASYVE KNYGŲ KAINAS PADIDINKITE 10 % IR PAŠALINKITE PASKUTINĘ KNYGĄ IŠ MASYVO.

knygos.pop();

for(var i=0; i<knygos.length; i++)

{

knygos[i].kaina = knygos[i].kaina \* 1.10;

console.log("Knygos pavadinimas: " + knygos[i].knygosPavadinimas);

console.log("Knygos kaina: "+ knygos[i].kaina);

}

**3.4.4. Tema. Objektai.**

*17 užduotis*. SUKURKITE KNYGŲ KATALOGĄ, KURIAME BŪTŲ 3 KATEGORIJOS IR 5 KNYGOS KIEKVIENOJE KATEGORIJOJE.

var Katalogas={

IT:{

H46DF:{

Metai:"2018",

Pavadinimas:"C++ pagrindai",

puslapiuSkaicius:300,

kiekis: 12,

kaina: 12.52

},

H11DF:{

Metai:"2018",

Pavadinimas:"HTML 5",

puslapiuSkaicius:200,

kiekis: 12,

kaina: 22.52

},

H22F:{

Metai:"2018",

Pavadinimas:"PHP 7",

puslapiuSkaicius:255,

kiekis: 12,

kaina: 32.52

},

H33DF:{

Metai:"2019",

Pavadinimas:"SASS",

puslapiuSkaicius:125,

kiekis: 12,

kaina: 18.52

},

H44DF:{

Metai:"2018",

Pavadinimas:"Programavimas linksmai",

puslapiuSkaicius:360,

kiekis: 12,

kaina: 20.52

}

},

Fantastine:{

H55DF:{

Metai:"2018",

Pavadinimas:"Marsas",

puslapiuSkaicius:220,

kiekis: 12,

kaina: 30.52

},

H66DF:{

Metai:"2018",

Pavadinimas:"Pasiklydęs mėnulyje",

puslapiuSkaicius:130,

kiekis: 12,

kaina: 50.52

},

H77DF:{

Metai:"2018",

Pavadinimas:"Naujos galatikos užkariavimas",

puslapiuSkaicius:430,

kiekis: 12,

kaina: 33.52

},

H88DF:{

Metai:"2018",

Pavadinimas:"Žemės puolimas",

puslapiuSkaicius:330,

kiekis: 12,

kaina: 99.52

},

H99DF:{

Metai:"2018",

Pavadinimas:"Tai, kas po ledu",

puslapiuSkaicius:150,

kiekis: 12,

kaina: 45.52

}

},

Istorine:{

H111DF:{

Metai:"2018",

Pavadinimas:"Įdomioji istorija",

puslapiuSkaicius:140,

kiekis: 12,

kaina: 18.52

},

H222DF:{

Metai:"2018",

Pavadinimas:"Nendartalietis",

puslapiuSkaicius:230,

kiekis: 12,

kaina: 28.52

},

H333DF:{

Metai:"2018",

Pavadinimas:"Senovė",

puslapiuSkaicius:130,

kiekis: 12,

kaina: 44.52

},

H444DF:{

Metai:"2018",

Pavadinimas:"Nauji atradimai",

puslapiuSkaicius:830,

kiekis: 12,

kaina: 14.52

},

H555DF:{

Metai:"2018",

Pavadinimas:"Nauji faktai apie žalgirio mūsšį",

puslapiuSkaicius:390,

kiekis: 12,

kaina: 5.52

}

}

}

*18 užduotis*. NARŠYKLĖS KONSOLĖJE IŠVESKITE *17 užduotyje* SUKURTĄ KNYGŲ KATALOGĄ PAGAL ŽEMIAU PATEIKTĄ PAVYZDĮ.

var komentaras= " (Nauja Knyga)";

for(var prop\_1 in Katalogas){

console.log("Kategorija: "+prop\_1+" (Knygų skaičius: "+Object.keys(Katalogas[prop\_1]).length+"):");

for(var prop\_2 in Katalogas[prop\_1]){

console.log(" ISBN: "+prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

if (Katalogas[prop\_1][prop\_2].Metai==2019){

Katalogas[prop\_1][prop\_2].Metai += komentaras;

console.log(" "+prop\_3+": "+" "+Katalogas[prop\_1][prop\_2][prop\_3]);

}else{

console.log(" "+prop\_3+": "+" "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

*19 užduotis*. SUKURKITE KNYGŲ PIRKĖJO KORTELĖS OBJEKTĄ PAGAL STRUKTŪRĄ ŽEMIAU PATEIKTAME PAVYZDYJE.

var pirkejas = {};

pirkejas.kortelesID = "TR-CS-5689";

pirkejas.vardas = "Vardenis";

pirkejas.pavarde = "Pavardenis";

pirkejas.tipas = "VIP klientas";

pirkejas.paskutiniaiPirkimai = ['Idomioji istorija', 'Neregėta lietuva', 'Gyvenimas'];

*20 užduotis*. NARŠYKLĖS KONSOLĖJE IŠVESKITE *19 užduotyje* SUKURTO OBJEKTO SAVYBES (VARDAS, PAVARDĖ, PASKUTINIŲ ĮSIGYTŲ KNYGŲ KIEKĮ).

console.log("Vardas: "+pirkejas.vardas);

console.log("Pavarde: "+pirkejas.pavarde);

console.log("Įsigytų knygų skaičius: "+pirkejas.paskutiniaiPirkimai.length);

*21 užduotis. 19 užduotyje* SUKURTO OBJEKTO DUOMENIS PAPILDYKITE. PRIDĖKITE DAR VIENĄ ĮSIGYTĄ KNYGĄ NEREDAGUOJANT PAČIO OBJEKTO (SET METODAS).

pirkejas.paskutiniaiPirkimai.push('JavaScript');

*22 užduotis*. PAGAL *19 užduotyje* PATEIKTĄ OBJEKTO STRUKTŪRĄ SUKURKITE KONSTRUKTORIŲ OBJEKTŲ KŪRIMUI.

function Pirkejas(id, vardas, pavarde, tipas, pirkimai)

{

this.id = id;

this.vardas = vardas;

this.pavarde = pavarde;

this.tipas = tipas;

this.pirkimai = pirkimai;

}

*23 užduotis*. PANAUDODAMI *22 užduotyje* SUKURTĄ KONSTRUKTORIŲ SUKURKITE 3 KNYGŲ PIRKĖJUS.

var pirkejas1 = new Pirkejas('TR-CS-5689', 'Vardenis', 'Pavardenis', 'VIP klientas',['Įdomioji istorija', 'Nerėgėta lietuva', 'Gyvenimas']);

var pirkejas2 = new Pirkejas('TR-CS-5677', 'Petras', 'Petraitis', 'paprastas klientas',['PHP 7', 'SASS', 'AI']);

var pirkejas3 = new Pirkejas('TR-CS-5678', 'Jonas', 'Jonaitis', 'VIP klientas',['Konstitucija', 'Civilinis kodeksas', 'Baudžiamasis kodeksas']);

console.log(pirkejas1);

console.log(pirkejas2);

console.log(pirkejas3);

*24* *užduotis*. SUKURKITE PIRKĖJO OBJEKTO PROTOTIPĄ SU METODU SPAUSDINTI.

function Pirkejas(id, vardas, pavarde, tipas, pirkimai)

{

this.id = id;

this.vardas = vardas;

this.pavarde = pavarde;

this.tipas = tipas;

this.pirkimai = pirkimai;

}

Pirkejas.prototype.spausdinti = function(){

console.log("Kortelės ID: "+this.id);

console.log("Vardas: "+this.vardas);

console.log("Pavardė: "+this.pavarde);

console.log("Kliento tipas: "+this.tipas);

console.log("Paskutinės įsigytos knygos: ")

for(var i=0; i<this.pirkimai.length; i++)

{

console.log(this.pirkimai[i]);

}

}

var pirkejas1 = new Pirkejas('TR-CS-5689', 'Vardenis', 'Pavardenis', 'VIP klientas',['Įdomioji istorija', 'Nerėgėta lietuva', 'Gyvenimas']);

var pirkejas2 = new Pirkejas('TR-CS-5677', 'Petras', 'Petraitis', 'paprastas klientas',['PHP 7', 'SASS', 'AI']);

var pirkejas3 = new Pirkejas('TR-CS-5678', 'Jonas', 'Jonaitis', 'VIP klientas',['Konstitucija', 'Civilinis kodeksas', 'Baudžiamasis kodeksas']);

pirkejas1.spausdinti();

pirkejas2.spausdinti();

pirkejas3.spausdinti();

*25 užduotis*. IŠPLĖSKITE 22 *užduotyje* SUKURTĄ PIRKĖJO KONSTRUKTORIŲ SUKURDAMI NAUJĄ PAVADINIMU KLUBO NARYS.

function KluboNarys(id, vardas, pavarde, tipas, pirkimai, lojalumoTaskai, reitingas)

{

Pirkejas.call(this, id, vardas, pavarde, tipas, pirkimai)

this.lojalumoTaskai = lojalumoTaskai;

this.reitingas = reitingas;

}

*26 užduotis*. SUKURKITE 3 KNYGŲ PIRKĖJUS PAGAL IŠPLĖSTA KONSTRUKTORIŲ *25* *užduotyje.*

var pirkejas1 = new KluboNarys('TR-CS-5689', 'Vardenis', 'Pavardenis', 'VIP klientas',['Įdomioji istorija', 'Nerėgėta lietuva', 'Gyvenimas'],200,7);

var pirkejas2 = new KluboNarys('TR-CS-5677', 'Petras', 'Petraitis', 'paprastas klientas',['PHP 7', 'SASS', 'AI']);

var pirkejas3 = new KluboNarys('TR-CS-5678', 'Jonas', 'Jonaitis', 'VIP klientas',['Konstitucija', 'Civilinis kodeksas', 'Baudžiamasis kodeksas'],300,9);

localStorage.setItem('pirkejas1', JSON.stringify(pirkejas1));

localStorage.setItem('pirkejas2', JSON.stringify(pirkejas2));

localStorage.setItem('pirkejas3', JSON.stringify(pirkejas3));

loadPirkejas1 = localStorage.getItem('pirkejas1');

loadPirkejas2 = localStorage.getItem('pirkejas2');

loadPirkejas3 = localStorage.getItem('pirkejas3');

console.log(JSON.parse(loadPirkejas1));

console.log(JSON.parse(loadPirkejas2));

console.log(JSON.parse(loadPirkejas3));

*27 užduotis*. *19* *užduotyje* SUKURTAM PIRKĖJO KORTELĖS OBJEKTUI ATLIKITE PAKEITIMUS UŽDRAUDŽIANT Į ĮSIGYTŲ KNYGŲ MASYVE PRIDĖTI NAUJĄ KNYGĄ.

function KluboNarys(id, vardas, pavarde, tipas, pirkimai, lojalumoTaskai, reitingas)

{

Pirkejas.call(this, id, vardas, pavarde, tipas, pirkimai)

this.lojalumoTaskai = lojalumoTaskai;

this.reitingas = reitingas;

Object.freeze(this.pirkimai);

Object.isFrozen(this.pirkimai);

}

*28 užduotis*. NEKEISDAMI ITERATORIAUS (KINTAMOJO) PAVADINIMO CIKLUOSE PATAISYKITE KODĄ, KAD BŪTŲ IŠVESTOS VISOS KATEGORIJOS IR JOMS PRIKLAUSANČIOS KNYGOS.

var knygos = ['Fantastika', 'IT', 'Technologijos'];

knygos['Fantastika']= ['Misijos į marsą istorija ', 'Ateiviai', 'Vanduo menulyje'];

knygos['IT']= ['Programuok linskmai', 'Naujos WEB technologijos'];

knygos['Technologijos'] = ['Elektromobiliai', 'Nauji kosminiai laiavai'];

for(let i=0; i<knygos.length;i++)

{

var kategorija = knygos[i];

console.log(kategorija);

for(let i=0; i<knygos[kategorija].length; i++)

{

console.log(knygos[kategorija][i])

}

}

*29 užduotis*. PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD BŪTŲ IŠVESTAS TEISINGAS REZULTATAS.

const a = [];

for(let i =0; i<10; i++)

{

a[i] = ()=> {console.log(i)};

}

a[3]()

a[7]()

*30 užduotis*. PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS

let knygos = ['Fantastika', 'IT', 'Technologijos'];

knygos['Fantastika']= ['Misijos į marsą istorija ', 'Ateiviai', 'Vanduo menulyje'];

knygos['IT']= ['Programuok linskmai', 'Naujos WEB technologijos'];

knygos['Technologijos'] = ['Elektromobiliai', 'Nauji kosminiai laiavai'];

knygos = ['Romanai','Grožinė literatūra','Tikslieji mokslai'];

*31 užduotis. 17 užduotyje* SUKURTO KNYGŲ KATALOGO IŠVEDIMUI SUKURKITE AROW FUNKCIJĄ, KURI PRIE KNYGOS KAINOS PRIDEDA PVM MOKESTĮ (21%) IR NARŠYKLĖS KONSOLĖJE IŠVEDA KAINĄ SU PVM.

var komentaras= " (Nauja Knyga)";

pvmKaina = (kaina) => kaina \*1.21;

for(var prop\_1 in Katalogas){

console.log(prop\_1+" (Knygos: "+Object.keys(Katalogas[prop\_1]).length+")");

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

Katalogas[prop\_1][prop\_2].kaina = pvmKaina(Katalogas[prop\_1][prop\_2].kaina);

if (Katalogas[prop\_1][prop\_2].Metai>=2018){

Katalogas[prop\_1][prop\_2].Metai += komentaras;

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}else{

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

*32 užduotis. 17 užduotyje* SUKURTĄ KNYGŲ KATALOGĄ SUKURKITE KAIP MODULĮ VIENAME FAILE, O IŠVEDIMĄ REAZLIZUOKITE KITAME FAILE JĮ IMPORTUODAMI.

**knygos.js**

export {Katalogas};

**app.js**

import \* as Katalogas from 'knygos.js';

*33 užduotis.* ŽEMIAU PATEIKTAM SKAITYTOJŲ MASYVUI PANAUDOKITE DESTRUKTORIŲ. IŠVESKITE SKAITYTOJŲ VARDĄ IR KUO DOMISI.

var skaitytojai = [

{

vardas: 'Vardenis Pavardenis',

papildomaInformacija: {

domisi: 'Aviacija',

mokosi: 'Kauno informaicniu technologiju mokykloje',

gyvena: 'Kaune'

},

amzius: 35

},

{

vardas: 'Jonas Jonaitis',

papildomaInformacija: {

domisi: 'Programavimu',

mokosi: 'Kauno informaicniu technologiju mokykloje',

gyvena: 'Kaune'

},

amzius: 25

}

];

for (var {vardas: v, papildomaInformacija: {domisi: f}} of skaitytojai) {

console.log('Vardas: ' + v + ', Domisi: ' + f);

}

**3.4.5. Tema. ES6 standarto ypatumai.**

*34 užduotis.* IŠVESKITE INFORMACIJĄ PAGAL DETALESNĮ APRAŠYMĄ APIE KNYGAS (*17 užduotyje aprašytas katalogas*) PAGAL ŽEMIAU NURODYTĄ PAVYZDĮ, NAUDODAMI ES6 TEMPLATE LITERALS.

for(var prop\_1 in Katalogas){

console.log(prop\_1+" (Knygos: "+Object.keys(Katalogas[prop\_1]).length+")");

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

let template = `Labai gera knyga ${Katalogas[prop\_1][prop\_2].Pavadinimas}, kurią paraštė nuotabus autorius. \n\ Džiaugiamės savo klientams galėdami pasiūlyti net ${Katalogas[prop\_1][prop\_2].kiekis} knygų. \n \ Norėdami įsigyti šią knygą mūsų knygynuose, ieškokite skyriaus pavadinimu ${prop\_1}.

\n \ Iki susitikimo mūsų knygynuose`

console.log(template);

}

}

}

**3.5. Mokymosi rezultatas. Vykdyti programinio kodo versijavimą naudojant versijavimo įrankius.**

**3.5.1. Tema. Išeities kodo saugyklos.**

*1 užduotis*. UŽRAŠYKITE TRIS KODO VERSIJŲ KONTROLĖS SISTEMAS, APRAŠYKITE PAGRINDINES VERSIJAVIMO SISTEMŲ FUNKCIJAS.

Dažniausiai naudojamos kodo versijų kontrolės sistemos: Git, Mercurial, SVN, Azure DevOps, BitKeeper ir pan.. Pagrindinės kodo versijavimo sistemų funkcijos:

1. versijuoja kiekvieną kodo pakeitimą;

2. palaiko keletą kodo versijų;

3. gali atstatyti kodą į bet kurią prieš tai buvusią būseną;

4. leidžia dirbti komandai vienu metu prie to paties kodo.

**Pavyzdys:** jus esate Web tinklapių programuotojas ir kuriate tinklalapius pagal dizainerio pateiktus šablonus. Kiekvieną savo parašytą kodo versiją jus galite išsaugoti ir grįžti prie jos bet kokiu metu. Galite turėti kelias to pačio kodo versijas. Galite dirbti komandoje ir dalintis savo patirtimi su kitais programuotojais.

*2 užduotis.* PARSISIŲSKITE, ĮSIDIEKITE GIT KODO VERSIJAVIMO ĮRANKĮ.

Git versijavimo įrankį galite atsiųsti iš čia:

Git (2019). *Downloads*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/downloads.>

Diegimo instrukcija:

Git (2019). *Getting Started - Installing Git*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.>

*3 užduotis*. SUKONFIGŪRUOKITE GIT SAUGYKLĄ: NUSTATYKITE GIT VARTOTOJO VARDĄ, ELEKTRONINĮ PAŠTĄ, NAUDOJAMĄ TEKSTINĮ REDAKTORIŲ.

Git saugyklos konfigūravimas:

Vartotojo vardo nustatymas: *git config --global user.name "John Doe"*

Vartotojo elektroninio pašto nustatymas: *git config --global user.email johndoe@example.com*

Vartotojo tekstinio redaktoriaus nustatymas: *git config --global core.editor emacs*

Papildomai apie Git konfigūravimą galite paskaityti čia:

Git (2019). *Getting Started - First-Time Git Setup*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup.>

*4 užduotis*. PAPILDYKITE SAVO GIT KONFIGŪRACIJĄ (žiūrėti 3 užduotį) PARAMETRAIS: COMMIT.TEMPLATE, CORE.PAGER, CORE.EXCLUDEFILE, HELP.AUTOCORRECT, COLOR.\*

Konfigūruojant papildomus parametrus naudokite šią instrukciją:

Git (2019). *Customizing Git - Git Configuration*. [žiūrėta 2019-05-27].

Prieiga per internetą <https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration.>

*5 užduotis*. APRAŠYKITE BAZINES GIT KOMANDAS: GIT INIT, GIT CLONE, GIT STATUS.

Aprašymui galite naudoti šiuos šaltinius:

1. git init: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-init>

2. git clone: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-clone>

3. git status: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-status>

Papildomai galite paskaityti:

Git (2019). *Git Basics - Getting a Git Repository*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository.>

*6 užduotis.* APRAŠYKITE BAZINES GIT KOMANDAS: GIT ADD, GIT COMMIT, GIT RESET, GIT CHECKOUT.

Aprašymui galite naudoti šiuos šaltinius:

1. git add: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-add>

2. git commit: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-commit>

3. git reset: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-reset>

4. git checkout: [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-checkout>

Papildomai galite paskaityti:

Git (2019). *Git Basics - Getting a Git Repository*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository.>

*7 užduotis.* PARAŠYKITE GIT KOMANDAS, KURIOS LEIS SUKURTI NAUJAS PROJEKTO ŠAKAS: TESTING, NEW-FEATURE.

Naujų Git šakų sukurimui naudojame komandas:

1. git branch testing arba git checkout -b testing

2. git branch new-feature arba git checkout new-feature

*8 užduotis.* PARAŠYKITE GIT KOMANDAS, KURIOS LEIS APJUNGTI *7 užduotyje* SUKURTAS ŠAKAS SU PAGRINDINE SAUGYKLOS ŠAKA MASTER.

Git šakų apjungimui naudojame komandas:

1. git merge master testing

2. git merge master new-feature

Git šakų trynimui naudojame komandą:

1. git branch -d testing

2. git branch -d new-feature

Papildomai galite paskaityti:

1. Git (2019). *Git Branching - Branches in a Nutshell*. [žiūrėta 2019-05-27].

Prieiga per internetą <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell.>

**3.5.2. Tema. Programinio kodo versijavimo vykdymas.**

*9 užduotis.* NUKOPIJUOKITE KODO PAVYZDĮ IŠ NUOTOLINĖS GITHUB KODO SAUGYKLOS*:* <https://github.com/fbeline/design-patterns-JS>.

Kodo kopijavimas iš nuotolinės saugyklos:

1. git clone https://github.com/fbeline/design-patterns-JS.

Redaguojame failą README.md (projekto aprašymas). Įrašome į šį failą eilutė: “This is my forked project.”. Pakeitimus įrašome į kodo saugyklą:

2. git add README.

3. git commit -m “Changed README.md file, change project description.”

Savo nuožiūra padarykite dar kelis pakeitimus projekto failuose ir juos įrašykite į kodo saugyklą.

*10 užduotis.* SUKURKITE LOKALIĄ GIT KODO SAUGYKLĄ APLANKE VARDU CALCULATOR.

Sukuriame aplanką vardu Calculator. Jame inicializuojame naują Git kodo saugyklą su Git komanda: git init. Toliau sukuriame projektą su atitinkama klase Calculator.JavaScript. Parašome skaičiuotuvo aritmetines operacijas. Kiekvieną parašytą aritmetinį veiksmą įrašome į kodo saugyklą.

Parašius šį kodą įvykdome Git komandas:

1. git add .

2. git commit -m “Add adder function in Calculator.”

3. git status

4 git log

*11 užduotis.* SAVO SKAIČIUOTUVUI *(10 užduotis)* PADARYKITE KODO PATAISYMUS IR JUOS EKSPORTUOKITE SU GIT KOMANDĄ GIT FORMAT-PATCH.

Pataisymų eksportui naudokite komandą: git format-patch <commit number>

Papildomai galite paskaityti:

Git (2019). *Git format patch*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-format-patch.>

*12 užduotis*. GITHUB KODO TALPINIMO PLATFORMOJE YRA PATALPINTAS REACT KODAS.

Jo repozitorija yra čia: <https://github.com/facebook/react>. Naudojant Git įrankį išanalizuokite kiek kodo eilučių buvo parašyta šiame projekte? Kiek programuotojų dirba prie pateikto projekto?

Eilučių skaičius gali būti suskaičiuotas naudojant šią komandą:

git ls-files | xargs cat | wc -l

Programuotojų skaičius:

git shortlog -s -n | wc –l

**3.6. Mokymosi rezultatas. Naudoti JavaScript funkcijas.**

**3.6.1. Tema. Funkcijos.**

*1 užduotis.* PATOBULINKITE KOMPETENCIJOS 3.4. *17 užduotyje* SUKURTO KNYGŲ KATALOGO IŠVEDIMĄ SUKURDAMI FUKCIJĄ, KURI SUSKAIČIUOJA KATEGORIJOJE ESANČIŲ KNYGŲ SKAIČIŲ.

var komentaras= " (Nauja Knyga)";

function kiekisKategorijoje(arg){

return Object.keys(Katalogas[arg]).length

}

for(var prop\_1 in Katalogas){

console.log(prop\_1+" (Knygos: "+kiekisKategorijoje(prop\_1)+")");

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

if (Katalogas[prop\_1][prop\_2].Metai>=2018){

Katalogas[prop\_1][prop\_2].Metai += komentaras;

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}else{

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

*2 užduotis*. SUKURKITE FUNKCIJĄ, KURIAI NURODŽIUS DUOMENŲ ŠALTINĮ IR KATEGORIJŲ KIEKĮ NARŠYKLĖS KONSOLĖJE BŪTŲ IŠVESTAS TIK NURODYTŲ KATEGORIJŲ KIEKIS IR JOMS PRIKLAUSANČIŲ KNYGŲ SĄRAŠĄ.

var counter = 0;

function kiekisKataloge(data, limit){

var Katalogas = data ;

for(var prop\_1 in Katalogas){

if(counter == limit)

{

break;

}

counter++;

console.log(prop\_1);

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

kiekisKataloge(Katalogas,2);

*3 užduotis*. PERRAŠYKITE KOMPETENCIJOS 3.4. *17 užduotyje* SUKURTĄ FUNKCIJĄ Į ARROW FUNKCIJĄ.

var counter = 0;

var kiekisKataloge =(data, limit) => {

var Katalogas = data ;

for(var prop\_1 in Katalogas){

if(counter == limit)

{

break;

}

counter++;

console.log(prop\_1);

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

*4 užduotis*. PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS (NEKEIČIANT PASKUTINĖS KODO EILUTĖS).

function pasisveikinti(){

const pirkejas = 'Vardenis Pavardenis';

return pirkejas

}

pirkejas = pasisveikinti();

**3.6.2. Tema. Standartinės JavaScript funkcijos.**

*5 užduotis*. PATAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD IŠKVIETUS OBJEKTO SPAUSDINK METODĄ RODYTIX BŪTŲ IŠVESTA x REIKŠMĖ (OBJEKTO KOREGUOTI NEGALIMA).

var spausdink = {

x: 54,

rodytiX: function() {

return this.x;

}

}

var xReiksme = spausdink.rodytiX

xReiksme = xReiksme.bind(spausdink);

console.log(xReiksme());

*6 užduotis*. SUKURKITE ANONIMINĘ FUNKCIJĄ, KURI KONVERTUOJA GAUTĄ MASYVĄ Į EILUTĘ (STRING).

let duomenys = function(data){

return JSON.stringify(data);

}

console.log(duomenys(Katalogas));

*7 užduotis*. SUKURKITE FUNKCIJĄ, KURI IŠVEDA KOMPETENCIJOS 3.4. *17 užduotyje* SUKURTĄ KNYGŲ KATALOGĄ.

(function () {

var komentaras= " (Nauja Knyga)";

for(var prop\_1 in Katalogas){

console.log(prop\_1);

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

if (Katalogas[prop\_1][prop\_2].Metai>=2018){

Katalogas[prop\_1][prop\_2].Metai += komentaras;

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}else{

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

})();

*8 užduotis*. ŽEMIAU PATEIKTAME MASYVE PATEIKTI KNYGŲ PARDAVIMAI PAGAL METUS. PANAUDODAMI MINIMALIOS PAIEŠKOS FUNKCIJĄ, SURASKITE MAŽIAUSIĄ PARDAVIMŲ SUMĄ 2019 M.

const pardavimai = ['2017', '2018', '2019','2020'];

pardavimai['2017'] = [1500,12005,2893, 35000,17000,1223898.12,56784,2455,124588,56597,125658,124788];

pardavimai['2018'] = [200,152005,2563, 35600,15000,125698.12,56856,23455,124788,5647,12548,125688];

pardavimai['2019'] = [5600,18005,2233, 45000,15000,127898.12,56544,23475,1255698,5497,12478,122588];

pardavimai['2020'] = [5600,19005,2233, 65000,14230,127898.12,5647,2347,12478,5627,123258,12474588];

console.log(Math.min(...pardavimai['2019']));

*9 užduotis*. PAGAL *8 užduotyje* APRAŠYTĄ PARDAVIMŲ MASYVĄ, PANAUDODAMI MAKSIMALIOS PAIEŠKOS FUNKCIJĄ, SURASKITE 2018 M. DIDŽIAUSIĄ PARDAVIMŲ SUMĄ.

console.log(Math.max(...pardavimai['2019']));

*10 užduotis*. PAGAL *8 užduotyje* APRAŠYTĄ PARDAVIMŲ MASYVĄ, PANAUDODAMI APVALINIMO FUNKCIJĄ, SUAPVALINKITE 2018 M. PARDAVIMŲ SUMAS.

for(i=0; i<pardavimai['2018'].length; i++)

{

console.log(Math.round(pardavimai['2018'][i]));

}

*11 užduotis*. PARAŠYKITE PROGRAMĄ, KURI PAŠALINTŲ SKAIČIUS PO KABLELIO IŠ 3.4. KOMPETENCIJOS *17 užduotyje* APRAŠYTO KATALOGO KNYGŲ KAINŲ (NEAPVALINANT) PANAUDOJUS MATEMATINĘ FUNKCIJĄ. KNYGAS IŠVESKITE NARŠYKLĖS KONSOLĖJE.

var komentaras= " (Nauja Knyga)";

for(var prop\_1 in Katalogas){

console.log(prop\_1);

for(var prop\_2 in Katalogas[prop\_1]){

console.log(prop\_2);

for(var prop\_3 in Katalogas[prop\_1][prop\_2]){

Katalogas[prop\_1][prop\_2].kaina = Math.floor(Katalogas[prop\_1][prop\_2].kaina);

if (Katalogas[prop\_1][prop\_2].Metai>=2018){

Katalogas[prop\_1][prop\_2].Metai += komentaras;

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}else{

console.log(prop\_3+": "+Katalogas[prop\_1][prop\_2][prop\_3]);

}

}

}

}

*12 užduotis*. PRAPLĖSKITE MATH OBJEKTĄ SUKURDAMI SAVO METODĄ.

Math.daliklis = function() {

if (argumentai.length == 2) {

if (argumentai[1] == 0)

return argumentai[0];

else

return Math.daliklis(argumentai[1], argumentai[0] % argumentai[1]);

} else if (argumentai.length > 2) {

var rezultatas = Math.daliklis(argumentai[0], argumentai[1]);

for (var i = 2; i < argumentai.length; i++)

rezultatas = Math.daliklis(rezultatas, argumentai[i]);

return rezultatas;

}

};

*13 užduotis*. IŠVESKITE 3.4. KOMPETENCIJOS *17 užduotyje* SUKURTO KNYGŲ KATALOGO OBJEKTŲ SAVYBES PANAUDODAMI OBJEKTŲ FUNKCIJĄ, KURI GRĄŽINA OBJEKTO SAVYBES. GRĄŽINTOS SAVYBĖS TURI BŪTI PATALPINTOS MASYVE IR IŠVESTOS NARŠYKLĖS KONSOLĖJE.

console.log(Object.getOwnPropertyNames(Katalogas));

*14 užduotis*. SUKURKITE NAUJĄ – TUŠČIĄ OBJEKTĄ PAVADINIMU NAUJAS KATALOGAS. NUKOPIJUOKITE 3.4. KOMPETENCIJOS *17 užduotyje* ESANČIŲ OBJEKTŲ SAVYBES SU REIKŠMĖMIS IR PRISKIRKIT NAUJAI SUKURTAM OBJEKTUI.

const naujasKatalogas = {};

const rodytiKataloga = Object.assign(naujasKatalogas, Katalogas);

console.log(rodytiKataloga);

**3.6.3. Tema. Pagrindinės masyvų funkcijos.**

*15 užduotis*. PANAUDODAMI MASYVŲ FUNKCIJĄ FIND() IŠ 8 užduotyje SUKURTO PARDAVIMŲ MASYVO, SURASKITE 2017M. PARDAVIMO SUMĄ, KURI DIDESNĖ UŽ 15000.

const naujasKatalogas = {};

var surasta = pardavimai['2017'].find(function(element) {

return element > 15000;

});

*16 užduotis.* IŠ ŽEMIAU PATEIKTO MASYVO PANAUDODAMI MASYVŲ FUNKCIJĄ FILTER, SURASKITE KNYGAS, KURIŲ PAVADINIMO ILGIS YRA DIDESNIS UŽ 15 SIMBOLIŲ.

const rezultatas = knyguPavadinimai.filter(word => word.length > 15);

console.log(rezultatas);

*17 užduotis*. PANAUDODAMI MASYVŲ FUNKCIJĄ SORT() SURIKIUOKITE KNYGAS IŠ *18 užduotyje* SUKURTO KNYGŲ KATALOGO PAGAL PAVADINIMĄ ABĖCELĖS TVARKA.

const rezultatas = knyguPavadinimai.sort();

console.log(rezultatas);

*18 užduotis*. PANAUDODAMI MASYVŲ FUNKCIJĄ REDUCE() APSKAIČIUOKITE *8 užduotyje* SUKURTO MASYVO 2019 M. APRAŠYTŲ PARDAVIMŲ BENDRĄ SUMĄ. REZULTATĄ IŠVESKITE NARŠYKLĖS KONSOLĖJE.

const reducer = (accumulator, currentValue) => accumulator + currentValue;

console.log(pardavimai['2019'].reduce(reducer));

*19 užduotis*. PANAUDODAMI MASYVŲ FUNKCIJĄ MAP() APSKAIČIUOKITE *8 užduotyje* SUKURTO MASYVO 2019 M. SUMAS. PADAUGINKITE IŠ 2.

const padididntosKainos = pardavimai['2019'].map(x => x \* 2);

console.log(padididntosKainos)

**3.7. Mokymosi rezultatas. Taikyti objektinio programavimo principus programuojant.**

**3.7.1. Tema. Objektinis programavimas.**

*1 užduotis*. SUKURKITE KLASĘ KNYGA.

class Knyga{

constructor(isbn, autorius, leidimoMetai, pavadinimas, aparasymas, puslapiuSkaicius){

this.\_isbn = isbn;

this.\_autorius = autorius;

this.\_leidimoMetai = leidimoMetai;

this.\_pavadinimas = pavadinimas;

this.\_aprasymas = aparasymas;

this.\_puslapiuSkaicius = puslapiuSkaicius;

}

spausdinti() {

console.log("ISBN :" + this.isbn);

console.log("Autorius :" + this.isbn);

console.log("Leidimo metai :" + this.leidimoMetai);

console.log("Pavadinimas :" + this.pavadinimas);

console.log("Aprašymas :" + this.aprasymas);

console.log("Puslapių skaičius :" + this.puslapiuSkaicius);

}

}

*2 užduotis.1 užduotyje* SUKURTOS KLASĖS PAGRINDU SUKURKITE 3 KNYGŲ OBJEKTUS.

let knyga1 = new Knyga('TR-89745', 'Jonas Jonaitis', '2018 m.', 'Lietuva', 'Labai gera knyga', 230);

knyga1.spausdinti();

let knyga2 = new Knyga('TR-87745', 'Vardenis Pavardenis', '2019 m.', 'Latvija', 'Nuostabi knyga', 250);

knyga2.spausdinti();

let knyga3 = new Knyga('TR-669745', 'Petras Petraitis', '2017 m.', 'Estija', 'Puiki knyga', 220);

knyga3.spausdinti();

**3.7.2. Tema. Klasių instancijavimas ir metodai.**

*3 užduotis*. PRAPLĖSKITE *1 užduotyje* SUKURTĄ KLASĘ SUKURDAMI NAUJĄ KLASĘ PAVADINIMU ELKNYGA.

class ElKnyga extends Knyga{

constructor(isbn, autorius, leidimoMetai, pavadinimas, aparasymas, puslapiuSkaicius, formatas){

super(isbn, autorius, leidimoMetai, pavadinimas, aparasymas, puslapiuSkaicius);

this.\_formatas = formatas;

}

toArray()

{

let data = [];

data.push(this.\_isbn);

data.push(this.\_autorius);

data.push(this.\_leidimoMetai);

data.push(this.\_pavadinimas);

data.push(this.\_aprasymas);

data.push(this.\_puslapiuSkaicius);

data.push(this.\_formatas);

return data;

}

}

let knyga1 = new ElKnyga('TR-89745', 'Jonas Jonaitis', '2018 m.', 'Lietuva', 'Labai gera knyga', 230, 'pdf');

console.log(knyga1.toArray());

let knyga2 = new ElKnyga('TR-87745', 'Vardenis Pavardenis', '2019 m.', 'Latvija', 'Nuostabi knyga', 250, 'jpg');

console.log(knyga2.toArray());

let knyga3 = new ElKnyga('TR-669745', 'Petras Petraitis', '2017 m.', 'Estija', 'Puiki knyga', 220, 'epub');

*4 užduotis. 3 užduotyje* SUKURTĄ KLASĘ PRAPLĖSKITE STATINIU METODU PAVADINTU PALYGINIMAS.

class ElKnyga extends Knyga{

constructor(isbn, autorius, leidimoMetai, pavadinimas, aparasymas, puslapiuSkaicius, formatas){

super(isbn, autorius, leidimoMetai, pavadinimas, aparasymas, puslapiuSkaicius);

this.\_formatas = formatas;

}

toArray()

{

let data = [];

data.push(this.\_isbn);

data.push(this.\_autorius);

data.push(this.\_leidimoMetai);

data.push(this.\_pavadinimas);

data.push(this.\_aprasymas);

data.push(this.\_puslapiuSkaicius);

data.push(this.\_formatas);

return data;

}

static palyginti(arg1, arg2) {

console.log(Object.values(arg1));

console.log(Object.values(arg2));

}

}

let knyga1 = new ElKnyga('TR-89745', 'Jonas Jonaitis', '2018 m.', 'Lietuva', 'Labai gera knyga', 230, 'pdf');

let knyga2 = new ElKnyga('TR-87745', 'Vardenis Pavardenis', '2019 m.', 'Latvija', 'Nuostabi knyga', 250, 'jpg');

let knyga3 = new ElKnyga('TR-669745', 'Petras Petraitis', '2017 m.', 'Estija', 'Puiki knyga', 220, 'epub');

ElKnyga.palyginti(knyga1, knyga2);

* 1. **Mokymosi rezultatas. Keisti ir manipuliuoti HTML elementų medį (DOM).**

**3.8.1. Tema. Darbas su JavaScript Document Object Model (DOM).**

*1 užduotis*. SUKURKITE PROGRAMĄ, KURI SUGENERUOJA HTML LENTELĘ.

**HTML**

<section id="lentele">

<h3>Lentelės kūrimas</h3>

<div class="message"></div>

<input type="text" name="eilute" placeholder="Eilutės">

<input type="text" name="stulpelis" placeholder="Stulpeliai">

<button id="kurkLentele">Kurti lentelę</button>

<div id="table">

</div>

</section>

**JS**

const mygtukasKurkLentele = document.querySelector('#kurkLentele');

mygtukasKurkLentele.addEventListener('click', kurk);

function kurk(){

var eilutes = document.querySelector("input[name='eilute']").value;

var stulpeliai = document.querySelector("input[name='stulpelis']").value;

var klaida = document.querySelector(".message");

if(eilutes && stulpeliai) {

klaida.innerHTML="";

var table = document.createElement('table'), tr, td, eilute, langelis;

for (eilute = 0; eilute < eilutes; eilute++) {

tr = document.createElement('tr');

for (langelis = 0; langelis < stulpeliai; langelis++) {

td = document.createElement('td');

tr.appendChild(td);

td.innerHTML = "...";

}

table.appendChild(tr);

}

document.getElementById('table').appendChild(table);

} else{

klaida.innerHTML='<span class="klaida">Neįvestas eilučių ir stulpelių skaičius</span>';

}

}

*2 užduotis*. SUKURKITE PROGRAMĄ, KURI HTML BODY ELEMENTUI PER NUSTATYTĄ LAIKĄ PAKEIČIA ATSITIKTINĘ FONO SPALVĄ.

<body>

<script>

var btn = document.querySelector('button');

var body = document.querySelector('body');

random = function(color){

return Math.floor(color \* Math.random());

}

setInterval(function() {

var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';

body.style.backgroundColor = rndCol;

body.style.transform = 'scale(1)';

}, 2000);

</script>

*3 užduotis*. SUKURKITE PAVEIKSLĖLIŲ ATVAIZDAVIMO IR MAIŠYMO PROGRAMĄ.

**HTML**

<div class="container">

<div class="row">

<div class="col-sm-12 galerija">

<button class="rodyti">Rodyti</button>

<button class="maisyti">Maišyti</button>

<ul>

</ul>

</div>

</div>

</div>

**JS**

var mygtukasRodyti = document.querySelector('button.rodyti');

var mygtukasMaisyti = document.querySelector('button.maisyti');

var nuotraukos =['1.png', '2.jpg', '3.png', '4.png', '5.jpg', '6.jpg', '7.jpg', '8.jpg', '9.jpg', '10.png'];

var nuotraukosUl = document.querySelector('.galerija ul');

var imgItem = document.querySelector('.galerija img');

mygtukasRodyti.addEventListener('click',parodyk);

mygtukasMaisyti.addEventListener('click',maisyti);

mygtukasMaisyti.addEventListener('click',maisyti);

imgItem.addEventListener('dblclick', salinti);

function parodyk (){

for(i=0; i<nuotraukos.length;i++){

var li =document.createElement('li');

li.className = "col-sm-3";

nuotraukosUl.appendChild(li);

li.innerHTML='<img src="img/galerija/'+nuotraukos[i]+'">';

}

}

function maisyti(){

nuotraukos.sort(function(a, b){return 0.5 - Math.random()});

var imgElementai = document.querySelectorAll('.galerija img');

for(i=0; i<nuotraukos.length; i++){

imgElementai[i].src='img/galerija/'+nuotraukos[i];

}

}

*4 užduotis****.*** SUKURKITE TESTĄ, TIKRINANTĮ IR PATEIKIANTĮ TEISINGUS ATSAKYMUS.

Rezultatai:

1. Index.html failas:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>JS testas</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h2>Bootstrap Quiz</h2>

<div id="quiz"></div>

<button id="submit">Submit Quiz</button>

<div id="results"></div>

<script src="script.js"></script>

</body>

</html>

2. Script.js failas:

function() {

function buildQuiz() {

// we'll need a place to store the HTML output

const output = [];

// for each question...

myQuestions.forEach((currentQuestion, questionNumber) => {

// we'll want to store the list of answer choices

const answers = [];

// and for each available answer...

for (letter in currentQuestion.answers) {

// ...add an HTML radio button

answers.push(

`<label>

<input type="radio" name="question${questionNumber}" value="${letter}">

${letter} :

${currentQuestion.answers[letter]}

</label>`

);

}

// add this question and its answers to the output

output.push(

`<div class="question"> ${currentQuestion.question} </div>

<div class="answers"> ${answers.join("")} </div>`

);

});

// finally combine our output list into one string of HTML and put it on the page

quizContainer.innerHTML = output.join("");

}

function showResults() {

// gather answer containers from our quiz

const answerContainers = quizContainer.querySelectorAll(".answers");

// keep track of user's answers

let numCorrect = 0;

// for each question...

myQuestions.forEach((currentQuestion, questionNumber) => {

// find selected answer

const answerContainer = answerContainers[questionNumber];

const selector = `input[name=question${questionNumber}]:checked`;

const userAnswer = (answerContainer.querySelector(selector) || {}).value;

// if answer is correct

if (userAnswer === currentQuestion.correctAnswer) {

// add to the number of correct answers

numCorrect++;

// color the answers green

answerContainers[questionNumber].style.color = "lightgreen";

} else {

// if answer is wrong or blank

// color the answers red

answerContainers[questionNumber].style.color = "red";

}

});

// show number of correct answers out of total

resultsContainer.innerHTML = `${numCorrect} out of ${myQuestions.length}`;

}

const quizContainer = document.getElementById("quiz");

const resultsContainer = document.getElementById("results");

const submitButton = document.getElementById("submit");

const myQuestions = [

{

question: "The Bootstrap grid system is based on how many columns?",

answers: {

a: "6",

b: "8",

c: "12"

},

correctAnswer: "c"

},

{

question: "Which class is used to create a big box for calling extra attention?",

answers: {

a: ".bigbox",

b: ".jumbo",

c: ".jumbotron"

},

correctAnswer: "c"

},

{

question: "Bootstrap is mobile-first",

answers: {

a: "true",

b: "false",

},

correctAnswer: "a"

}

];

// display quiz right away

buildQuiz();

// on submit, show results

submitButton.addEventListener("click", showResults);

})();

3. Style.css failas:

body{

font-size: 20px;

font-family: sans-serif;

color: #333;

}

.question{

font-weight: 600;

}

.answers {

margin-bottom: 20px;

}

.answers label{

display: block;

}

#submit{

font-family: sans-serif;

font-size: 20px;

background-color: #279;

color: #fff;

border: 0px;

border-radius: 3px;

padding: 20px;

cursor: pointer;

margin-bottom: 20px;

}

#submit:hover{

background-color: #38a;

}

*5 užduotis.* SUKURTI DINAMIŠKĄ SĄVOKŲ SĄRAŠĄ.

Rezultatai:

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="style.css">

<title>WEB technologies</title>

</head>

<body>

<h1>WEB technologies</h1>

<div class="accordion">

<div class="accordion-header">HTML</div>

<div class="accordion-content">HTML is the standard markup language for creating Web pages.</div>

<div class="accordion-header">CSS</div>

<div class="accordion-content">CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media.</div>

<div class="accordion-header">JavaScript</div>

<div class="accordion-content">JavaScript is a scripting or programming language that allows you to implement complex things on web pages</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.js"></script>

<script src="script.js"></script>

</body>

</html>

2. Script.js

$(".accordion").on("click", ".accordion-header", function() {

$(this).toggleClass("active").next().slideToggle();

});

3. Style.css

html {

min-height: 100%;

font-family: 'Nunito', sans-serif;

-webkit-font-smoothing: antialiased;

}

body {

background: linear-gradient(to bottom right, #56ab2f, #a8e063);

line-height: 1.5;

}

h1 {

font-weight: 200;

font-size: 3rem;

color: white;

text-align: center;

}

.accordion {

background: linear-gradient(to bottom right, white, #F8F8F8);

max-width: 400px;

background: white;

margin: 0 auto;

box-shadow: 0 15px 20px -15px rgba(0, 0, 0, 0.3), 0 35px 50px -25px rgba(0, 0, 0, 0.3), 0 85px 60px -25px rgba(0, 0, 0, 0.1);

}

.accordion-header {

border-bottom: 1px solid #DDE0E7;

color: #2a313a;

cursor: pointer;

font-weight: 700;

padding: 1.5rem;

}

.accordion-header:hover {

background: #F6F7F9;

}

.accordion-content {

display: none;

border-bottom: 1px solid #DDE0E7;

background: #F6F7F9;

padding: 1.5rem;

color: #4a5666;

}

.accordion-header::before {

content: '';

vertical-align: middle;

display: inline-block;

width: .75rem;

height: .75rem;

border-radius: 50%;

background-color: #B1B5BE;

margin-right: .75rem;

}

.active.accordion-header::before {

background-color: #d66d75 ;

}

*6 užduotis.* SUKURKITE IŠŠOKANTĮ LANGĄ (MODAL BOX).

Rezultatai:

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="style.css">

<title>Modal jQuery</title>

</head>

<body>

<h1>Modal</h1>

<div class="overlay"></div>

<div class="modal">

<h2>Modal Box</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti perspiciatis magnam eum sed similique rerum laboriosam amet adipisci explicabo ipsa.</p>

<button class="close">Close</button>

</div>

<button class="open">Open</button>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.js"></script>

<script src="script.js"></script>

</body>

</html>

1. Script.js

$(".open").on("click", function () {

$(".overlay, .modal").addClass("active");

});

$(".close, .overlay").on("click", function () {

$(".overlay, .modal").removeClass("active");

});

$(document).keyup(function (e) {

if (e.keyCode === 27) {

$(".overlay, .modal").removeClass("active");

}

});

1. Style.css

html {

min-height: 100%;

font-family: 'Nunito', sans-serif;

-webkit-font-smoothing: antialiased;

}

body {

background: linear-gradient(to bottom right, #30cfd0, #f6d365);

line-height: 1.5;

text-align: center;

}

h1 {

font-weight: 200;

font-size: 3rem;

color: white;

text-align: center;

}

.overlay {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.5);

opacity: 0;

transition: .5s ease;

z-index: -1;

}

.overlay.active {

opacity: 1;

z-index: 2;

}

.modal {

max-height: calc(100% - 100px);

position: fixed;

top: 50%;

left: 50%;

max-width: 450px;

transform: translate(-50%, -50%);

color: #4a5666;

background: linear-gradient(to bottom right, white, #F8F8F8) !important;

box-shadow: 0 15px 20px -15px rgba(0, 0, 0, 0.3), 0 35px 50px -25px rgba(0, 0, 0, 0.3), 0 85px 60px -25px rgba(0, 0, 0, 0.1);

z-index: 3;

visibility: hidden;

opacity: 0;

transition: .5s ease;

}

.modal h2 {

margin: 0;

font-weight: 400;

padding: 1rem;

border-bottom: 1px solid #DDE0E7;

}

.modal p {

padding: 1rem;

}

.modal.active {

visibility: visible;

opacity: 1;

}

button {

display: inline-block;

background: rgba(10, 20, 30, .3);

border: 1px solid transparent;

color: white;

text-decoration: none;

font-size: 1.2rem;

padding: 1rem 2rem;

border-radius: 45px;

margin: .25rem 0;

vertical-align: middle;

line-height: 1;

overflow: visible;

white-space: nowrap;

cursor: pointer;

}

button:hover {

border: 1px solid rgba(255, 255, 255, .8);

color: white;

background: rgba(255, 255, 255, .1);

}

button.close {

margin: 0 0 1rem;

background: #FC6468;

}

button.close:hover {

border: 1px solid #FC6468;

color: #FC6468;

background: transparent;

}

*7 užduotis.* SUKURKITE DINAMIŠKĄ LENTELĘ SU GALIMYBE PRIDĖTI IR PAŠALINTI LENTELĖS EILUTES.

Rezultatai:

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS"

crossorigin="anonymous">

<title>Add/Remove Table Rows Dynamically</title>

</head>

<body>

<div class="container">

<div class="row mt-5">

<div class="col-10 offset-1">

<form>

<div class="form-group">

<input type="text" id="name" placeholder="Name">

<input type="text" id="email" placeholder="Email Address">

<input type="button" class="add-row btn btn-info" value="Add Row">

</div>

</form>

<table class="table table-striped">

<thead class="bg-info">

<tr>

<th>Select</th>

<th>Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td><input type="checkbox" name="record"></td>

<td>Name </td>

<td>name.surname@mail.com</td>

</tr>

</tbody>

</table>

<button type="button" class="delete-row btn btn-danger">Delete Row</button>

</div>

</div>

</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"

crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"

crossorigin="anonymous"></script>

<script src="script.js"></script>

</body>

</html>

1. Script.js

$(".add-row").on("click", function () {

var name = $("#name").val();

var email = $("#email").val();

$("#name, #email").val("");

var markup = "<tr><td><input type='checkbox' name='record'></td><td>" + name + "</td><td>" + email + "</td></tr>";

$("table tbody").append(markup);

});

$(".delete-row").on("click", function (){

$("table tbody").find('input[name="record"]').each(function () {

if ($(this).is(":checked")) {

$(this).parents("tr").remove();

}

});

});

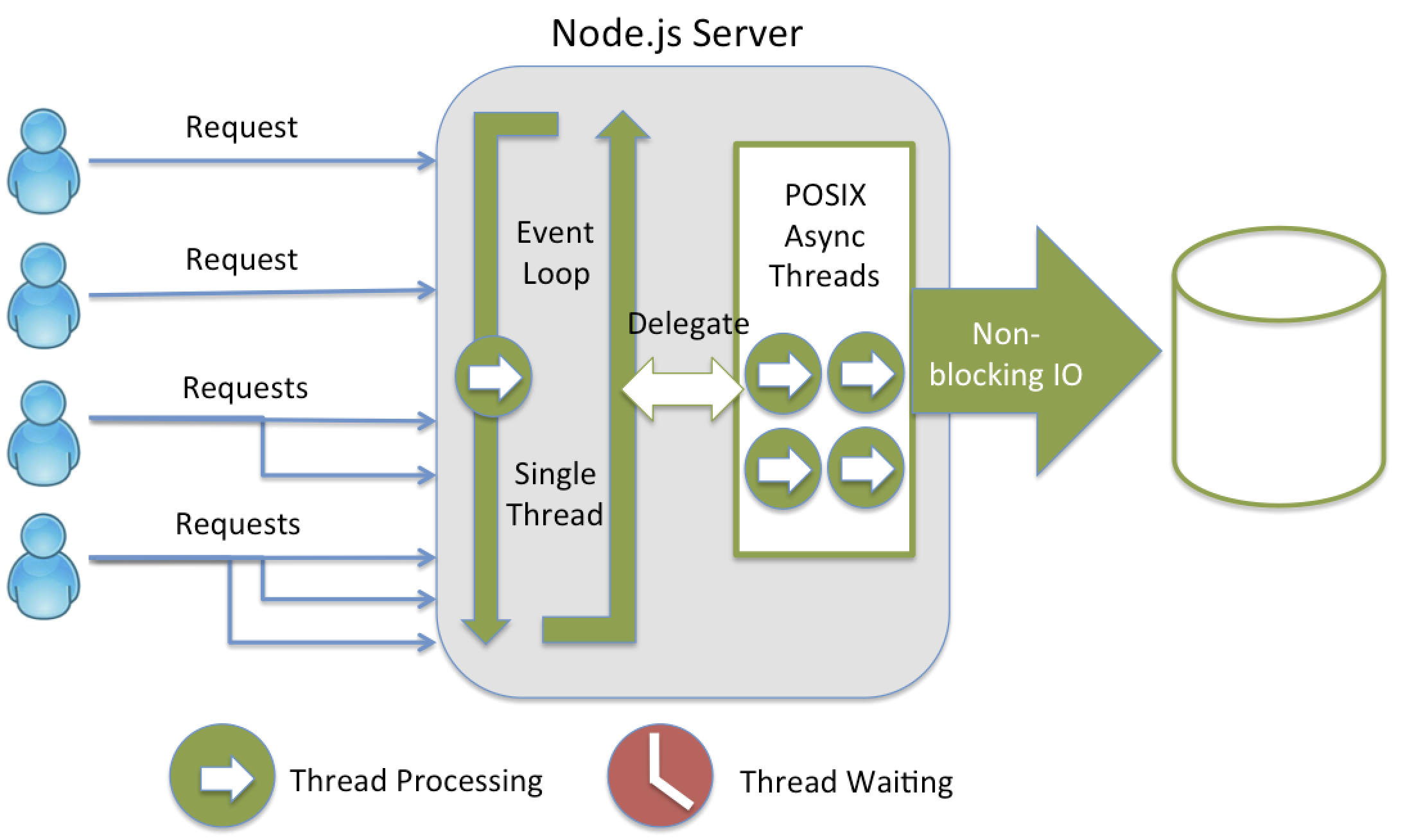
* 1. **Mokymosi rezultatas. Kurti paprastas serveryje vykdomas aplikacijas naudojant JavaScript.**

**3.9.1. Tema. NodeJS vykdymo aplinka.**

*1 užduotis*. KOKIE YRA NODE.JS APLINKOS NAUDOJIMO PRIVALUMAI IR SAVYBĖS?

Pagrindiniai „Node.js.“ naudojimo privalumai ir savybės:

* „Node.js“ yra asinchroninė valdymo aplinka. Ši funkcija reiškia, kad jei „Node.js“ priima įvesties / išvesties operacijos užklausą, ji bus atlikta foniniam režime ir bus tęsiamas kitų užklausų apdorojimas, tokiu būdu pagreitinant visų užklausų apdorojimą.
* „Node.js“ greitai apdoroja užklausas ir vykdo procesus. Jis naudoja „V8 JavaScript Runtime“ variklį, kurį naudoja „Google Chrome“.
* „Node.js“ naudoja vienos gijos modelį (angl. „single thread“) ir įvykių ciklą (angl. „event loop“). Kiekviena užklausa gali pasiekti vykdymo būseną serveryje nedelsiant. Tačiau tai netrukdo kitų užklausų vykdymui. Tokiu būdu „Node.js“ labiau lankstesnė ir efektyvesnė. Tradiciniai serveriai naudoja ribotą gijų skaičių, skirtų užklausoms vykdyti, o „Node.js“ sukuria vieną giją, kuri teikia paslaugas daug didesniam tokių užklausų skaičiui.
* „Node.js“ naudoja Node Package Manager (NPM). Išorinių bibliotekų integracija naudojant šį įrankį pagreitina kūrimo procesą.



Paveikslėlio šaltinis: <https://strongloop.com/strongblog/node-js-is-faster-than-java/>

Papildomai skaitykite:

Features of Node.js: <https://www.tutorialspoint.com/nodejs/nodejs_introduction>

*2 užduotis*. PARAŠYKITE KOMANDĄ, KURI IMPORTUOTŲ IŠORINĘ „HTTP“ BIBLIOTEKĄ NODE.JS PROJEKTE.

Komanda require naudojama išorinėms bibliotekoms importuoti. Pavyzdžiui: var http = require (“http”) – ši komanda importuoja “http” išorinę biblioteką.

Papildomai skaitykite:

NodeJS documentation. Import required modules. <https://www.tutorialspoint.com/nodejs/nodejs_first_application.htm>

*3 užduotis*. PARAŠYKITE, KOKIA BUS ŠIO NODE.JS PROGRAMINIO KODO IŠVESTIS. Išvestis:

|  |
| --- |
| **pirmas**  **trecias**  **antras** |

Papildomai skaitykite:

Node.js documentation. Timers in Node.js: <https://nodejs.org/de/docs/guides/timers-in-node/>

*4 užduotis*. PARAŠYKITE PROGRAMĄ NAUDODAMI ŠĮ FUNCKIONALUMO APRAŠYMĄ:

Atlikta užduotis:

var pilnamet = 18;

process.stdout.write('Įveskite savo amžių: '); //Vartotojo informavimas

process.stdin.on('data', function(ivestis) {

var amzius = parseInt(ivestis, 10);

if (isNaN(amzius) || amzius <= 0) { //Teisingos įvesties tikrinimas

console.log('%s yra netinkama įvestis', ivestis);

} else if (amzius < pilnamet) { //Amžiaus tikrinimas

console.log('Jums nėra %d metų, ' +

'sugrįžkite po %d metų',

pilnamet, pilnamet - amzius);

} else {

console.log('Sveiki atvyke! Jums yra %d metų', amzius);

}

process.stdin.pause();

});

Papildomai skaitykite:

Node.js documentation. Standart input and output. <https://nodejs.org/api/process.html#process_process_stderr>

*5 užduotis.* SUKURKITE NODE.JS HTTP WEB SERVERĮ.

Atlikta užduotis:

//Įkeliami moduliai

var http = require('http');

var fs = require('fs');

//Neegzistuojančio kelio funkcija

function neegzistuoja\_kelias(response) {

//Atsakas

response.writeHead(404, { 'Content-Type': 'text/plain' }); //HTTP statusas

response.write('Šis kelias neegzistuoja');

response.end(); //užbaigiamas atsakas

}

var server = http.createServer(function (req, res) { //Sukuriamas web serveris

if (req.method == 'GET' && req.url == '/') { //Teisingos užklausos tikrinimas

res.writeHead(200, { 'content-type': 'text/html' }); //HTTP statusas

fs.createReadStream('./index.html').pipe(res); //index.html failo įkelimas

}

else {

neegzistuoja\_kelias(res); //Neteisingos užklausos funkcijos iškvietimas

}

}).listen(3000); //Klausomasi 3000 prievado

console.log('HTTP serveris veikia!'); //Konsolės išvestis

Papildomai skaitykite:

Node.js documentation. HTTP Server and File System: <https://nodejs.org/api/http.html#http_response_writehead_statuscode_statusmessage_headers>, <https://nodejs.org/api/fs.html>

**3.9.2. Tema. Express karkasas.**

*6 užduotis*. KAS YRA TARPINĖ PROGRAMINĖ ĮRANGA (ANGL. „MIDDLEWARE“) „EXPRESS JS“ KARKASE? KOKIOS YRA TARPINĖS PROGRAMINĖS ĮRANGOS PAGRINDINĖS FUNKCIJOS?

Tarpinės programinės įrangos (angl. „middleware“) funkcijos turi prieigą prie užklausos objekto („req“), atsakymo objekto („res“) ir kitos tarpinės programinės įrangos funkcijos programos užklausos-atsakymo (angl. „request-response“) cikle.

Tarpinės programinės įrangos (angl. „middleware“) funkcijos gali atlikti šias užduotis:

* Vykdyti programinį kodą
* Atlikti užklausos ir atsakymo objektų pakeitimus.
* Užbaigti užklausos ir atsakymo (angl. „request-response“) ciklą.
* Kreiptis į kitą tarpinės funkciją.

Papildomai skaitykite:

Express.js documentation. Middleware: <https://expressjs.com/en/guide/using-middleware.html>

*7 užduotis.* UŽRAŠYKITE KELIŲ SUKŪRIMO (ANGL. „ROUTING“) STRUKTŪRĄ NAUDOJANT „EXPRESS JS“ KARKASĄ. PAAIŠKINKITE STRUKTŪROJE KIEKVIENĄ PANAUDOTĄ ELEMENTĄ.

Kelių kūrimo struktūra:

app.method(path, handler);

app – „Express js“ karkaso objektas.

method – HTTP užklausos metodas (metodo pavadinimas turi būti mažosiomis raidėmis).

path – universaliojo ištekliaus adreso (angl. „URL“) kelias. Nurodytu keliu bus sugeneruojamas pabaigos taškas (angl. „endpoint“).

handler – doroklė yra funkcija, kuri yra vykdoma tada kai yra kreipiamasi naudojant universaliojo ištekliaus adreso (angl. „URL“) kelią.

Papildomai skaitykite:

Express.js documentation. Routing <https://www.w3jar.com/express-js-routing/>

*8 užduotis*. SUPROGRAMUOKITE ŽINIATINKLIO SERVERĮ NAUDODAMI „EXPRESS JS“ KARKASĄ. ŠIS SERVERIS TURI ATITIKTI ŠIUOS REIKALAVIMUS.

Atlikta pavyzdinė užduotis:

const express = require("express");

const bodyParser = require("body-parser");

const app = express();

const { check, validationResult } = require('express-validator');

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.set("view engine", "ejs"); //įkeliama ejs biblioteka

//GET užklausos maršrutas

app.get("/registracija", function (req, res) {

res.render('index', { klaida: "" }); //įkeliamas index.ejs failas ir priskiriamas kintamasis

});

//POST užklausos maršrutas

app.post('/registracija', [

//nustatomi validžios informacijos standartai

check('vardas').isLength({ min: 3 }).withMessage('turite naudoti ilgesnis nei 3 simboliai'),

check('pastas').isEmail().withMessage('turite naudoti standarinę formą'),

check('amzius').isNumeric().withMessage('turite naudoti skaitinę vertę')

], (req, res) => {

//suformatuojama klaidos žinutė

const errorFormatter = ({ msg, param }) => {

return ` Klaidingai nurodytas ${param} - ${msg}`;

};

const klaida = validationResult(req).formatWith(errorFormatter);

if (!klaida.isEmpty()) { //tikrinama ar yra validacijos klaidų

res.render('index', { klaida: klaida.array() }); //įkeliamas index.ejs failas ir naudojamos klaidos žinutės

}

else {

//nukreipiama į sėkmingos registracijos žinutę

res.send("Sveikiname užsiregistravus!"

+ "<br> Vardas: " + req.body.vardas

+ "<br> El.paštas: " + req.body.pastas

+ "<br> Amžius: " + req.body.amzius);

}

});

app.listen(3000, function (){});

Papildomai skaitykite:

Modules documentation. Embedded JavaScript templating and validation <https://ejs.co/#install>, <https://express-validator.github.io/docs/index.html>

**3.9.3. Tema. Duomenų bazės.**

*9 užduotis*. SURAŠYKITE Į REIKIAMĄ LENTELĖS STULPELĮ IŠVARDINTAS DUOMENŲ BAZIŲ SAVYBES PRISKIRDAMI JAS ARBA REALIACINĖMS (SQL) ARBA NOSQL DUOMENŲ BAZĖMS.

|  |  |
| --- | --- |
| **Reliacinės (SQL) duomenų bazės** | **NoSQL duomenų bazės** |
| Iš anksto nustatyta schema. | Dinaminė schema. |
| Duomenys saugomi lentelėse. | Duomenys saugomi rakto-reikšmės porose (angl. key-value pairs)  Duomenys saugomi dokumentų saugyklose (angl. document based)  Duomenys saugomi grafų duomenų bazėse.  Duomenys saugomi kaip stulepelių saugykla (angl. wide-column stores) |
| Yra optimizuotos darbui su sudėtingomis užklausomis. | Gali dirbti su labai dideliais duomenų kiekiais realiu laiku. |
| Užtikrina duomenų vientisumą ir neprieštaringumą po kiekvienos transakcijos. | Užtikrina duomenų vientisumą ir neprieštaringumą tik praėjus tam tikram laikui. |

Papildomai skaitykite:

The Geek Stuff. SQL vs NoSQL Database Differences Explained with few Example DB. <https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db>

Geeks For Geeks. Difference between SQL and NoSQL. <https://www.geeksforgeeks.org/difference-between-sql-and-nosql/>

*10 užduotis*. ĮDIEKITE MONGODB DUOMENŲ BAZĘ. ĮDIEKITE NODE.JS MODULĮ, SKIRTĄ DARBUI SU MONGO DB.

Parašykite NODE.JS programą, kuri sukurtų duomenų bazę vardu „MYDB“, SUKURTU duomenų rinkinį vardu „CARS“ ir įkelkite į rinkinę 3 automobilius, sutekdami jiems gamintoją „BRAND“, modelį „MODEL“, SPALVĄ „COLOR“ ir numerį „NUMBER“, iš kurių bent du turėtų juodą spalvą

MongoDB įdiegimas – nuoroda MongoDB parsisiuntimui: <https://www.mongodb.com/download-center/community>

* Node.js modulio skirto MongoDB įdiegimas – naudodami Windows įrankį „Command Promt“ nueikite į katalogą, kuriame dirbsite su Node.js failais. (Pastaba: Turite būti įsidiegę NPM įrankį.) Komandinėje eilutėje paleiskite komandą „npm install mongodb“. Pvz.:

|  |
| --- |
| C:\Users\Your Name\Node>npm install mongodb |

* Node.js programa sukurianti įrašus faile mongodb\_create.js:

|  |
| --- |
| var MongoClient = require('mongodb').MongoClient;  var url = "mongodb://localhost:27017/";  MongoClient.connect(url, function(err, db) {  if (err) throw err;  var dbo = db.db("mydb");  var myobj = { brand: "BMW", model: "525", color: "black", number: "AAA 111" };  dbo.collection("cars").insertOne(myobj, function(err, res) {  if (err) throw err;  console.log("1 car inserted");  });    myobj = { brand: "Toyota", model: "Auris", color: "red", number: "ABC 123" };  dbo.collection("cars").insertOne(myobj, function(err, res) {  if (err) throw err;  console.log("1 car inserted");  db.close();  });  myobj = { brand: "Skoda", model: "Octavia", color: "black", number: "AAA 123" };  dbo.collection("cars").insertOne(myobj, function(err, res) {  if (err) throw err;  console.log("1 car inserted");  });  }); |

Komanda ir rezultatas:

|  |
| --- |
| C:\Users\Your Name\Node>node mongodb\_create.js  1 car inserted  1 car inserted  1 car inserted |

Papildomai skaitykite:

w3schools.com. Node.js MongoDB. <https://www.w3schools.com/nodejs/nodejs_mongodb.asp>

*11 užduotis*. PARAŠYKITE NODE.JS PROGRAMĄ, KURI SURASTŲ SUKURTOJE „MYDB“ DUOMENŲ BAZĖJE AUTOMOBILIUS, KURIE TURI JUODĄ SPALVĄ. KAIP REZULTATĄ PROGRAMA TURI ATSPAUSDINTI VISUS SURASTUS ĮRAŠUS SU VISAIS DUOMENŲ LAUKAIS.

Node.js programa surandanti įrašus. Failas „mongodb\_query.js“.

|  |
| --- |
| var MongoClient = require('mongodb').MongoClient;  var url = "mongodb://localhost:27017/";  MongoClient.connect(url, function(err, db) {  if (err) throw err;  var dbo = db.db("mydb1");  var query = { color: "black" };  dbo.collection("cars").find(query).toArray(function(err, result) {  if (err) throw err;  console.log(result);  db.close();  });  }); |

Komanda ir rezultatas:

|  |
| --- |
| C:\Users\Your Name\Node>node mongodb\_query.js  [ { \_id: 5d3e0f20917395eac071e061,  brand: 'BMW',  model: '525',  color: 'black',  number: 'AAA 111' },  { \_id: 5d3e121cddd19d025d68ae50,  brand: 'Skoda',  model: 'Octavia',  color: 'black',  number: 'AAA 123' } ] |

Papildomai skaitykite:

w3schools.com. Node.js MongoDB Query. <https://www.w3schools.com/nodejs/nodejs_mongodb_query.asp>

*12 užduotis*. ĮDIEKITE NODE.JS MONGOOSE MODULĮ.

Sukurkite NODE.JS programą, kuri sukurtų modelį CARS, turintį dvi savybes „BRAND“ IR „GAS“, bei sukurkite šio modelio įrašą ir jį išsaugokite į duomenų bazę „CARS“. Taip pat modeliui „CARS“ sukurkite metodą „PRINT“, kuris atspausdintų automobilio gamintoją „BRAND“ ir jo kuro kiekį „GAS“

Mongoose įdiegimo komanda:

|  |
| --- |
| C:\Users\Your Name\Node>npm install mongoose |

* Node.js programa sukurianti modelį, modelio įrašą ir jį išsauganti duomenų bazėje. Failas mongoose\_create.js:

|  |
| --- |
| var mongoose = require('mongoose');  mongoose.connect('mongodb://localhost:27017/cars', {useNewUrlParser: true});  var db = mongoose.connection;  db.on('error', console.error.bind(console, 'connection error:'));  db.once('open', function() {  var carSchema = new mongoose.Schema({  brand: String,  gas: Number  });    carSchema.methods.print = function () {  var cartext = this.brand  ? "Car brand is " + this.brand + " and has " + this.gas + " l. gas"  : "I don't have a brand";  console.log(cartext);  }  var Car = mongoose.model('Car', carSchema);    var car1 = new Car({ brand: 'BMW', gas: 50 });    car1.save(function (err, fluffy) {  if (err) return console.error(err);  console.log("Car saved: " + car1.brand);  car1.print();  db.close();  });    }); |

Komanda ir rezultatas:

|  |
| --- |
| C:\Users\Your Name\Node>node mongoose\_create.js  Car saved: BMW  Car brand is BMW and has 50 l. gas |

Papildomai skaitykite:

Mongoose Documentation. Getting Started. <https://mongoosejs.com/docs/index.html>

*13 užduotis*. NAUDODAMI NODE.JS MONGOOSE MODULĮ, SUKURKITE PROGRAMĄ, KURI PRISIJUNGTŲ PRIE SUKURTOS DUOMENŲ BAZĖS „CARS“ BEI SURASTŲ ĮRAŠĄ IŠ ŠIOS DUOMENŲ BAZĖS.

Node.js programa surandanti duomenų bazėje įrašą. Failas mongoose\_db\_find.js:

|  |
| --- |
| var mongoose = require('mongoose');  mongoose.connect('mongodb://localhost:27017/cars', {useNewUrlParser: true});  var db = mongoose.connection;  db.on('error', console.error.bind(console, 'connection error:'));  db.once('open', function() {  var carSchema = new mongoose.Schema({  brand: String,  gas: Number  });    var Car = mongoose.model('Car', carSchema);    Car.find(function (err, cars) {  if (err) return console.error(err);  console.log("Found: " + cars);  db.close();  })    }); |

Komanda ir rezultatas:

|  |
| --- |
| C:\Users\Your Name\Node>node mongoose\_db\_find.js  Found: { \_id: 5d3e256c9c342250f457deb3, brand: 'BMW', gas: 50, \_\_v: 0 } |

Papildomai skaitykite:

Mongoose Documentation. Getting Started. <https://mongoosejs.com/docs/index.html>

* 1. **Mokymosi rezultatas. Kurti aplikacijas naudojant React karkasą.**

**3.10.1. Tema. Populiarių karkasų palyginimas.**

*1 užduotis***.** PASIRINKITE TEISINGĄ ATSAKYMĄ**:**

1) d

2) c

3) b

4) a

5) b

6) a

7) c

8) b

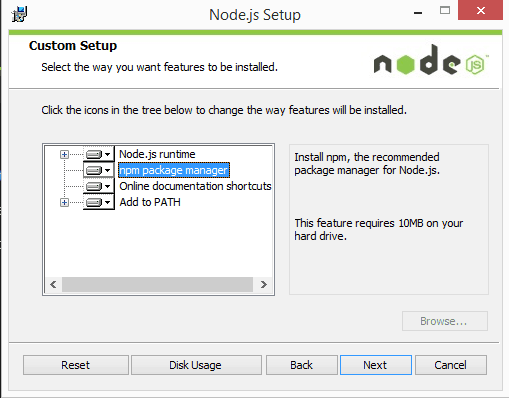
**3.10.2. Tema. React karkasas**

*2 užduotis*. ĮDIEKITE NODE JS SAVO KOMPIUTERYJE.

1. Atsisiųsti diegimo faila iš <http://nodejs.org/>

2. Paleisti diegimo failą ir sekti instrukcijas

3. Perkrauti kompiuterį



4. Patikrinti veikimą

node –v

*3 užduotis*. https://reactjs.org/community/examples.html TINKLAPYJE YRA PATEIKTI DEMO PROJEKTAI SU NUORODA Į GITHUB. IŠSIRINKITE VIENĄ PROJEKTĄ IR NUSIKLONUOKITE. PAANALIZUOKITE KODĄ.

Paleiskite projekto aplikaciją savo kompiuteryje naudojant NPM.

git clone <https://github.com/Rhymond/product-compare-react.git> .

npm install

npm start

*4 užduotis*. SUKURKITE NAUJĄ REACT APLIKACIJĄ PAVADINIMU „KNYGOS“.

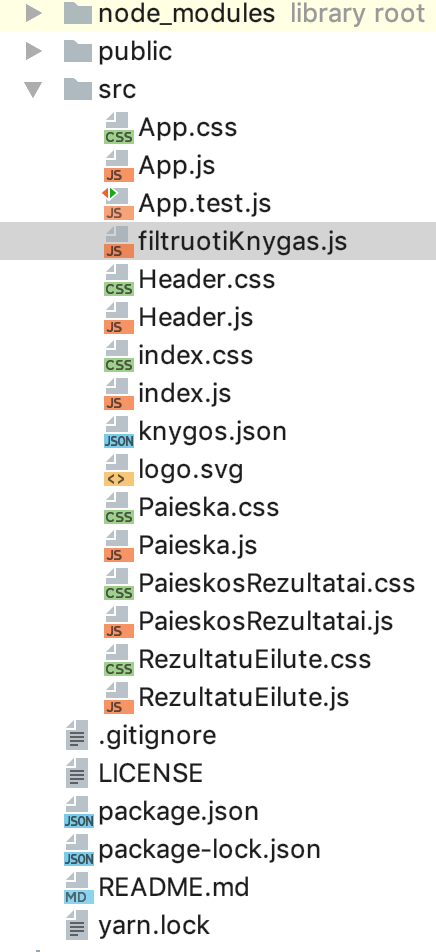
**Aplikacijos sukūrimas**

npx create-react-app knygos

cd knygos

npm start

**Projekto struktūra:**



**App.js**

import React, { PureComponent } from "react";

import { BrowserRouter as Router, Route, Link } from "react-router-dom";

import Header from "./Header";

import Paieska from "./Paieska.js";

import PaieskosRezultatai from "./PaieskosRezultatai";

import filtruotiKnyga from "./filtruotiKnygas";

export default class App extends PureComponent {

constructor(props) {

super(props);

this.state = {

filtruotosKnygos: filtruotiKnyga("", 20)

};

}

handleSearchChange = event => {

this.setState({

filtruotosKnygos: filtruotiKnyga(event.target.value, 20)

});

};

render() {

return (

<div>

<Header />

<Paieska textChange={this.handleSearchChange} />

<PaieskosRezultatai knyguDuomenys={this.state.filtruotosKnygos} />

</div>

);

}

}

**filtruotiKnygas.js**

import knyguSarasas from "./knygos.json";

export default function filtruotiKnyga(searchText, maxResults) {

return knyguSarasas

.filter(knyga => {

if (knyga.pavadinimas.toLowerCase().includes(searchText.toLowerCase())) {

return true;

}

if (knyga.keywords.includes(searchText)) {

return true;

}

return false;

})

.slice(0, maxResults);

}

**Header.js**

import React, { PureComponent } from "react";

import "./Header.css";

export default class Header extends PureComponent {

render() {

return (

<Router>

<header className="component-header">

<nav>

<ul><li><Link to="/apie/">Apie projektą</Link></li></ul>

</nav>

Paieška knygų kataloge

</header>

<Route path="/" exact component={Index} />

<Route path="/about/" component={Apie} />

</Router>

);

}

}

**Index.js**

import React from "react";

import ReactDOM from "react-dom";

import App from "./App";

import "./index.css";

import "github-fork-ribbon-css/gh-fork-ribbon.css";

ReactDOM.render(<App />, document.getElementById("root"));

**knygos.js**

[

{

"pavadinimas": "Programavimas linksmai",

"kodas": "KN-12334",

"aprasymas": "Netradicinis programavimo mokymas"

},

{

"pavadinimas": "C++",

"kodas": "KN-12356",

"aprasymas": "Programavimas c++ programavimo kalba"

},

{

"pavadinimas": "PHP",

"kodas": "KN-12388",

"aprasymas": "Programavimas PHP programavimo kalba"

},

{

"pavadinimas": "SASS",

"kodas": " KN-15388",

"aprasymas": "SASS preprocesoriaus naudojimas"

}

]

**Paieska.js**

import React, { PureComponent } from "react";

import PropTypes from "prop-types";

import "./Paieska.css";

export default class Paieska extends PureComponent {

static propTypes = {

textChange: PropTypes.func

};

handleChange = event => {

this.props.textChange(event);

};

render() {

return (

<div className="component-search-input">

<div>

<input onChange={this.handleChange} />

</div>

</div>

);

}

}

**PaieskosRezultatai.js**

import React, { PureComponent } from "react";

import PropTypes from "prop-types";

import Clipboard from "clipboard";

import RezultatuEilute from "./RezultatuEilute";

import "./RezultatuEilute.css";

export default class KnyguRezultatai extends PureComponent {

static propTypes = {

knyguDuomenys: PropTypes.array

};

componentDidMount() {

this.clipboard = new Clipboard(".copy-to-clipboard");

}

componentWillUnmount() {

this.clipboard.destroy();

}

render() {

return (

<div className="component-emoji-results">

{this.props.knyguDuomenys.map(knyguDuomenys => (

<RezultatuEilute

key={knyguDuomenys.pavadinimas}

symbol={knyguDuomenys.kodas}

title={knyguDuomenys.pavadinimas}

/>

))}

</div>

);

}

}

**PaieskosRezultatuEilute.js**

import React, { PureComponent } from "react";

import PropTypes from "prop-types";

import "./RezultatuEilute.css";

export default class EmojiResultsRow extends PureComponent {

static propTypes = {

pavadinimas: PropTypes.string,

kodas: PropTypes.string

};

render() {

return (

<div>

<span className="title">{this.props.pavadinimas}</span>

<span className="info">Kopijuoti knygos kodą</span>

</div>

);

}

}

1. **Kompetencija. Analizuoti skirtingų tipų reikalavimus, apibūdinančius kompiuterinę programą.**

**4.1. Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo reikalavimų programinei įrangai formatą.**

**4.1.1. Tema. Reikalavimų programinei įrangai formatai.**

*1* *užduotis.* REIKALAVIMŲ PROGRAMINEI ĮRANGAI FORMATAI.

Reikalavimas yra apibrėžtas kaip specifikacija. Panaudojimo atvejis šios specifikacijos yra pateikiamas atskirai.

Reikalavimui ištestuoti gali būti naudojami šie testavimo atvejai:

* Prisijungti prie sistemos nurodant neteisingą slaptažodį slaptažodį -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos nurodant neteisingą slaptažodį, pakartoti kelis kartus, patikrinti, ar paskyra blokuojama;
* Patikrinti, ar yra galimybė atblokuoti paskyrą atstatant slaptažodį (kokiu būdu atblokavimas vyksta, nenurodyta reikalavime);
* Patikrinti visus vartotojo registravimo formos įvedimo laukus (formatus, ilgius, būtinumo požymį);
* Prisijungti prie sistemos su neregistruotu vartotoju -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos su egzistuojančiu vartotoju, suvedant neteisingą slaptažodį -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos su registruotu vartotoju -> atidaromas pagrindinis sistemos langas su priregistravimo kortele;
* Registruotam vartotojui pakoreguoti profilio duomenis (kokie konkrečiai duomenys gali būti keičiami nenurodoma)

*2 užduotis.* UŽRAŠYKITE VARTOTOJO PASAKOJIMO (USER STORY) FORMATĄ.

Pateikite pavyzdžių

Vartotojo pasakojimo formatas:

**Kaip** <rolė>

**aš noriu** <veiksmas>

**tam, kad** <vertė>.

**Rolė** - aktyvus sistemos dalyvis - žmogus arba automatinė sistema.

**Veiksmas** - veiksmas kuris turi įvykti sistemoje.

**Vertė** - atspindi vertę verslui. Kam to reikia?

**Pavyzdžiai:**

1. Kaip personalo vadovas aš noriu turėti atrankos klausimyną tam, kad galėčiau suprasti, ar reikia perduoti kandidatą funkciniam vadovui.

2. Kaip vadovas aš noriu turėti galimybę peržiūrėti jau sukurtus klausimus tam, kad prisiminčiau, kokius klausimus jau turiu ir nuspręsti, ar pozicijai, kuriai šiuo metu vykdau atranką, galiu juos vėl panaudoti ar reikia atnaujinti esamą klausimyną.

3. Kaip sistemos vartotojas aš noriu pažymėti failus kurių nereikia kopijuoti, kad mano atsarginės kopijos erdvė neužsipildytų dalykais, kurių aš nenoriu išsaugoti.

Papildomai galite paskaityti:

* <https://en.wikipedia.org/wiki/User_story>

*2 užduotis.* YRA PATEIKTI FUNKCINIAI PROGRAMINĖS ĮRANGOS REIKALAVIMAI. PABANDYKITE SUFORMUOTI TRIS VARTOTOJO PASAKOJIMUS (USER STORIES*).*

*3 užduotis.*  YRA PATEIKTI FUNKCINIAI PROGRAMINĖS ĮRANGOS REIKALAVIMAI. PABANDYKITE SUFORMUOTI TRIS VARTOTOJO PASAKOJIMUS (USER STORIES*).*

Vartotojo pasakojimo pavyzdžiai pagal funkcinius reikalavimus:

* Aš kaip administratorius noriu registruoti naujus vartotojus sistemoje.
* Aš kaip administratorius norius keisti vartotojų slaptažodžius.
* Aš kaip vartotojas noriu prisijungti prie sistemos su savo vartotojo vardu ir slaptažodžiu

*4 užduotis.* APRAŠYKITE GERŲ VARTOTOJO PASAKOJIMO FORMATĄ (INVEST).

INVEST vartotojo pasakojimai turintys šias savybes:

* Nepriklausomi (independent) - pasakojimai turėtų būti nepriklausomi vienas nuo kito.
* Aptariami (negotiable) - pasakojimai aptariami ir diskutuojami.
* Vertingi (valuable) - turi būti vertingi klientui arba pirkėjui.
* Įvertinami (estimable) - turi būti įmanoma pasakojimus įvertinti.
* Pakankamai maži (small enough) - rekomenduojamas dydis 1-5 dienos.
* Patikrinami (testable) - kiekvieną pasakojimą turi būti įmanoma patikrinti (ištestuoti).

**4.1.2. Tema. Reikalavimų peržiūros procesas.**

*4* *užduotis.* REIKALAVIMŲ PERŽIŪROS PROCESAS.

**Vartotojo pasakojimas:** Kaip prisiregistravęs sistemos vartotojas noriu išsiregistruoti iš sistemos paspausdamas mygtuką „Atsijungti“, nepriklausomai nuo to, kokį darbą su sistema būčiau dirbęs.

**Reikalavimo paruošimo kriterijai:** vartotojo pasakojimas yra apibrėžtas aiškiai, yra aiškus ir vienareikšmiškai suvokiamas kontekstas. Priėmimo kriterijai yra aiškūs, testuotini.

**Priėmimo kriterijų pavyzdžiai:**

* Atsijungimo nuo sistemos mygtukas rodomas tik prisijungusiems vartotojams;
* Paspaudus “Atsijungti” mygtuką, Sistema atjungia vartotoją, uždaro jo pradėtus darbus ir atvaizduoja pradinį puslapį;
* Jei vartotojas atsijungia nuo sistemos nepabaigęs kokio nors darbo, jo pradėti, bet nepabaigti darbai nėra išsaugomi, juos reikės atlikti iš naujo;
* Atsijungęs vartotojas vėl mato “Prisijungti” mygtuką ir gali registruotis vėl;
* Vartotojas gali atsijungti nuo sistemos nepriklausomai nuo to, kokiame sistemos lange jis yra.

**4.2. Mokymosi rezultatas. Naudoti funkcinius, nefunkcinius ir techninius kompiuterinės programos reikalavimus.**

**4.2.1. Tema. Funkciniai reikalavimai.**

*1* *užduotis.* FUNKCINIAI REIKALAVIMAI.

Opencart – Paieška (supaprastinti paieškos funkcionalumo reikalavimai)

- Jei į paieškos lauką įvedamas produkto tikslus pavadinimas ir spaudžiamas "Paieška" mytgtukas, sistema pateikia sąrašą visų šio produkto atmainų, turimu pardavime. OK (atitinka)

- jei įvedamas raktažodis pagal produkto kategoriją ir spaudžiamas "Paieška" mygtukas, sistema pateikia atitinkamos kategorijos produktų sąrašą; NOK (Not OK, neatitinka).

Komentaras: Sistema pateikia tik vieną telefoną, nors <Phones & PDA‘s> kategorijoje yra daugiau produktų. Priežastis gali būti netinkama produkto konfigūracija.

- Jei į paieškos laukelį neįvedame nieko ir spaudžiame "Paieška" mygtuką, sistema grąžiną visu turimų prekių sąrašą. NOK.

Komentaras: Nieko neįvedus į paieškos laukelį ir paspaudus paieškos mygtuką, sistema nieko nepateikia. Klaida gali būti arba reikalavimuose (netinkamai aprašyti, nesuprasti), arba implementacijoje, arba CMS paieškos modulio konfigūracijoje.

- Jei į paieškos laukelį įvedame neegzistuojančio produkto pavadinimą ir spaudžiame "Paieška" mygtuką, sistema pateikia pranešimą, apie tokio produkto nebuvimą. OK.

Komentaras: Bet, jei įvesime tik fragmentą, kuris atitiks kurio nors egzistuojančio produkto pavadinimo dalį, šis produktas bus atvaizduotas paieškos rezultatuose.

- Jei paieškos rezultatų yra daug, jie yra skirstomi į puslapius po 10 rezultatų kiekviename puslapyje. NOK.

Komentaras: Pagal nutylėjimą produktai nėra skirstomi puslapiais. Tačiau turi būti patikrinti CMS paieškos modulio nustatymai. Gali būti, kad puslapiavimas yra implementuotas, tačiau neįjungtas.

- Paieškos rezultatai yra surūšiuoti abėcėlės tvarka. OK

**4.2.2. Tema. Nefunkciniai reikalavimai**

*2* *užduotis.* NEFUNKCINIAI (TECHNINIAI, SAUGUMO) REIKALAVIMAI.

1. Funkciniai reikalavimai testuoja sistemos funkcijas (skaičiavimus, logiką ir pan), nefunkciniai – sistemos charakteristikas (greitį, išvaizdą, saugumą ir pan.)
2. Saugumo, našumo, atsparumo, suderinamumo, panaudojamumo, t.t.
3. Iš esmės, atliekant testavimą su skirtingomis naršyklėmis, testuojame svetainės išvaizdą ir funkcionalumą įvairiose naršyklėse ir įrenginiuose. Kaip žmonės interpretuoja dalykus skirtingai, taip ir ne visos naršyklės “supranta” dalykus vienodai. Pvz. „Internet Explorer 8“ CSS stiliai gali būti skirtingi nei naujesnėse „Internet Explorer“ ir „Google Chrome“ versijose. Elegantiškas stilius ir efektai, vaizdai, šešėliai ir netgi šriftai, gali labai skirtis. Nors šiuos vaizdinius skirtumus dažnai neįmanoma panaikinti visiškai (ypač su senesnėmis naršyklėmis), testavimo su skirtingomis naršyklėmis ir įrenginiais tikslas yra užtikrinti, kad naudotojai galėtų pasiekti visą turinį ir vykdyti visas pagrindines žiniatinklio programos funkcijas be jokių svarbių problemų.
4. Galima matuoti puslapio našumą /užkrovimo greitį, testuoti turinio pasikeitimus, SEO žymas (angl. tags), atlikti nesudėtingą puslapio derinimą (ang. debugging).

|  |  |
| --- | --- |
| Funkciniai reikalavimai:  b,c, d, f, h, | Nefunkciniai reikalavimai:  a, e, g, i, j |

1. Reikalavimų peržiūra yra labiau nefunkcinis testavimas. Funkcinis testavimas yra tiesiogiai susijęs su sistemos vykdymu. Reikalavimų peržiūra yra statinės analizės dalis, kuri nėra priskiriama prie funkcinio testavimo.
2. a), b), c), d)
3. c)
4. Panaudojamumas – tai naudotojo veiklos veiksmingumas, našumas ir jaučiamas pasitenkinimas, su kuriuo konkretus naudotojas gali pasiekti konkrečių tikslų konkrečiose aplinkose, šiuo atveju, naudodamasis mūsų sistema.
5. a)

**4.3. Mokymosi rezultatas. Testuoti programinę įrangą naudojant su JavaScript programavimo kalba suderinamus testavimo įrankius ir metodus.**

* + 1. **Tema. Testavimo principai.**

*1* *užduotis.*  DUOTA FUNKCIJA. KIEK REIKĖS TESTAVIMO SCENARIJŲ, KAD PADENGTI VISUS ATVEJUS. IŠVARDINKITE JUOS.

Atsakymas: 3

1. a < b

2. a > b & a > c

3. a > b & a < c

2 *užduotis.*  IŠVARDINKITE PAGRINDINES RIZIKAS/ NAUDAS, KODĖL SVARBU PROGRAMOS KLAIDĄ SURASTI KUO ANKSČIAU.

Atsakymas:

a.      Mažesnė tikimybė kad klaida pasikartos per kelias vietas

b.      Pigiau ir greičiau ištaisoma klaida

**4.3.2. Modulių (unit) testavimas.**

3 *užduotis.*  ATSAKYKITE Į ŽEMIAU PATEIKTUS KLAUSIMUS. Kiekvienas klausimas gali turėti vieną arba daugiau teisingų atsakymų.

1. a
2. b
3. a
4. d

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atsakinga grupė | Komponentų testavimas | Integracijos testavimas | Sistemos testavimas | Priėmimo testavimas |
| Programuotojai | x | x |  |  |
| Testuotojai |  | x | x | x |
| Verslo atstovai / galutiniai vartotojai |  |  | x | x |

1. c
2. c
3. a
4. c
5. d
6. b

**4.3.3. Pilnas (End-to-end) testavimas.**

*4* *užduotis.*  ATSAKYKITE Į ŽEMIAU PATEIKTUS KLAUSIMUS. Kiekvienas klausimas gali turėti vieną arba daugiau teisingų atsakymų

|  |  |  |
| --- | --- | --- |
| Funkcinis testavimas  Priėmimo testavimas  Komponentų testavimas | Nefunkcinis testavimas  Saugumo testavimas  Našumo testavimas  Apkrovos testavimas  Prieinamumo testavimas  Lokalizavimo testavimas | Palaikymo testaviams  Palaikymo testavimas |

1. b
2. a,c
3. c
4. Nėra teisingo atsakymo
5. b,c

|  |  |
| --- | --- |
| Juodosios dėžės testavimo metodai:  Atsitiktinis testavimas  Testavimas pagal reikalavimų specifikaciją  Apkrovos testavimas  Našumo testavimas  Scenarijų testavimas  Modifikuotas sprendimų testavimas  Panaudos atvejų testavimas | Baltosios dėžės testavimo metodai:  Sakinių testavimas  Atšakų testavimas  Kelių testavimas |

1. b
2. b
3. b
4. b.

**Modulis „Nesudėtingų duomenų bazių projektavimas ir kūrimas (JavaSript)“**

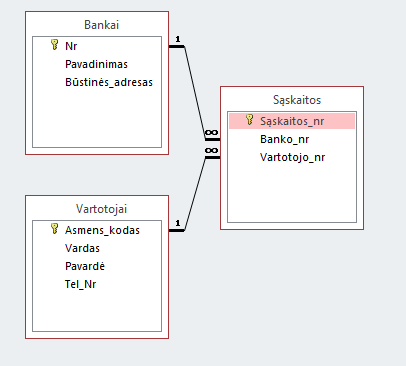
**1.** **Kompetencija. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes.**

**1.1.** **Mokymosi rezultatas. Pateikti reliacinės duomenų bazės schemą.**

**1.1.2. Tema. Duomenų bazių projektavimas (CREATE TABLE sakinys, duomenų normalizavimas).**

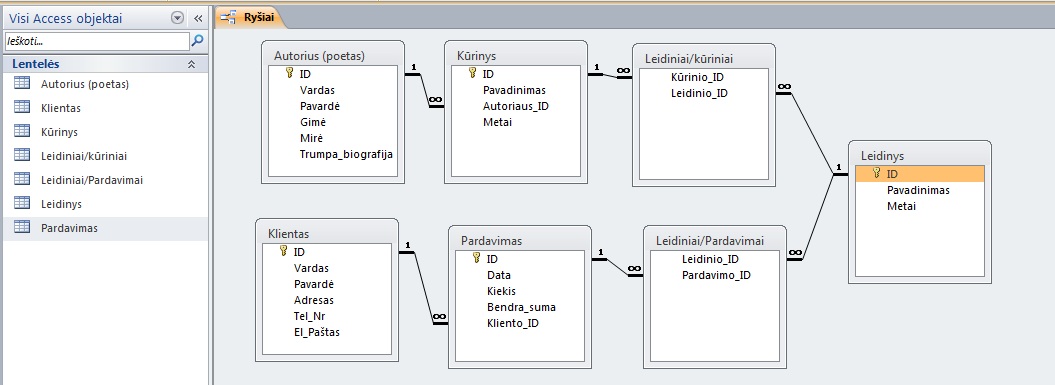
*1 užduotis*. SUPROJEKTUOKITE RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LIETUVOS BANKAS).

Schema:



*2 užduotis*. SUPROJEKTUOKITE RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LEIDYBOS STEBĖJIMO SISTEMA).

Schema:



**1.2.**  **Mokymosi rezultatas. Pateikti nereliacinės (NoSQL) duomenų bazės schemą.**

**1.2.2. Tema. NoSQL duomenų bazių tipai ir jų savybės.**

*1 užduotis*. PATEIKITE KEY-VALUE DUOMENŲ BAZĖS MODELĮ PARDUOTUVEI.

Įvertinkite, kaip saugoti sudėtinius raktus, kaip saugoti reikšmes, ryšius tarp prekių ir parduotuvės

Parduotuvės atveju, raktas – parduotuvės kodas. Reikšmę pasirinkti saugoti galima JSON, XML, Protobuf ar kitu viešai prieinamu formatu.

Prekių parduotuvėje atveju, raktas sudaromas sujungiant parduotuvės kodas su prekės kodu atskiriant jas skirtuku, pavyzdžiui brūkšniu (pvz.: V1-11559245), reikšmė kaip ir parduotuvės atveju, saugoma vienu iš plačiai prieinamų formatų.

Rekomenduojama rinktis Redis duomenų bazę.

*2 užduotis*. PATEIKITE COLUMN FAMILY DUOMENŲ MODELĮ PARDUOTUVĖS SCENARIJUS.

Schema:

CREATE KEYSPACE shopspace WITH replication = {'class':'SimpleStrategy', 'replication\_factor' : 1};

CREATE TABLE shopspace.shops (code TEXT PRIMARY KEY, area INT, address TEXT);

CREATE TABLE shopspace.items (shop\_code TEXT, item\_code INT, name TEXT, unit\_weight DECIMAL, unit\_price DECIMAL, count INT, PRIMARY KEY(shop\_code, item\_code));

shopspace.items lentelės atveju particijos raktas parduotuvės kodas, rūšiavimo raktas – prekės kodas.

Užklausos:

1. SELECT \* FROM shopspace.shops WHERE code = 'V1';
2. SELECT \* FROM shopspace.items WHERE shop\_code = 'V1';
3. SELECT count FROM shopspace.items WHERE shop\_code = 'V1' AND item\_code = 11;

**1.2.3. Tema. NoSQL duomenų bazių valdymas.**

*3 užduotis*. PARDUOTUVĖJE ATSIRADO NAUJAS REIKALAVIMAS – PAGAL PREKĖS KODĄ GAUTI PARDUOTUVIŲ, KURIOSE YRA ŠI PREKĖ, SĄRAŠĄ SU PREKĖS VIENETŲ SKAIČIUMI.

Kaip pakeisti duomenų bazės schemą ir darbo su ja užklausas šiam reikalavimui įgyvendinti?

Kadangi Cassandra leidžia saugiai užklausas vykdyti tik pagal raktą, sukuriame papildomą lentelę:

CREATE TABLE shopspace.shopitems (item\_code INT, shop\_code TEXT, count INT, PRIMARY KEY(item\_code, shop\_code));

Tuomet užklausa atrodo taip:

SELECT shop\_code, count FROM shopspace.shopitems WHERE item\_code = 257;

Įterpimas keičiasi vykdant įterpimą į abi lenteles BATCH sakinio pagalba. Pavyzdžiui:

BEGIN BATCH

INSERT INTO shopspace.shopitems (item\_code, shop\_code, count) VALUES (45, 'V2', 2);

INSERT INTO shopspace.items (shop\_code, item\_code, name, unit\_weight, unit\_price, count) VALUES ('V2', 45, 'Skanioji Duona', 0.9, 1.2, 4);

APPLY BATCH;

Alternatyva - galima panaudoti antrinius indeksus.

*4 užduotis*. SUMODELIUOKITE PARDUOTUVĖS SCENARIJŲ DOKUMENTŲ DUOMENŲ BAZĖJE.

Priimtini sprendimai tiek su viena, tiek su keliomis kolekcijomis.

Pavyzdžiui sudedant į vieną kolekciją, parduotuvė su jos prekėmis galėtų atrodyti taip:

{

\_id: “V1”,

area: 2500,

address: “Linkmenu g. 3, Vilnius”,

items: [

{ code: 11559245, name: “Skalsioji duona”, unit\_weight: 1.2, unit\_price: 1.7, count: 11 },

...

]

}

*5 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS.

db.getCollection('shops').find({\_id: "V1"});

1. db.getCollection('shops').find({\_id: "V1"}, {items: 1});
2. db.getCollection('shops').find({"items.code": 23});
3. db.getCollection('shops').aggregate([{$unwind: "$items"}, {$group: {\_id: "$\_id", total: {$sum: "$items.count"}}}]);

Jei modeliuojama parduotuvė su prekėmis viename dokumente, tai tiek užklausa #1, tiek #2 grąžins visas prekes.

*6 užduotis*. SUMODELIUOKITE PARDUOTUVĖS SCENARIJŲ GRAFŲ DUOMENŲ BAZĖJE.

Parduotuvės ir prekės modeliuojamos kaip viršūnės su skirtingomis žymomis ir susiejamos ryšiu.

create (V1:Shop {code: 'K1', area: 3000, address: 'Kauno g. 11'})

create (i11559245:Item {code: 11559245, name: 'Skalsioji duona', unit\_weight: 1.2, unit\_price: 1.8, count: 11})

match (s:Shop),(i:Item)

where s.code = 'K1' AND i.code = 2445

create (s)-[r:HAS]->(i)

return s,i

*7 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS:

1. match (s:Shop {code:'V1'}) return s
2. match (s:Shop {code:'V1'})-[r:HAS]-(i:Item) return i
3. match (s:Shop)-[r:HAS]-(i:Item {code: 1445}) return s
4. match (s:Shop)-[r:HAS]-(i:Item) return s,sum(i.count)

*8 užduotis*. ATSIRADO PAPILDOMAS REIKALAVIMAS SUSIETI PREKES SU GAMINTOJO INFORMACIJA.

Kiekvienas gamintojas identifikuojamas pagal įmonės pavadinimą, kartu saugoma gamintojo valstybė. Prekės, gaminamos parduotuvėje, nėra susiejamos su gamintoju.

Grafas papildomas naujais elementais pažymėtais žymomis Producer, prekėms sukuriamas ryšys pažymėtas PRODUCEDBY.

create (p2:Producer {name: "Makers", country: "Germany"})

match (i:Item {code: 2445}),(p:Producer {name: "Makers"})

create (i)-[r:PRODUCEDBY]->(p)

return i,p

Užklausos:

1. match (s:Shop)-[r:HAS]-(i:Item)-[pb:PRODUCEDBY]-(p:Producer {country: 'Lithuania'}) return distinct s
2. match (s:Shop)-[r:HAS]-(i:Item)-[pb:PRODUCEDBY]-(p:Producer {country: 'Lithuania'}) where i.unit\_price > 1 return distinct s

**2. Kompetencija. Programiškai įgyvendinti ir administruoti duomenų bazes.**

* 1. **Mokymosi rezultatas. Diegti ir valdyti duomenų bazių valdymo sistemą.**

**2.1.1. Tema. DBVS diegimas.**

*1* *užduotis*. PRISIJUNKITE PRIE DUOMENŲ BAZIŲ VALDYMO SISTEMOS NAUDODAMI TELNET.

Naudokite komandą

mysql -h hostname -u user\_name -p

hostname ir user\_name reikia pakeisti atitinkamais duomenimis, kuriuos, jei pats neturite administratoriaus teisių, turėtumete gauti iš serverio administratoriaus, sukūrusio jums duomenų bazę. hostname yra serverio adresas, kuriame yra instaliuota MySQL DBVS. Jei duomenų bazė yra tame pačiame serveryje (kompiuteryje), iš kurio jungiatės, jis turėtų būti localhost. Jei jungiatės prie kito (remote) kompiuterio, reikia nurodyti to kompiuterio adresą. -p komanda nurodo MySQL DBVS, kad jos klientas paprašyti slaptažodžio. Jei nenurodysite šios komandos, tuomet MySQL manys, kad jūsų vartotojo vardui nereikalingas slaptažodis prisijungimui prie sistemos. Jei jūs nesate privilegijuotas vartotojas (t.y. administratorius) nenurodžius -p jūs nebūsite prijungtas prie sistemos. Komanda quit atjungia jus nuo serverio.

*2 užduotis*. PAKARTOKITE TUOS PAČIUS VEIKSMUS NAUDODAMI PHPMYADMIN.

Naudodami phpMyAdmin atitinkamus meniu, prisijunkite ir atsijunkite.

**2.2.**  **Mokymosi rezultatas. Naudoti SQL kalbą duomenų bazės užpildymui ir informacijos išrinkimui.**

**2.2.1. Tema. Duomenų išrinkimas naudojant SQL select sakinį ir pagrindinius select elementus.**

*1 užduotis*. PANAUDOKITE TINKAMAS PAGRINDINES SQL KOMANDAS.

SQL komandos:

1. SELECT \* FROM DARBUOTOJAI;
2. SELECT ASMENSKODAS FROM DARBUOTOJAI;
3. SELECT VARDAS, PAVARDE, PAREIGOS FROM DARBUOTOJAI;
4. SELECT DISTINCT SKYRIAUSPAVADINIMAS FROM DARBUOTOJAI;
5. SELECT \* FROM DARBUOTOJAI WHERE SKYRIAUSPAVADINIMAS = ‘Daug\_dirbantys’;
6. SELECT PAREIGOS FROM DARBUOTOJAI WHERE VARDAS = ‘Toma’;
7. SELECT \* FROM DARBUOTOJAI WHERE GIMIMOMETAI = ‘1960-05-04’;
8. SELECT VARDAS FROM DARBUOTOJAI WHERE PAVARDE = ‘Morkinis’;
9. SELECT VARDAS, PAVARDE FROM DARBUOTOJAI WHERE SKYRIAUSPAVADINIMAS =’Daug\_dirbantys’;
10. INSERT INTO DARBUOTOJAI VALUES(38807291235, ‘Regimantas’, ‘Sabonis’, ‘2013-01-21’,’1988-07-29’,’Testuotojas’,’Testavimo’,3);
11. INSERT INTO DARBUOTOJAI (ASMENSKODAS,VARDAS, PAVARDE, DIRBANUO, GIMIMOMETAI) VALUES (38101122335,’Petras’,’Petraitis’, ‘2009-10-30’,’1981-01-11’);
12. UPDATE DARBUOTOJAI SET PAREIGOS=’Programuotojas’, SKYRIAUSPAVADINIMAS=’Daug dirbantys’, PROJEKTONUMERIS=2 WHERE ASMENSKODAS=38101122335;
13. DELETE FROM DARBUOTOJAI WHERE ASMENSKODAS=38101122335;
14. INSERT INTO DARBUOTOJAI (PAVARDE, PAREIGOS) VALUES (‘Antanaitis’,’Programuotojas’),(‘Antanaitis’,’Programuotojas’);
15. UPDATE DARBUOTOJAI SET PAREIGOS= ‘Testuotojas’ WHERE PAVARDE= ‘Antanaitis’;
16. SELECT PAREIGOS FROM DARBUOTOJAI WHERE PAREIGOS = ‘Testuotojas; SELECT PAREIGOS FROM DARBUOTOJAI WHERE PAREIGOS = ‘Testuotoja’;

**2.2.2. Tema. Duomenų išrinkimas naudojant sąryšius (SQL select su join).**

*2 užduotis*. PANAUDOKITE TINKAMAS SĄLYGŲ IR GRUPAVIMO SQL KOMANDAS.

SQL komandos:

1. SELECT ASMENSKODAS, VARDAS, PAVARDE FROM DARBUOTOJAI WHERE GIMIMOMETAI=’1988-06-15’;
2. SELECT \* FROM DARBUOTOJAI WHERE GIMIMOMETAI < '1988-07-29';
3. SELECT DIRBANUO ,GIMIMOMETAI FROM DARBUOTOJAI WHERE DIRBANUO BETWEEN '2000-10-30' AND 2012-11-11';
4. SELECT VARDAS ,SKYRIAUSPAVADINIMAS ,PROJEKTONUMERIS FROM DARBUOTOJAI WHER PROJEKTONUMERIS IN (2,3);
5. SELECT VARDAS ,PAVARDE ,ASMENSKODAS FROM DARBUOTOJAI WHERE ASMENSKODAS LIKE '4%';
6. SELECT \* FROM DARBUOTOJAI WHERE GIMIMOMETAI LIKE '%-12';
7. SELECT \* FROM DARBUOTOJAI WHERE SKYRIAUSPAVADINIMAS LIKE '\_\_u%';
8. SELECT \* FROM DARBUOTOJAI WHERE PAREIGOS IS NULL;
9. SELECT VARDAS ,PAVARDE ,DIRBANUO ,PAREIGOS FROM DARBUOTOJAI WHERE DIRBANUO = '2010-08-01' AND PAREIGOS = 'Programuotojas';
10. SELECT VARDAS ,PAVARDE ,SKYRIAUSPAVADINIMAS ,PROJEKTONUMERIS FROM DARBUOTOJAI WHERE SKYRIAUSPAVADINIMAS = 'Mažai dirbantys' OR PROJEKTONUMERIS =1;
11. SELECT VARDAS FROM DARBUOTOJAI WHERE VARDAS NOT LIKE 'J%';
12. SELECT VARDAS ,DIRBANUO ,GIMIMOMETAI FROM DARBUOTOJAI WHERE DIRBANUO NOT BETWEEN '2009-10-30' AND '2012-11-11';
13. SELECT VARDAS ,PAVARDE ,GIMIMOMETAI FROM DARBUOTOJAI ORDER BY GIMIMOMETAI;
14. SELECT VARDAS ,PAVARDE ,GIMIMOMETAI FROM DARBUOTOJAI ORDER BY GIMIMOMETAI DESC;
15. SELECT MIN(PROJEKTONUMERIS), MAX(PROJEKTONUMERIS) FROM DARBUOTOJAI;
16. SELECT PROJEKTONUMERIS, COUNT(\*) FROM DARBUOTOJAI GROUP BY PROJEKTONUMERIS;
17. SELECT PROJEKTONUMERIS,PAREIGOS, COUNT(\*) FROM DARBUOTOJAI WHERE PAREIGOS LIKE 'Programuotoja%' GROUP BY PROJEKTONUMERIS, PAREIGOS;
18. SELECT PROJEKTONUMERIS,PAREIGOS, COUNT(\*) FROM DARBUOTOJAI WHERE PAREIGOS LIKE 'Programuotoja%' GROUP BY PROJEKTONUMERIS, PAREIGOS HAVING COUNT(\*)>2;

**Modulis „Programavimo aplinkos ir kūrimo proceso valdymas (JavaScript)“**

**1.** **Kompetencija. Naudoti tarnybinių stočių operacines sistemas.**

**1.1.** **Mokymosi rezultatas. Suprasti serverių veikimo principus ir panaudojimo galimybes.**

**1.1.1. Tema. Serveriai ir jų operacinės sistemos.**

*1 užduotis*. DEBESŲ KOMPIUTERIJA.

Debesų kompiuterija (angl. Cloud Computing) – paslaugos, kurioms pateikti reikalingas tik interneto ryšys. Trumpas debesų kompiuterijos sąvokos apibrėžimas - tai apjungti skaičiavimo ištekliai, teikiami internetu paslaugos pavidalu. Debesį sudaro programos, kurių nereikia įdiegti į kompiuterį. Google Apps suteikia galimybę paleisti verslui skirtas programas naudojant tik interneto naršyklę. Tai paslaugos Google Docs, Sheets, Forms ir pan.

Debesys – tai serveriai (kompiuteriai) pastoviai prijungti prie interneto į kuriuos vartotojai už tam tikrą mokestį (o kartais ir nemokamai) gali įsikelti savo norimus failus ir vėliau juos atsidaryti ar parsisiųsti iš serverio naudodami bet kurį įrenginį prijungtą prie interneto (kompiuterį, telefoną, planšetę) suvedę savo prisijungimo duomenis.

Virtualusis dedikuotas serveris (VDS) – tai toks serverio tipas, kai fizinis serveris „padalinamas“ į kelis virtualius serverius. Kiekvienam virtualiam serveriui priskiriama (rezervuojama) dalis fizinio serverio resursų: atminties, diskinės talpos, procesoriaus laiko ir pan. Kiekvienas virtualus dedikuotas serveris gali dirbti nepriklausomai nuo kitų tame pačiame fiziniame serveryje esančių virtualių serverių: vykdyti skirtingas operacines sistemas, teikti skirtingas paslaugas (angl. Services) ir pan. Virtualiųjų dedikuotų serverių (VDS) paslauga nuo tinklalapių hostingo (talpinimo) skiriasi daugiausiai tuo, jog pastaruoju atveju viename serveryje tuos pačius resursus naudoja šimtai skirtingų tinklalapių ir vienam jų netikėtai sulaukus didelio vartotojų antplūdžio yra stipriai paveikiamas kitų tinklalapių pasiekiamumas. Tuo tarpu atskirti ir rezervuoti procesoriaus, disko ir atminties resursai virtualiuose dedikuotuose serveriuose leidžia išvengti panašių atvejų. Galimybė kurti virtualius dedikuotus serverius atsirado sukūrus virtualizacijos technologiją.

*2 užduotis.* SERVERIO IŠTEKLIAI.

WordPress reikalavimai serveriui: https://wordpress.org/about/requirements:

1. PHP version 7.2 or greater.
2. MySQL version 5.6 or greater OR MariaDB version 10.0 or greater.
3. HTTPS support

Serverio parinkimas: https://www.iv.lt/dokumentai/profesionalus.pdf. Serveris gali būti Solo, Multi, Pro arba Individualus. Taip pat galima rinktis virtualų serverį su Linux arba Windows OS. Kaina priklausys nuo tinklapio apkrovimo, nes serverio resursai kainuoja priklausomai nuo pasirinkto plano.

*3 užduotis.* LINUX OS PARINKIMAS.

Teisingas atsakymas formuojamas https://distrochooser.de/en sistemos pagalba. Atsakymas turi būti pirmi tris Linux OS pavadinimai, kuriuos pasiūlė sistema.

**1.1.2. Tema. Serverio operacinės sistemos diegimas.**

*4 užduotis*. XUBUNTU OS DIEGIMAS VIRTUALIOJE APLINKOJE.

Užduotis atlikta pagal šias instrukcijas:

1. https://itsfoss.com/install-xfce-desktop-xubuntu
2. https://help.ubuntu.com/community/Installation?action=show&redirect=InstallingXubuntu
3. <https://virtualboxes.org/doc/installing-guest-additions-on-ubuntu>

**1.1.3. Tema. Bazinis serverio operacinės sistemos funkcionalumas.**

*5 užduotis*. XUBUNTU OS KONFIGŪRAVIMAS.

Darbas vykdomas pagal lektoriaus nurodymus. Darbas vykdomas Xubuntu OS aplinkoje. Konfigūruojame klaviatūrą: en, lt. Konfigūruojame XFce išvaizdą: pasirenkame norimą spalvinę temą.

**1.2.**  **Mokymosi rezultatas. Valdyti programinius paketus.**

**1.2.1. Tema. Programiniai paketai.**

*1 užduotis*. LINUX OS PAKETŲ ADMINISTRAVIMAS.

Sprendimas:

$ apt-cache search xcowsay

$ sudo apt-get install xcowsay

$ aptitude install figlet

$ xcowsay , $ figlet

https://launchpad.net/~webupd8team/+archive/ubuntu/java

$ sudo add-apt-repository ppa:webupd8team/java

$ sudo apt-get install eclipse && $ sudo apt-get install oracle-java8-installer

**1.3. Mokymosi rezultatas. Naudoti Web serverio programinę įrangą HTTP bylų viešinimui.**

**1.3.1. Tema. Web serverio aplinka operacinėje sistemoje.**

*1 užduotis*. WEB SERVERIO DIEGIMAS IR KONFIGŪRAVIMAS.

Sprendimas – veiksmai atlikti pagal šias instrukcijas:

1. https://tutorials.ubuntu.com/tutorial/install-and-configure-apache#0
2. https://help.ubuntu.com/lts/serverguide/httpd.html
3. https://help.ubuntu.com/lts/serverguide/web-servers.html
4. <https://howtoubuntu.org/how-to-install-lamp-on-ubuntu>

**2. Kompetencija.Taikyti aktualias programinės įrangos kūrimo metodikas.**



**2.1.** **Mokymosi rezultatas. Suprasti SCRUM proceso dalis ir komandos narių atsakomybes.**

**2.1.1. Tema. Scrum procesas.**

*1 užduotis.* KAS YRA SPRINTO ĮSIPAREIGOJIMŲ SAVININKAS?

(2) Visa komanda kartu

*2 užduotis*. DAUG ŽMONIŲ MANO, KAD PROGRAMAVIMAS POROMIS (PAIR PROGRAMMING) MAŽINA KLAIDŲ SKAIČIŲ IR PALENGVINA KODO PRIEŽIŪRĄ.

Programavimas poromis - kas tai?

(2) Du žmonės dalinasi viena darbo vieta (kompiuteriu) paprastai pasikeisdami paeiliui renka kodą ar atlieka veiksmus klaviatūra, kitas stebi, atkreipia dėmesį ir padeda pirmajam

*3 užduotis*. KOKS YRA REKOMENDUOJAMAS SCRUM KOMANDOS DYDIS?

(

4) 7 plius ar minus 2

*4 užduotis*. KOKIA KASDIENIO SCRUM SUSITIKIMO (DAILY SCRUM MEETING) TRUKMĖ (TIME-BOX)?

(3) 15 minučių

*5 užduotis*. KAS ATSAKO UŽ ĮRANKIŲ PASIRINKIMĄ IR KONFIGŪRAVIMĄ ORGANIZACIJOJE, KURI PRIPAŽĮSTA AGILE VERTYBES?

(1) Komandos, kurios turėtų susiderinti tarpusavyje

* 1. **Mokymosi rezultatas. Analizuoti pateiktus reikalavimus, nustatant programos atitikimą reikalavimams.**

**2.2.1. Tema. Programinės įrangos reikalavimų analizė.**

*1 užduotis*. KAS VYKSTA PROGRAMINĖS ĮRANGOS KŪRIMO CIKLO (angl. SOFTWARE DEVELOPMENT LIFECYCLE, SDLC) REIKALAVIMŲ SURINKIMO IR ANALIZĖS ETAPE?

(2) Klientas išsako savo lūkesčius projekte

*2 užduotis.* REIKALAVIMŲ ANALIZĖ NEAPIMA JŲ ATSEKAMUMO.

(2) Ne, apima.

(Explanation: Requirements traceability is concerned with documenting the life of a requirement and providing bi-directional traceability between various associated requirements, hence requirements must be traceable.)

*3 užduotis*. REIKALAVIMŲ ANALIZĖ YRA ITERATYVUS PROCESAS.

(1) Taip, yra.

(Explanation: Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized solutions can satisfy customer requirements.)

4 užduotis. KAIP ORGANIZUOJAMAS PRODUKTO DARBŲ SĄRAŠAS (angl. PRODUCT BACKLOG)?

(4) Svarbiausi darbai sąrašo pradžioje, mažiausiai svarbūs gale

(4) Kuo greičiau po to kai jos identifikuojamos, jei jos nekeičia

*5 užduotis*. KADA VYKSTANT SPRINTUI GALIMA PRIDĖTI NAUJAS SPRINTO UŽDUOTIS?

Darbo apimčių (scope change) siekiant užsibrėžtų sprinto tikslų (sprint goals)

*6 užduotis*. KAIP VERTINTUMĖTE TOKĮ VARTOTOJO PASAKOJIMĄ?

„Kaip pardavimo agentas, aš noriu turėti klientų paieškos galimybę kad galėčiau rasti savo klientus greitai ir lengvai“

(2) Gerai suformuluotas. Reiktų paaiškinimo, ką reiškia “greitai ir lengvai” kad būtų galima testuoti vartotojo sąsają.

**2.3. Mokymosi rezultatas. Naudoti projekto eigos valdymo principus.**

**2.3.1. Tema. Projekto eigos valdymas.**

*1 užduotis*. KOKS SPRINTO UŽDUOTIES DYDIS LAIKOMAS TINKAMU?

(1) Viena žmogaus diena arba mažiau, kad kiti komandos nariai galėtų lengvai pastebėti, jei užduoties įgyvendinimas užstrigo

*2 užduotis*. KAIP DAŽNAI TURI VYKTI PRODUKTO DARBŲ SĄRAŠO PERŽIŪRA (angl. BACKLOG GROOMING)?

(3) Kiekvieno sprinto metu

*3 užduotis.* KĄ SCRUM KOMANDA TURĖTŲ PADARYTI PIRMO SPRINTO METU? (galimi keli teisingi atsakymai)

(1) Testuoti produktą

ir

(4) Sukurti nedidelį gabaliuką veikiančio funkcionalumo (angl. potentially shippable functionality)

*4 užduotis.* ĮMONĖS VADOVAS PAPRAŠO KOMANDOS NARIO ATLIKTI DARBĄ, KURIS NEĮEINA Į VYKSTANČIO SPRINTO NUMATYTĄ APIMTĮ.

Ką tokiu atveju turėtų daryti komandos narys?

(3) Informuoti Produkto savininką, kad jis galėtų aptarti tai su įmonės vadovu

*5 užduotis.* KADA BAIGIAMAS SPRINTO VYKDYMAS?

(4) Kai baigiasi sprintui skirtas laikas (time-box expires)

**3.**  **Kompetencija. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą.**



**3.1. Mokymosi rezultatas. Diegti ir valdyti programavimo JavaScript kalba darbo aplinką.**

**3.1.1. Tema. JavaScript programavimo aplinka.**

*1 užduotis*. PAAIŠKINKITE KOKIE YRA NETBEANS IDE NAUDOJIMO PRIVALUMAI DIRBANT SU JAVASCRIPT FAILAIS?

NetBeans IDE naudojimo privalumai dirbant su JavaScript failais:

* Naujų failų kūrimas.
* Sintaksė ir semantinis paryškinimas.
* Kodo išplėtimas/sutraukimas.
* Naršymas tarp failų ir kodo elementų.
* Palengvintas kintamojo deklaravimas.
* Fono analizatorius - pateikia išsamius įspėjimus bei patarimus, kaip išspręsti galimas problemas.
* Skliaustų suderinimas.
* Automatinis formavimas – įtraukos, tarpai.
* JSON failų tipo palaikymas.
* PHP, JSP ir HTML failų įterpimas.

Papildomai galite paskaityti:

NetBeans IDE Documentation. Editing JavaScript [https://netbeans.org/kb/73/ide/JavaScript-editor.html](https://netbeans.org/kb/73/ide/javascript-editor.html)

*2 užduotis*. ĮDIEKITE NETBEANS IDE APLINKĄ.

NetBeans IDE aplinkos įdiegimas:

* Iš puslapio <https://netbeans.apache.org/download/index.html> atsisiunčiama naujausia NetBeans versija.
* Siuntimui pasibaigus, paleidžiama diegimo programa.
* Peržiūrima licencijos sutartis ir patvirtinamas sutikimas.
* Reikia spustelėti „Diegti“, kad pradėti NetBeans programos diegimą.
* Diegimu pasibaigus spausti „Baigti“.

Papildomai galite paskaityti:

NetBeans IDE 8.2 Installation Instructions. <https://netbeans.org/community/releases/82/install.html>

*3 užduotis*. SUKURKITE NAUJĄ NETBEANS IDE PROJEKTĄ JAVA PROGRAMAI.

Projekto pavadinimas – „FIRSTAPPLICATION“, pagrindinės klasės pavadinimas – „FIRSTAPPMAIN.FIRSTAPPLICATION“

Naujo NetBeans IDE projekto sukūrimo žingsniai:

* Paleisti „NetBeans IDE“.
* IDE pasirinkti File 🡪 New Project.
* Vedlyje „Naujas projektas“ išplėsti „Java“ kategoriją ir pasirinkti „Java“ programą (angl. „Java Application). Tada paspausti „Pirmyn“.
* Laukelyje „Projekto pavadinimas“ įvesti pavadinimą.
* Laukelyje „Kurti pagrindinę klasę“ (angl. Create Main Class) įvesti klasės pavadinimą.
* Paspausti „Baigti“.

Papildomai galite paskaityti:

NetBeans IDE Java Quick Start Tutorial. <https://netbeans.org/kb/docs/java/quickstart.html>

*4 užduotis*. KAIP ATSISIŲSTI KOMANDOS PROJEKTĄ IŠ „GITHUB“ Į NETBEANS IDE.

Projekto atsisiuntimas iš GitHub į NetBeans IDE.

* Nukopijuoti „GitHub“ saugyklos URL.
* Pagrindiniame meniu pasirinkti „Komanda“ (angl. „Team“), tada „Git“, tada „Klonuoti“ (angl. „Clone“).
* Įvesti saugyklos URL, naudotojo vardą ir slaptažodį ir spustelėti „Kitas“.
* Nuotolinių šakų puslapyje (angl. „Remote Branches page”) pasirinkti saugyklų šaką (-as), kuria (-ias) atsisiųsite į vietinę saugyklą, ir spustelėti „Pirmyn“.
* Lauke „Tėvinis katalogas“ (angl. „Parent Directory“) reikia nurodyti kelią į katalogą, skirtą klonuotai saugyklai kompiuterio diske.
* Lauke „Klono pavadinimas“ (angl. „Clone name“) reikia nurodyti aplanko pavadinimą į kurį bus atsiunčiamas projektas iš „Git“. (Pagal nutylėjimą Klono pavadinimas yra užpildytas faktiniu „Git“ saugyklos pavadinimu.)
* „Checkout Branch“ lauke pasirinkti šaką (angl. „branch“), iš kurios bus atsiunčiami projekto failai.
* Lauke „Nuotolinio vardo pavadinimas“ (angl. „Remote Name field“) reikia nurodyti klonuojamos saugyklos originalų pavadinimą.

Papildomai galite paskaityti:

Using Git Support in NetBeans IDE. Cloning a Git Repository. „<https://netbeans.org/kb/docs/ide/git.html>

### 3.2. Mokymosi rezultatas. Sekti programavimo darbų vykdymą naudojant komandinio darbų planavimo sistemas.

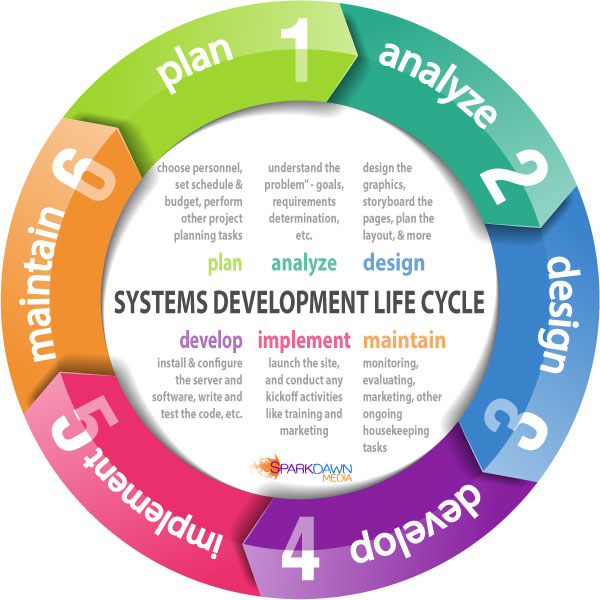
**3.2.1. Tema. Komandinio darbų planavimo sistemos.**

*1 užduotis.* UŽRAŠYKITE PAGRINDINIUS PROGRAMINĖS ĮRANGOS KŪRIMO ETAPUS.

Pagrindiniai programinės įrangos kūrimo etapai: planavimas (1), reikalavimų surinkimas ir analizė (2), dizaino ir architektūros projektavimas (3), programinės įrangos kūrimas (kodo rašymas) ir testavimas(4), integravimas ir testavimas (5), diegimas ir palaikymas (6).

Papildomai galite paskaityti:

*1.* Wikipedia (2019). *Systems development life cycle.* [žiūrėta 2019-05-27]. Prieiga per internetą<https://en.wikipedia.org/wiki/Systems_development_life_cycle.>



*2 užduotis*. APRAŠYKITE PROGRAMINĖS ĮRANGOS GYVAVIMO CIKLĄ.

Programinės įrangos gyvavimo ciklas:

1. **Planavimas:** šis etapas apima projektų ir produktų valdymo aspektus. Tai gali būti: išteklių paskirstymas, pajėgumų planavimas, projekto planavimas, sąnaudų įvertinimas ir t.t.. Planavimo etapo rezultatai apima: projekto planus, tvarkaraščius, sąnaudų įvertinimus ir pirkimų reikalavimus.

2. **Reikalavimų surinkimas ir analizė:** šiame etape projektuotojas kartu su užsakovu (klientu) bando sukurti aprašą, ką programinė įranga turėtų daryti, koks bus jos funkcionalumas. Reikalavimai formuluojami iš kliento perspektyvos. Dažnai iš pradžių jie formuluojami natūralia kalba užsakovui suprantamomis sąvokomis. Šio etapo vienas iš uždavinių – perkelti reikalavimus iš natūralios kalbos į labiau formalizuotą kalbą.

3. **Dizaino ir architektūros projektavimas:** šiame etape programinės įrangos architektai ir kūrėjai pradeda kurti programinę įrangą. Projektavimo procese naudojami nustatyti architektūros ir programinės įrangos kūrimo modeliai. Architektai kuria programinės įrangos architektūrą. Programuotojai kuria dizaino modelius, kad nuosekliai išspręstų algoritmines problemas. Šis etapas taip pat gali apimti greitą prototipų kūrimą Šio etapo rezultatai: dokumentai, kuriuose išvardyti projektui pasirinkti modeliai, komponentai, prototipai. Taip pat šiame etape gali būti aprašomi nefunkciniai reikalavimai

4. **Programinės įrangos kūrimas (kodo rašymas) ir testavimas:** šiame etape rašomas programos kodas, kuris po to testuojamas pagal ankstesnės veiklos metu apibrėžtus kriterijus. Šis etapas gali būti vykdomas pasirenkant reikiamą metodologiją: Waterfall, Agile ir pan.. Nepriklausomai nuo metodikos, kūrimo komandos turėtų kuo greičiau gaminti programinę įrangą. Testuojant parašytą kodą turi būti užtikrinta: kodo kokybė, funkciniai reikalavimai, saugumas.

5. **Integravimas ir testavimas:** realizavus pakankamai programinės įrangos komponentų, jie apjungiami, kaip aprašyta sistemos architektūros apraše ir atliekamas jų testavimas. Šiame etape jau galima atlikti ir tam tikrus priimtinumo testus su klientu (užsakovu). Atlikus testavimą su integruota sistema, ją galima perduoti užsakovui.

6. **Diegimas ir palaikymas:** šiame etape vykdomas programinės įrangos perdavimas ir diegimas. Veikla po programinės įrangos perdavimo vadinama palaikymu. Tai ilgiausiai trunkanti veikla. Jos metu taisomos klaidos, pastebėtos po programinės įrangos perdavimo.

Papildomai apie programinės įrangos gyvavimo ciklą galite paskaityti čia:

Wikipedia (2019). *Introduction to Software Engineering/Process/Life Cycle.* [žiūrėta 2019-05-27].Prieiga per internetą<https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Life_Cycle.>

*3 užduotis*. UŽRAŠYKITE TRIS KOMANDINIO DARBŲ PLANAVIMO SISTEMAS.

Komandinio darbų planavimo sistemos: Jira (Atlassian), Microsoft Project (Microsoft), Trello (Atlassian), HeySpace (Time Solutions TimeCamp Inc.), Taiga (Taiga) ir pan..

*4 užduotis*. APRAŠYKITE JIRA FUNKCIONALUMĄ IR PANAUDOJIMO GALIMYBES KURIANT IR TESTUOJANT PROGRAMINĘ ĮRANGĄ.

Jira siūlo tris programinės įrangos paketus: Jira Core, Jira Software, Jira Service Desk. Pagrindinis Jira funkcionalumas visiems programinės įrangos paketams:

1. lankstus projekto planavimas nuo reikalavimų iki konkrečios veiklos

2. pilnai konfigūruojamos Kanban ir Scrum lentos

3. galimybė įvertinti laiką, kai yra nustatyti prioritetai

4. ataskaitų teikimo funkcijos - nuo diagramų iki proceso greičio matavimų

5. pritaikomas darbo eigos procesas, atitinkantys jūsų poreikius

Jira gali būti naudojama organizuojant programinės įrangos kūrimo procesus. Užtikrina pilną kūrimo ir testavimo procesą.

Papildomą informaciją galite rasti čia:

1*.* Atlassian (2019). *Project management for non-project managers.* [žiūrėta 2019-05-27].Prieiga per internetą<https://www.atlassian.com/project-management.>

2. Atlassian (2019*). Jira Software best practices.* [žiūrėta 2019-05-27].Prieiga per internetą<https://www.atlassian.com/software/jira/guides/getting-started/best-practices.>

*5 užduotis.* APRAŠYKITE PAGRINDINIUS JIRA SISTEMOS APLINKOS ELEMENTUS PAPRASTAM VARTOTOJU

Aprašant aplinkos elementus naudokite Jira dokumentaciją:

Atlassian (2019). *Atlassian Documentation.* [žiūrėta 2019-05-27].Prieiga per internetą<https://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html.>

### 3.2.2. Tema. Programavimo darbų vykdymo sekimas.

*6 užduotis.* APRAŠYKITE KAIP YRA VYKDOMAS PROJEKTINIS DARBAS JIRA DARBŲ PLANAVIMO SISTEMOJE.

Projektinis darbas Jira darbų planavimo sistemoje yra vykdomas etapais:

1. sukuriamas projektas:

Atlassian (2019). *Create a project.* [žiūrėta 2019-05-27].Prieiga per internetą<https://confluence.atlassian.com/get-started-with-jira-core/create-a-project-917965385.html.>

2. sukuriamos projekto užduotys:

Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27].Prieiga per internetą<https://confluence.atlassian.com/jiracoreserver073/creating-issues-and-sub-tasks-861257329.html.>

3. pasirenkamas užduočių vykdymo modelis:

Idalko (2018). *A guide to Jira workflow best practices (with examples*)*.* [žiūrėta 2019-05-27].Prieiga per internetą<https://www.idalko.com/jira-workflow-best-practices.>

4. vykdomos projekto užduotys pagal pasirinktą modelį:

Atlassian (2019). *Working with boards.* [žiūrėta 2019-05-27].Prieiga per internetą<https://confluence.atlassian.com/jiracorecloud/working-with-boards-800712866.html.>

*7 užduotis*. APRAŠYKITE KAIP YRA PLANUOJAMOS, SUKURIAMOS, APRAŠOMOS UŽDUOTYS JIRA DARBŲ PLANAVIMO SISTEMOJE.

Aprašymui galite naudoti Jira dokumentaciją:

1. Atlassian (2019). *Working with issues.* [žiūrėta 2019-05-27]. Prieiga per internetą<https://confluence.atlassian.com/jirasoftwarecloud/working-with-issues-764478424.html.>

2. Atlassian (2019). *Working with issues in Jira Software*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.atlassian.com/agile/tutorials/issues.>

3. Atlassian (2019). *Issue types.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/adminjiracloud/issue-types-844500742.html>.

4. Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/creating-issues-and-sub-tasks-764478439.html.>

*8 užduotis*. APRAŠYKITE KAIP YRA PRISKIRIAMI DARBAI (UŽDUOTYS) ATSKIRIEMS VARTOTOJAMS.

Aprašymui galite naudoti Jira dokumentaciją:

1. Atlassian (2019). *Working with issues*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jiracorecloud/working-with-issues-765593800.html.>

2. Atlassian (2019). *Watch, share, and comment on issues*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/watch-share-and-comment-on-issues-962349057.html.>

*9 užduotis*. APRAŠYKITE KAIP YRA UŽBAIGIAMI DARBAI (UŽDUOTYS) JIRA DARBŲ PLANAVIMO SISTEMOJE.

Aprašymui galite naudoti Jira dokumentaciją:

1. Atlassian (2019). *Jira Sprints Tutorial.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.atlassian.com/agile/tutorials/sprints.>

2. Atlassian (2019*). Working with issues.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/working-with-issues-764478424.html.>

3. Atlassian (2019). *Creating issues and sub-tasks*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/creating-issues-and-sub-tasks-764478439.html.>

4. Atlassian (2019). *Deploying a release.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/deploying-a-release-764478183.html.>

*10 užduotis*. APRAŠYKITE KAIP YRA VYKDOMA PAIEŠKA JIRA DARBŲ PLANAVIMO SISTEMOJE.

Aprašymui galite naudoti Jira dokumentaciją:

1. Atlassian (2019). *Basic searching*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/basic-searching-764478306.html.>

2. Atlassian (2019). *Searching for issues*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/searching-for-issues-764478280.html.>

3. Atlassian (2019). *Advanced searching*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwareserver073/advanced-searching-861256227.html.>

4. Atlassian (2019*). Advanced searching*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/advanced-searching-764478330.html.>

# Modulis „Įvadas į darbo rinką“

# *TESTO ATSAKYMAI*

* + - 1. b)
      2. a)
      3. b)
      4. b)
      5. a)
      6. c)
      7. b)
      8. a)
      9. a)
      10. d)
      11. b)
      12. a)
      13. Blokuojantys ir neblokuojantys Node.js procesai:

Blokuojantys Node.js procesai yra tokie procesai, kurie vykdomi sinchroniškai ir dažniausiai yra susiję su I/O (input/output) operacijomis, ar priklausomi nuo nuotolinio serviso atsakymo. Šie procesai sustabdo visų kitų procesų vykdymą, nes yra priklausomi nuo išorinių, ne JavaScript operacijų. Tokių procesų reikia vengti Node.js programiniuose koduose, nes jie gali užblokuoti visos programos vykdymą.

Neblokuojantys Node.js procesai yra tokie procesai, kurie vykdomi asinchroniškai ir neblokuoja vienas kito vykdymo Node įvykių ciklo (angl. Event Loop) procese. Darbą su failais (I/O operacijas), bei kitus procesus galima įgyvendinti ir neblokuojančių procesų būdu.

Papildomai skaitykite:

Node.js Documentation. Overview of Blocking vs Non-Blocking. [žiūrėta 2019-07-29]. Prieiga per internetą <https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/>

* + - 1. Node.js web serverio programa įterpianti įrašus į duomenų bazę, bei atliekanti paiešką:

|  |
| --- |
| var http = require('http');  var url = require('url');  var MongoClient = require('mongodb').MongoClient;  var mongo\_url = "mongodb://localhost:27017/";  http.createServer(function (req, res) {  res.writeHead(200, {'Content-Type': 'text/html'});  var q = url.parse(req.url, true).query;  var txt = "";  if (q.action == "insert" && q.brand != "" && q.color != "") {  MongoClient.connect(mongo\_url, function(err, db) {  if (err) throw err;  var dbo = db.db("mydb1");  var myobj = { brand: q.brand, color: q.color};  dbo.collection("cars").insertOne(myobj, function(err, result) {  if (err) throw err;  txt = "1 car inserted into database";  res.end(txt);  db.close();  });  });  }  else if (q.action == "find" && q.color != "") {  MongoClient.connect(mongo\_url, function(err, db) {  if (err) throw err;  var dbo = db.db("mydb1");  //var query = { color: q.color };  var query = { color: "white" };  dbo.collection("cars").find(query).toArray(function(err, result) {  if (err) throw err;  var string\_results = result.map(item => {  return "<br>Brand:" + item['brand'] + "; Color:"+ item.color;  })  txt = "Results: " + string\_results;  res.end(txt);  console.log(result);  db.close();  });  });  }  else {  txt = "Bad request parameters.";  res.end(txt);  }  }).listen(8080); |

Papildomai skaitykite:

w3schools.com. Node.js MongoDB. [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.w3schools.com/nodejs/nodejs_mongodb.asp>

Node.js documentation. HTTP Server and File System. [žiūrėta 2019-07-29]. Prieiga per internetą <https://nodejs.org/api/http.html#http_response_writehead_statuscode_statusmessage_headers>, <https://nodejs.org/api/fs.html>

* + - 1. Node.js programa surandanti duomenų bazėje įrašus. Failas mongoose\_db\_find\_2.js:

|  |
| --- |
| var mongoose = require('mongoose');  mongoose.connect('mongodb://localhost:27017/cars', {useNewUrlParser: true});  var db = mongoose.connection;  db.on('error', console.error.bind(console, 'connection error:'));  db.once('open', function() {  var carSchema = new mongoose.Schema({  brand: String,  gas: Number  });    var Car = mongoose.model('Car', carSchema);    Car.find({ gas: { $gte: 20, $lte: 55 } }, function (err, cars) {  if (err) return console.error(err);  console.log("Found: " + cars);  db.close();  })  }); |

Komanda ir rezultatas:

|  |
| --- |
| C:\Users\Your Name\Node>node mongoose\_db\_find\_2.js  Found: { \_id: 5d3e256c9c342250f457deb3, brand: 'BMW', gas: 50 },{ \_id: 5d498fb9f37b799ef44dd33a, brand: 'Audi', gas: 45, } |

Papildomai skaitykite:

Mongoose Documentation. Query. [žiūrėta 2019-07-29]. Prieiga per internetą <https://mongoosejs.com/docs/api/query.html#query_Query-find>

* + - 1. c)
      2. b)
      3. c)
      4. a)
      5. b)

**Literatūros sąrašas**

1. Apache NetBeans 11.1 [žiūrėta 2019-07-29]. Prieiga per internetą <https://netbeans.apache.org/download/index.html>

2. Atlassian (2019). *Project management for non-project managers.* [žiūrėta 2019-05-27].Prieiga per internetą https://www.atlassian.com/project-management

3. Atlassian (2019*). Jira Software best practices.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://www.atlassian.com/software/jira/guides/getting-started/best-practices

4. Atlassian (2019). *Atlassian Documentation.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html

5. Atlassian (2019). *Create a project.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://confluence.atlassian.com/get-started-with-jira-core/create-a-project-917965385.html

6. Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://confluence.atlassian.com/jiracoreserver073/creating-issues-and-sub-tasks-861257329.html

7. Atlassian (2019). *Working with boards.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://confluence.atlassian.com/jiracorecloud/working-with-boards-800712866.html

8. Atlassian (2019). *Working with issues.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jirasoftwarecloud/working-with-issues-764478424.html

9. Atlassian (2019). *Working with issues in Jira Software*. [žiūrėta 2019-05-27]. Prieiga per internetą https://www.atlassian.com/agile/tutorials/issues

10. Atlassian (2019). *Issue types.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/adminjiracloud/issue-types-844500742.html>

11. Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwarecloud/creating-issues-and-sub-tasks-764478439.html 12. Atlassian (2019). *Watch, share, and comment on issues*. [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwarecloud/watch-share-and-comment-on-issues-962349057.html

13. Atlassian (2019). *Jira Sprints Tutorial.* [žiūrėta 2019-05-27]. Prieiga per internetą https://www.atlassian.com/agile/tutorials/sprints

14. Atlassian (2019). *Deploying a release.* [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwarecloud/deploying-a-release-764478183.html

15. Atlassian (2019). *Basic searching*. [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwarecloud/basic-searching-764478306.htm

16. Atlassian (2019). *Searching for issues*. [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwarecloud/searching-for-issues-764478280.html

17. Atlassian (2019). *Advanced searching*. [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwareserver073/advanced-searching-861256227.html

18. ECMAS*cript 6* [žiūrėta 2019-07-29]. Prieiga per internetą http://es6-features.org

19. Express.js documentation. *Middleware.* [žiūrėta 2019-07-29]. Prieiga per internetą https://expressjs.com/en/guide/using-middleware.html

20. Express.js documentation. *Routing*. [žiūrėta 2019-07-29]. Prieiga per internetą https://www.w3jar.com/express-js-routing/

21. Features of Node*.js.* [žiūrėta 2019-07-29]. Prieiga per internetą https://www.tutorialspoint.com/nodejs/nodejs\_introduction

22. Geeks For Geeks. *Difference between SQL and NoSQL*. [žiūrėta 2019-07-29]. Prieiga per internetą https://www.geeksforgeeks.org/difference-between-sql-and-nosql/

23. GbmbOrg (2019). *Data Units Conversion.* [žiūrėta 2019-05-27]. Prieiga per internetą https://www.gbmb.org.

24. Git (2019). *Downloads.* [žiūrėta 2019-05-27]. Prieiga per internetą https://git-scm.com/downloads

25. Git (2019). *Getting Started - Installing Git*. [žiūrėta 2019-05-27]. Prieiga per internetą https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

26. Git (2019). *Getting Started - First-Time Git Setup*. [žiūrėta 2019-05-27]. Prieiga per internetą https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

27. Git (2019). *Customizing Git - Git Configuration*. [žiūrėta 2019-05-27].

Prieiga per internetą https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration

28. Git (2019). *Git Basics - Getting a Git Repository*. [žiūrėta 2019-05-27]. Prieiga per internetą https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

29. Git Branching - Branches in a Nutshell. [žiūrėta 2019-05-27].

Prieiga per internetą https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell.

30. Git (2019). *Git format patch*. [žiūrėta 2019-05-27]. Prieiga per internetą https://git-scm.com/docs/git-format-patch

31. Idalko (2018). *A guide to Jira workflow best practices (with examples)*. [žiūrėta 2019-05-27]. Prieiga per internetą https://www.idalko.com/jira-workflow-best-practices

32. JavaScript [žiūrėta 2019-07-29]. Prieiga per internetą https://developer.mozilla.org/en-US/docs/Web/JavaScript

33. JavaScript [žiūrėta 2019-07-29]. Prieiga per internetą https://JavaScript.info/

34. JGraph Ltd (2019). *draw.io*. [žiūrėta 2019-05-27]. Prieiga per internetą https://www.draw.io.

35. Lietuvos statistikos departamentas (2019). *Ekonominės veiklos rūšių klasifikatorius*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://osp.stat.gov.lt/static/evrk2.htm>

36. Maskeliūnas, S. (2007). *Programų sistemų architektūra ir projektavimas.* Mokymo medžiaga parengta vykdant projektą „Programų sistemų magistrantūros įsteigimas“. [žiūrėta 2019-05-27]. Prieiga per internetą <https://klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf>

37. Modules documentation. *Embedded JavaScript templating and validation*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://ejs.co/#install>, <https://express-validator.github.io/docs/index.html>

38. MongoDB Download Center: *Community Server*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.mongodb.com/download-center/community>

39. Mongoose Documentation. *Getting Started.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://mongoosejs.com/docs/index.html>

40. Mongoose Documentation. *Getting Started. [*žiūrėta 2019-07-29]. Prieiga per internetą <https://mongoosejs.com/docs/index.html>

41. NetBeans IDE Documentation. *Editing JavaScript.* [žiūrėta 2019-07-29]. Prieiga per internetą [https://netbeans.org/kb/73/ide/JavaScript-editor.html](https://netbeans.org/kb/73/ide/javascript-editor.html)

42. NPM documentation. *Features of NPM.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/about-npm/> , <https://www.npmjs.com/package/feature>

43. NPM documentation. *CLI commands*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/cli/init> , <https://docs.npmjs.com/cli/install>

44. NPM documentation*. Packages and modules.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/about-packages-and-modules>

45. NPM documentation. *The semantic versioner for NPM.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/misc/semver>

46. NPM documentation*. NPM configuration files*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/files/package-lock.json>

47. NPM documentation. *Run arbitrary package script*s. [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/cli/run-script>

48. NPM documentation. CLI commands. [žiūrėta 2019-07-29]. Prieiga per internetą <https://docs.npmjs.com/cli/install>

49. NodeJS documentation. *Import required module*s. [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.tutorialspoint.com/nodejs/nodejs_first_application.htm>

50. Node.js documentation. *Timers in Node.js*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://nodejs.org/de/docs/guides/timers-in-node/>

51. Node.js documentation. *Standart input and output.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://nodejs.org/api/process.html#process_process_stderr>

52. Node.js documentation. *HTTP Server and File System*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://nodejs.org/api/http.html#http_response_writehead_statuscode_statusmessage_headers>, <https://nodejs.org/api/fs.html>

53. Oracle (2019). *Lambda Expressions*. [žiūrėta 2019-05-27]. Prieiga per internetą https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html

54. ReactJS [žiūrėta 2019-07-29]. Prieiga per internetą <https://reactjs.org/>

Stephen Rollins. *Live editing HTML and CSS with Chrome DevTools.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.lucidchart.com/techblog/2018/05/01/live-editing-html-css-chrome-devtools/>

55. Strong loop by IBM. *What Makes Node.js Faster Than Java*? [žiūrėta 2019-07-29]. Prieiga per internetą <https://strongloop.com/strongblog/node-js-is-faster-than-java/>

56. The Geek Stuff. *SQL vs NoSQL Database Differences Explained with few Example D*B. [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db>

57. Tools for web Developers*. Crome DevTools*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/chrome-devtools/?utm_source=dcc&utm_medium=redirect&utm_campaign=2018Q2>

58. Tools for web Developers. *Crome DevTools.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/chrome-devtools/?utm_source=dcc&utm_medium=redirect&utm_campaign=2018Q2>

59. Tools for web Developers. *Edit the DOM*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/chrome-devtools/inspect-styles/edit-dom>

60. Tools for web Developers*. Get Started With Viewing And Changing CSS.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/chrome-devtools/css/>

61. Tools for web Developers. *Performance Analysis Referenc*e. [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/reference#top_of_page>

62. Tools for web Developers. *Lighthouse.* [žiūrėta 2019-07-29]. Prieiga per internetą <https://developers.google.com/web/tools/lighthouse/>

63. Tools for web Developers .*Get Started with Debugging JavaScript in Chrome DevTools.* [žiūrėta 2019-07-29]. Prieiga per internetą [https://developers.google.com/web/tools/chrome-devtools/JavaScript/](https://developers.google.com/web/tools/chrome-devtools/javascript/)

64. Tools for web Developers*. Get Started with Debugging JavaScript in Chrome DevTool*s. [žiūrėta 2019-07-29]. Prieiga per internetą [https://developers.google.com/web/tools/chrome-devtools/JavaScript/](https://developers.google.com/web/tools/chrome-devtools/javascript/)

65. Tools for web Developers. *How To Pause Your Code With Breakpoints In Chrome DevTools.* [žiūrėta 2019-07-29]. Prieiga per internetą [https://developers.google.com/web/tools/chrome-devtools/JavaScript/breakpoints](https://developers.google.com/web/tools/chrome-devtools/javascript/breakpoints)

# 66. Tutorials Point (2019). *Number System Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą

# <https://www.tutorialspoint.com/computer_logical_organization/number_system_conversion.htm>

67. Using Git Support in NetBeans ID.E *Cloning a Git Repository*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://netbeans.org/kb/docs/ide/git.html>

68. Vilniaus universitetas (2019). *Apie skaičiavimo sistemas*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.mif.vu.lt/ljmm/rasmenys/skaiciavimo_sistemos.htm>.

69. Wikipedia (2019). *Introduction to Software Engineering/Process/Life Cycle.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://en.wikibooks.org/wiki/Introduction\_to\_Software\_Engineering/Process/Life\_Cycle

70. w3schools.com*. Node.js MongoDB.* [žiūrėta 2019-07-29]. Prieiga per internetą https://www.w3schools.com/nodejs/nodejs\_mongodb.asp

# 71. w3schools.com. *Node.js MongoDB Query*. [žiūrėta 2019-07-29]. Prieiga per internetą <https://www.w3schools.com/nodejs/nodejs_mongodb_query.asp>

72. Wikipedia (2019). *Numeral system*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Numeral_system>

73. Wikipedia (2019). *SOLID*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/SOLID

74. Wikipedia (2019). *Don’t repeat yourself*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Don’t\_repeat\_yourself

75. Wikipedia (2019). *Separation of Concerns*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Separation\_of\_concerns

76. Wikipedia (2019). *Code reuse*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Code\_reuse

77. Wikipedia (2019). *Data rate units*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Data-rate\_units

78. Wikipedia (2019). *Boolean Algebra*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Boolean\_algebra

79. Wikipedia (2019). *Projektavimo pavyzdys.* [žiūrėta 2019-05-27].Prieiga per internetąhttps://lt.wikipedia.org/wiki/Projektavimo\_pavyzdys

80. Wikipedia (2019). *Software design pattern.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://en.wikipedia.org/wiki/Software\_design\_pattern

81. Wikipedia (2019). *Units of information*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Units_of_information>

82. Wikipedia (2019). *List of numeral systems*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/List_of_numeral_systems>