

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

**Teorinių ir praktinių užduočių**

**mokinio sąsiuvinis**

Teorinių ir praktinių užduočių mokinio sąsiuvinis parengtas įgyvendinant Europos Sąjungos socialinio fondo ir Lietuvos Respublikos biudžeto lėšomis finansuojamą projektą „Kvalifikacijų formavimas ir modulinio profesinio mokymo sistemos kūrimas“ (projekto Nr. VP1-2.2-ŠMM-04-V-03-001).

Teorinių ir praktinių užduočių mokinio sąsiuvinis atnaujintas 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 Java programuotojo modulinės profesinio mokymo programos, 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:

Vaclav Zelenkevič

Giedrius Graževičius

# Modulis „Įvadas į profesiją“

## *TESTAS ĮSIVERTINTI GEBĖJIMAMS PRIEŠ PRADEDANT MOKYTIS*

1. Programuotojas tai:

a) asmuo, kuriantis, testuojantis, tobulinantis, prižiūrintis programas.

b) asmuo tvarkantis kompiuterius arba kompiuterinius tinklus.

c) asmuo atsakantis už kino ar televizijos filmo gamybą ir kūrimą.

2. Kuris iš pateiktų informacijos apibūdinimų yra tiksliausias?

a) Informacija – tai tik tokie duomenys, kurie kaupiami elektroninėse laikmenose.

b) Informacija – tai pranešimas, perduodamas telefonu.

c) Informacija – tai visuma žinių apie aplinkos ir visuomenės reiškinius, perduodamus visuomenės informavimo priemonėmis.

3. Kuriame atsakyme išvardyti tik išvesties įrenginiai?

a) Spausdintuvas, braižytuvas, garsintuvai, vaizduoklis.

b) Spausdintuvas, klaviatūra, skeneris, garsintuvai.

c) Pelė, braižytuvas, garsintuvai, vaizduoklis.

4. Kuri iš programų labiausiai tinka rengiant pranešimą konferencijai?

a) Skaičiuoklė.

b) Pateikčių programa.

c) Duomenų bazių valdymo sistema.

5. Su kuria natūraliojo skaičiaus x reikšme šio loginio reiškinio reikšmė bus FALSE?

(x > 18) or (x <> 19) and (x = 18)

a) 17

b) 19

c) 20

6. Kas būdinga atvirajai programinei įrangai?

a) Nemokamai galima naudotis tik kai kuriomis šios programinės įrangos funkcijomis.

b) Ši įranga yra nemokama. Jos pirminį tekstą (angl. source) galima keisti.

c) Šia įranga galima naudotis laisvai, bet negalima keisti jos pirminio teksto.

7. Žinome, kad pašto programose būna aplankas Juodraščiai (Drafts). Kam skirtas šis aplankas?

a) Nepageidautiniems laiškams (spam) laikyti.

b) Pradėtiems rašyti, bet dar neišsiųstiems laiškams laikyti.

c) Gautiems laiškams, kurių siuntėjų adresai nėra įtraukti į adresų knygą, laikyti.

8. Duota funkcija. Koks bus funkcijos kintamojo x rezultatas, kai bus įvesti skaitmenys: a = 3, b = 42, c = 5?

class Foo {

public void myFoo(int a, int b, int c) {

int x = (a + b) + (a % c) - b;

}

}

a) 4

b) 8

c) 6

9. Kas yra IT?

a) Kompiuterio sutrumpintas pavadinimas

b) Informacinės technologijos

c) Tinklų technologijos

10. Kas yra aparatinė įranga?

a) Kompiuterių technikos fizinių priemonių visuma

b) Automobiliui skirta įranga

c) Programinė įranga

11. Koks dešimtainės sistemos skaičius yra užrašytas: 10110111?

a) 138

b) 183

c) 191

12. Petras nuskynė 1 agurku daugiau nei Tomas. Jonas nuskynė 1 agurku daugiau nei Karolis. Petras su Tomu nuskynė 10 agurkų daugiau nei Jonas su Karoliu. Visi kartu nuskynė 60 agurkų. Kiek agurkų nuskynė Petras?

a) 20

b) 19

c) 18

13. Jeigu 12 nakties lyja lietus, ar galima tikėtis, kad po 72 valandų švies saulė?

a) Taip, nes po 72 valandų bus diena.

b) Ne, nes po 72 valandų vėl bus naktis.

c) Ne, nes po 72 valandų vėl lis.

14. IP adresas – tai:

a) į tinklą įjungto kompiuterio skaitmeninis vardas

b) kompiuterio procesoriaus identifikavimo numeris

c) internetu persiunčiamų duomenų protokolo vardas

15. Kokia galėjo būti kintamojo C reikšmė, jeigu, įvykdžius programos fragmentą, buvo išspausdintas skaičius 4?

A := 2; R := 9; C :=?;

while A < R do

if A > C

then R := R - 5

else A := A + 2;

Write(A);

a) 4

b) 17

c) 3

# Modulis „Informacinių sistemų projektavimas ir kūrimas (Java)“

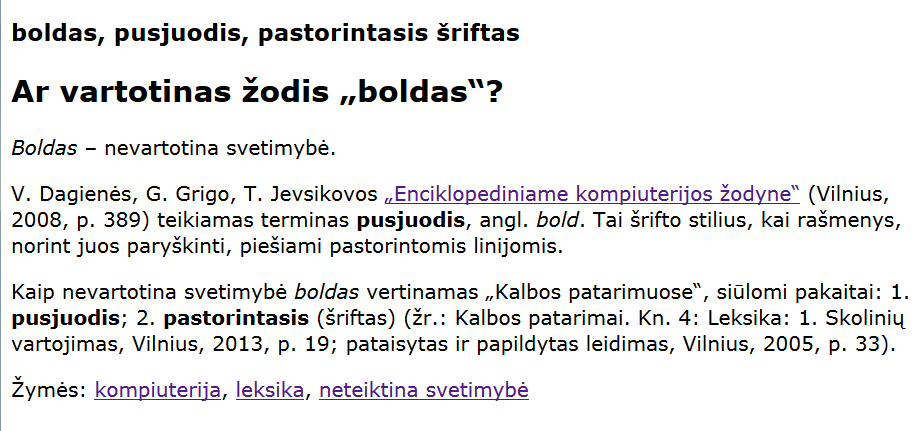
## Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.

### Mokymosi rezultatas. Pateikti internetinio puslapio turinį naudojant kompiuterinę žymėjimo kalbą.

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

UŽDUOTIES SĄLYGA: naudojant HTML antraščių, pastraipos, teksto formatavimo, nuorodų elementus, sukurti 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

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

UŽDUOTIES SĄLYGA: naudojant HTML antraščių, pastraipos, sąrašų ir kitus elementus, sukurti 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**

1. Paskutinė minutė
2. Poilsinės kelionės
3. 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:**

1. Vilnius – Barselona 14:00 – 16:30
2. 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**

1. Paskutinė minutė
2. Poilsinės kelionės
3. 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**

1. Paskutinė minutė
2. Poilsinės kelionės
3. 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, sukurti 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.

### 1.2. Mokymosi rezultatas. Apipavidalinti internetinį puslapį naudojant pakopinius stilių šablonus ir karkasus.

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

UŽDUOTIES SĄLYGA: naudojant skirtingas CSS selektorių kategorijas, aprašyti 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, aprašyti 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>

1. 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.

#### *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.

### 1.3. Mokymosi rezultatas. Programuoti vartotojo užduočių vykdymą naudojant JavaScript kalbą.

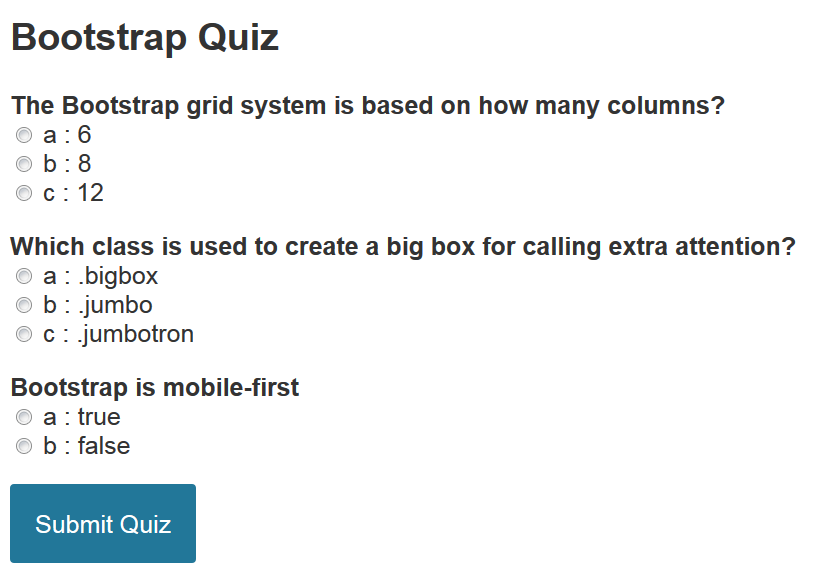
#### *1 užduotis.* SUKURKITE TESTĄ, TIKRINANTĮ IR PATEIKIANTĮ TEISINGUS ATSAKYMUS.

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:**



#### *2 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 ;

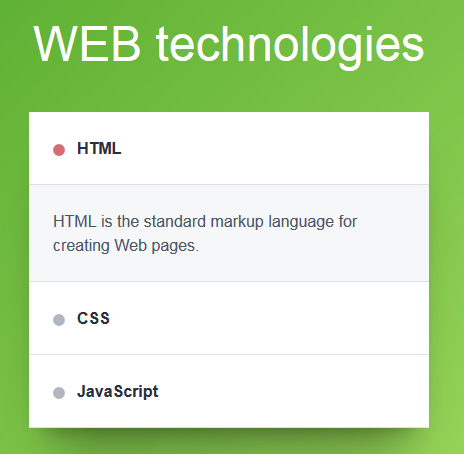
}

**Reikalavimai:**

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

**Pavyzdys:**





#### *3 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: 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;

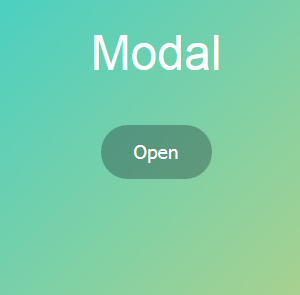
background: transparent;

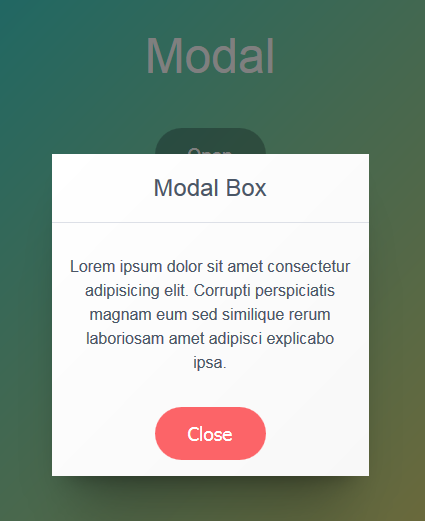
}

**Reikalavimai:**

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

**Pavyzdys:**





#### *4 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:

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>

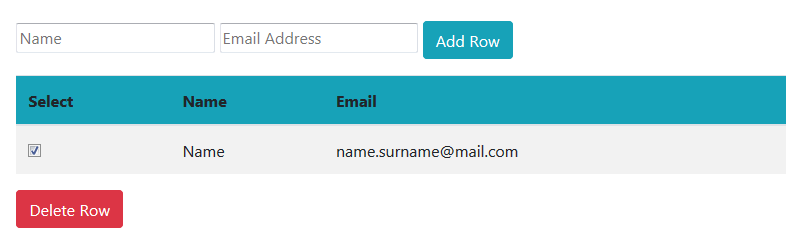
</body>

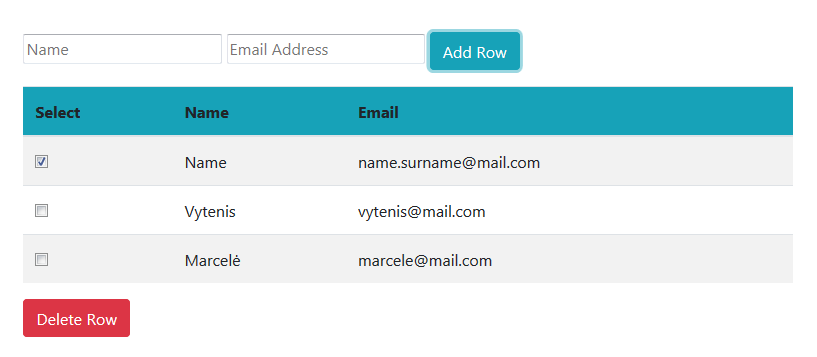
</html>

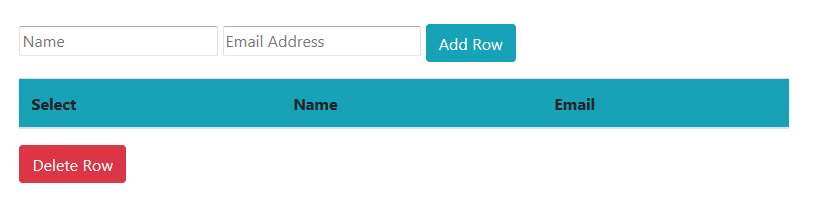
**Reikalavimai:**

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

**Pavyzdys:**







**2. Kompetencija. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.**

### 2.1. Mokymosi rezultatas. Suprasti skaičiavimo sistemas.

*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.

*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. Priimame, kad interneto greitaveika nesikeičia ir yra pastovi.

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

#### *15 užduotis.* JUMS REIKIA PERDUOTI 100GB DUOMENŲ.

#### Jūsų tinklo greitaveika yra 300mbps. Tinkle dirba kiti kompiuteriai. Įvertinkite esamą situaciją ir nuspręskite ar galima bus persiųsti pateiktą duomenų kiekį per 5 valandas.

### Mokymosi rezultatas. Taikyti algoritmų ir logikos mokslo pagrindus programuojant.

#### *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”)

#### *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 IEŠKO MAŽIAUSIĄ SKAIČIŲ AIBĖJE.

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

#### *13 užduotis*. ĮVERTINKITE KURIE IŠ ALGORITMŲ IR KOKIAIS ATVEJAIS YRA NAUDINGESNI IR GREIČIAU VEIKIA:

- algoritmai su rekursija;

- algoritmai be rekursijos.

### 

### Mokymosi rezultatas. Taikyti programinio kodo dizaino modelius programuojant.

#### *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.

#### 

#### *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).

#### 

#### *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.

#### 

#### *8 užduotis.* PARAŠYKITE JAVA PROGRAMĄ, KURI REALIZUOTŲ ADAPTER ŠABLONĄ.

#### 

#### *9 užduotis.* PARAŠYKITE JAVA PROGRAMĄ, KURI REALIZUOTŲ TEMPLATE METHOD ŠABLONĄ.

#### *10 užduotis.* APGALVOKITE IR NUSPRĘSKITE KOKĮ DIZAINO ŠABLONĄ GALIMA BŪTŲ PANAUDOTI APRAŠANT PAUKŠČIŲ KLASES.

### Mokymosi rezultatas. Naudoti informacinių sistemų kūrimo principus ir metodus programinės įrangos projektavime.

#### 

#### *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 JAVA 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 SPRING KARKASĄ SUKURKITE PAPRASTĄ WEB VARTOTOJŲ REGISTRACIJOS PROGRAMĄ.

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

#### 

#### *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.

#### *9 užduotis*. KOKIAIS ATVEJAIS FUNKCINIS PROGRAMAVIMAS YRA EFEKTYVESNIS UŽ OBJEKTINĮ?

## Kompetencija. Kurti tipinę programinę įrangą.

### 

### Mokymosi rezultatas. Naudoti Java programavimo kalbos įrankius ir sintaksę.

#### *1 užduotis*. WINDOWS, MAC ARBA LINUX OPERACINĖJE SISTEMOJE ĮSIDIEKITE JRE (JAVA RUNTIME ENVIRONMENT).

#### Sukonfigūruokite JRE. Patikrinkite ar operacinėje sistemoje veikia Java programų paleidimo įrankis.

#### *2 užduotis*. WINDOWS, MAC ARBA LINUX OPERACINĖJE SISTEMOJE ĮSIDIEKITE JDK (JAVA DEVELOPMENT KIT).

#### Sukonfigūruokite JDK . Patikrinkite ar operacinėje sistemoje veikia JAVA programų surinkimo įrankis.

#### 

#### *3* *užduotis*. NAUDOJANT TEKSTINĮ REDAKTORIŲ (PAVYZDŽIUI: NOTEPAD++) SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE HELLOWORLD.JAVA.

#### Klasė turi būti patalpinta į Java paketą com.oracle.javabasics. Klasės HelloWorld turinys:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World! (Text editor)");

}

}

#### 

#### *4 užduotis*. NAUDOJANT INTEGRUOTĄ KŪRIMO APLINKĄ (PAVYZDŽIUI: ECLIPSE) SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE HELLOWORLD.JAVA.

#### Klasė turi būti patalpinta į Java paketą com.oracle.javabasics. Klasės HelloWorld turinys:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World! (IDE)");

}

}

#### 

#### *5 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKIUS JAVAC IR JAVA, SURINKITE IR PALEISKITE *3 užduotyje* PARAŠYTĄ HELLOWORLD PROGRAMĄ.

#### 

#### Darbas vykdomas operacinės sistemos komandinėje eilutėje.

#### 

#### *6 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKIUS JDEPS IR JAVAP, PATIKRINKITE *3 užduotyje* PARAŠYTOS HELLOWORLD PROGRAMOS PRIKLAUSOMYBES IR PAŽIŪRĖKITE PROGRAMOS“BYTE” KODĄ.

#### Darbas vykdomas operacinės sistemos komandinėje eilutėje.

#### 

#### *7 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKĮ JAR SUKURKITE *3 užduotyje* PARAŠYTOS HELLOWORLD PROGRAMOS PALEIDŽIAMĄ ARCHYVĄ.

#### 

#### Darbas vykdomas operacinės sistemos komandinėje eilutėje.

#### 

#### *8 užduotis*. NAUDOJANT TEKSTINĮ REDAKTORIŲ SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE JAVACLASSPATH.JAVA.

#### Klasė turi būti patalpinta į Java paketą com.javabasics. Klasės JavaClasspath turinys:

package com.javabasics;

public class JavaClasspath {

public static void main(String[] args) {

System.out.println("Hello World! (Text editor)");

}

}

Surinkite klasę ir naudojant Java Classpath parametrą pabandykite paleisti surinktą klasę.

*9 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAELEMENTS.JAVA.

Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurtoje klasėje turi būti aprašytos dvi paprogramės (Methods) main ir printAll. Taip pat klasėje turi būti sukurti: lokali konstanta, objekto kintamasis, statinis klasės kintamasis, metodo kintamasis.

*10 užduotis*. PAPILDYKITE JAVAELEMENTS *9 užduotyje* PROGRAMĄ KODO KOMENTARAIS.

Turi būti panaudoti vienos ir kelių eilučių komentarai. komentarai turi būti prasmingi. papildomai parašykite JavaDoc tipo komentarą paprogramei printAll.

#### 

#### *11 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAEXPRESSIONSSTATEMENTSVARIABLES.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javabasics. darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurtoje klasėje turi būti aprašyti: du Java reiškiniai (Expressions), du sakiniai (Statements) ir du kintamieji (Variables). Pavadinimams turi būti panaudoti teisingi identifikatoriai (Identifiers).

#### 

#### *12 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAOPERATORS.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurtoje klasėje turi būti panaudoti Java operatoriai: +, -, /, \*, %, ++, --.

#### 

#### *13 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPRIMITIVESDATATYPES.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurtoje klasėje turi būti panaudoti Java primityvūs duomenų tipai: byte, short, int, long, float, double, char, boolean. Papildomai turi būti atspausdintos šių duomenų tipų reikšmes pagal nutylėjimą ir reikšmių diapazonas kiekvienam duomenų tipui.

#### 

#### *14 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVANONPRIMITIVESDATATYPES.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurtoje klasėje turi būti panaudoti Java objekto (rodykliniai) duomenų tipai: string, array. papildomai turi būti atspausdinta sukurto objekto nuoroda (Reference).

#### 

#### *15 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMETHODS.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje realizuokite paprastą skaičiuotuvą, kuris moka sudėti, atimti, padauginti ir padalinti skaičius. Kiekvienam skaičiuotuvo veiksmui (+, -, \*, /) parašykite atskirą paprogramę (method). Paprogramės turi mokėti priimti du skaičius ir grąžinti veiksmo rezultatą.

#### 

#### *16 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMETHODSOVERLOAD.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javabasics. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje realizuokite įvairių geometrinių figūrų (kvadratas, apskritimas ir t.t.) ploto skaičiuoklę. Kuriant paprogrames turi būti panaudotas paprogramių perkrovimo (Overload) mechanizmas.

#### *17 užduotis.* PARAŠYKITE FAKTORIALO SKAIČIAVIMO PROGRAMĄ, KURI NENAUDOJA REKURSIJOS.

#### *18 užduotis*. PARAŠYKITE FIBONAČIO SKAIČIŲ SEKOS SKAIČIAVIMO PROGRAMĄ, KURI NENAUDOJA REKURSIJOS.

### 

### Mokymosi rezultatas. Kurti nesudėtingą programinį kodą Java programavimo kalba.

#### 

#### *1 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPROPERTIESWRITE.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurta klasė turi įrašyti nustatymų (parametrų) sąrašą (Properties) į failą. nustatymai, kurie turi būti įrašyti į failą: database, dbuser, dbpassword. nustatymų failo pavadinimas: config.properties.

#### 

#### *2 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPROPERTIESREAD.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurta klasė turi nuskaityti nustatymų (parametrų) sąrašą (Properties) iš failo. Nuskaitomi nustatymai: database, dbuser, dbpassword. nustatymų failo pavadinimas: config.properties. Visi nustatymai turi būti atspausdinti komandinėje eilutėje.

#### 

#### *3 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVADATACONVERT.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti atlikti veiksmai su java duomenų konvertavimu: plėtimo (Widening), siaurinimo (Narrowing). Papildomai turi būti parašyta paprogramė, kuri nustato kintamojo objekto tipą naudojant operatorių instanceof.

#### 

#### *4 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAAUTOBOXINGUNBOXING.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti atlikti veiksmai su JAVA duomenų konvertavimu: Autoboxing, Unboxing.

#### 

#### *5 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASTREAMS.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti realizuotas vaisių pavadinimų įvedimas iš komandinė eilutės, vaisių parinkimas (filtravimas) pagal nustatytą požymį ir parinktų vaisių sąrašo spausdinimas komandinėje eilutėje. Vaisių sąrašas: bananas, ananasas, obuolys, kriaušė. požymis: vaisius turi raidę a

#### 

#### *6 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVABUFFERS.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi nuskaityti tekstinį failą ir jo turinį išvesti į komandinę eilutę. Turi būti panaudotos standartinės JAVA klasės: File, FileReader, BufferedReader, StringBuilder.

#### 

#### *7 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAIO.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti tekstinio failo kopijavimą. Turi būti panaudotos standartinės java klasės: FileInputStream ir FileOutputStream.

#### 

#### *8 užduotis*. SUKURKITE JAVA IŠVARDINIMŲ (ENUM) KLASĘ VARDU JAVAENUM.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi talpinti konstantų rinkinį: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY.

#### 

#### 9 *užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAWORKWITHENUM.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi atspausdinti komandinėje eilutėje *8 užduotyje* sukurtas konstantas klasėje Java.Enum.java. Papildomai visos konstantos turi bųti įrašytos į masyvą ir sunumeruotos. Masyvas turi būti atspausdfintas komandinėje eilutėje.

#### 

#### *10 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASQUAREAREA.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti kvadrato ploto skaičiavimą.

#### 

#### *11 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAWORKWITHPACKAGE.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javapackage. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Sukurta klasė turi iškviesti *10 užduotyje* sukurtą klasę ir paskaičiuoti kvadrato plotą.

#### 

#### *12 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAARRAYLIST.JAVA.

#### 

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti realizuotas ArrayList dinaminis sąrašas. Sąrašo dydis 20 elementų. Sąrašą užpildyti atsitiktiniais skaičiais (nuo 0 iki 100) ir surūšiuoti. Atspausdinti sąrašą komandinėje eilutėje.

#### 

#### *13 užduotis.* SUKURKITE JAVA KLASĘ VARDU JAVALINKEDLIST.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti realizuotas LinkedList dinaminis sąrašas. Sąrašo dydis 50 elementų. Sąrašą užpildyti atsitiktiniais skaičiais (nuo 0 iki 1000) ir sumaišyti. Atspausdinti sąrašą komandinėje eilutėje.

#### 

#### *14 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMAP.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba ide. Klasėje turi būti realizuotas Map dinaminis sąrašas. Sąrašo dydis 100 elementų. Sąrašą užpildyti atsitiktiniais skaičiais (nuo 0 iki 1000). Atspausdinti sąrašą komandinėje eilutėje. Papildomai turi patikrinta ar sąrašas turi skaičių 100.

#### 

#### *15 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASET.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javasimple. Darbą galite atlikti tekstiniame redaktoriuje arba ide. Klasėje turi būti realizuotas Set dinaminis sąrašas. Sąrašo dydis 40 elementų. Sąrašą užpildyti atsitiktiniais skaičiais (nuo 0 iki 20). Visi sąrašo elementai turi būti unikalūs. Atspausdinti sąrašą komandinėje eilutėje.

#### 

#### *16* *užduotis*. IŠNAGRINĖKITE PATEIKTĄ KODĄ IR SURASKITE VISAS KLAIDAS, KURIAS PADARĖ PROGRAMUOTOJAS, KURIS NESINAUDOJA JAVA KODAVIMO STANDARTAIS.

package javasimple;

public class javastandards {

public static void main(String[] args) {

String Text = “We are comparing content”;

StringBuffer stringBuffer = new StringBuffer(“We are comparing content”);

if (Text.contentEquals(stringBuffer)) {

System.out.println(“The content is equal”);

} else {

System.out.println(“The content is not equal”);

}

}

}

#### 

#### *17 užduotis*. NAUDOJANT JAVADOC STANDARTĄ PARAŠYKITE PRASMINGUS KOMENTARUS PATEIKTAM JAVA KODUI.

package javasimple;

import java.io.IOException;

public class AddNum {

public int addNum(int numA, int numB) {

return numA + numB;

}

public static void main(String args[]) throws IOException {

AddNum obj = new AddNum();

int sum = obj.addNum(10, 20);

System.out.println("Sum of 10 and 20 is :" + sum);

}

}

#### 

#### *18 užduotis*. YRA PATEIKTAS SĄRAŠAS: {1.25, 3.45, 0.58, 1.32, -2.58, 10.79, -1.00, 45.91}:

- atspausdinkite kiekvieną sąrašo skaičių naudojant ciklą;

- raskite sąrašo narių vidurkį;

- raskite sąrašo narių sumą;

- raskite mažiausią sąrašo narį;

- rasti didžiausią sąrašo narį.

**3.3. Mokymosi rezultatas. Taikyti objektinio programavimo principus programuojant***.*

*1 užduotis*. SUKURKITE JAVA KLASĘ VARDU CAR.JAVA.

Klasė turi būti patalpinta į paketą vardu javaobjects. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti automobilio brėžinį / šabloną. Automobilio parametrai: modelis, spalva, greitis, degalų tipas. Automobilis turi mokėti važiuoti.

*2 užduotis*. SUKURKITE JAVA KLASĘ VARDU MAIN.JAVA.

Klasė turi būti patalpinta į paketą vardu javaobjects. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje turi būti sukurti du automobiliai Toyota ir Mercedes. Automobilių objektai turi būti sukurti naudojant *1 užduotyje* sukurtą Car.java brėžinį. Patikrinti ar automobiliai moka važiuoti ir kuris automobilis yra greitesnis.

*3 užduotis*. PAKEISKITE *1 užduotyje* SUKURTĄ CAR.JAVA KLASĘ TAIP, KAD BŪTŲ PASLĖPTA PAPROGRAMĖ CANDRIVE().

Taip pat pakeiskite parametro „spalva” savybę taip, kad jo negalima būtų keisti.

*4 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAENCAPSULATION.JAVA.

Klasė turi būti patalpinta į paketą vardu javaobjects. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi parodyti Java inkapsuliacijos principą.

*5 užduotis*. SUKURKITE JAVA KLASĘ VARDU TOYOTA.JAVA.

Klasė turi būti patalpinta į paketą vardu javaobjects. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi paveldėti visas savybės iš klasės Car.java (*1 užduotis)*. Papildomai mūsų automobilis turi turėti naują parametrą: variklio tipas.

*6 užduotis*. SUKURKITE JAVA KLASĘ VARDU MERCEDES.JAVA.

Klasė turi būti patalpinta į paketą vardu javaobjects. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi paveldėti visas savybės iš klasės Car.java (*1 užduotis)*. Papildomai mūsų automobilis turi turėti naują parametrą: sėdimų vietų skaičius.

*7 užduotis*. PAKEISKITE *5 užduotyje* SUKURTĄ TOYOTA.JAVA KLASĘ TAIP, KAD AUTOMOBILIS GALĖTŲ NE TIK VAŽIUOTI BET IR SKRISTI.

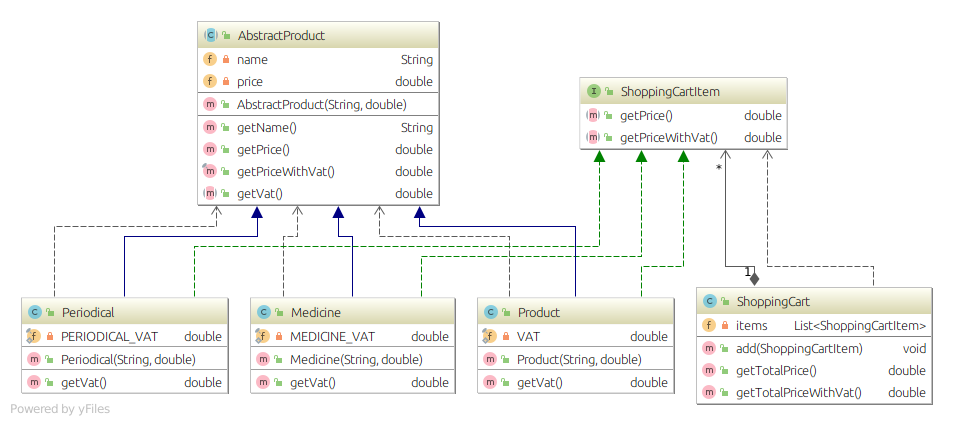
*8 užduotis*. SUKURKITE KLASES DEMOPOLYMORPHISM.JAVA IR TEST.JAVA.

#### Klasė DemoPolymorphism.java turi realizuoti statinio polimorfizmo principą. Klasė Test.java turi patikrinti šią savybę.

#### 

#### *9 užduotis.* PAGAL PATEIKTĄ PARDUOTUVĖS KLASIŲ DIAGRAMĄ SUKURKITE REIKALINGAS KLASES IR INTERFEISUS SU REIKALINGAIS KLASIŲ LAUKAIS, PAPROGRAMĖMIS.

#### Turi būti realizuota parduotuvė su prekių grupėmis: Product, Medicine, Periodical ir prekių krepšeliu, kuris moka skaičiuoti prekių kainą su ir be PVM. Prekių grupės privalo turėti nustatytą savo prekių kategorijai PVM.



#### *10 užduotis*. PAGAL PATEIKTAS KLASES NUBRAIŽYKITE UML JAVA KLASIŲ DIAGRAMĄ.

package javaobjects;

public class Shape {

private String color;

public Shape(String color) {

this.color = color;

}

@Override

public String toString() {

return "Shape[color=" + color + "]";

}

public double getArea() {

System.err.println("Shape unknown! Cannot compute area!");

return 0;

}

}

package javaobjects;

public class Rectangle extends Shape{

private int length;

private int width;

public Rectangle(String color, int length, int width) {

super(color);

this.length = length;

this.width = width;

}

@Override

public String toString() {

return "Rectangle[length=" + length + ",width=" + width + "," + super.toString() + "]";

}

@Override

public double getArea() {

return length \* width;

}

}

package javaobjects;

public class Triangle extends Shape{

private int base;

private int height;

public Triangle(String color, int base, int height) {

super(color);

this.base = base;

this.height = height;

}

@Override

public String toString() {

return "Triangle[base=" + base + ",height=" + height + "," + super.toString() + "]";

}

@Override

public double getArea() {

return 0.5 \* base \* height;

}

}

#### 

#### *11 užduotis*. TRIMATĖJE KOORDINAČIŲ SISTEMOJE REIKIA SAUGOTI TAŠKŲ KOORDINATES.

#### Taškai be koordinačių taip pat turi spalvą ir svorį. Sukurkite reikiamą klasę koordinatės taškui point.

#### *12 užduotis*. NORINT PATIKRINTI ŽMOGAUS KŪNO MASĖS INDEKSĄ, REIKALINGAS SKAIČIUOTUVAS. SUKURKITE PROJEKTĄ BMI, SKIRTĄ ŠIAI UŽDUOČIAI.

#### Turi būti galimybė atlikti kūno masės skaičiavimo veiksmus nekuriant papildomų objektų. Projekte sukurti metodus: bmiCount, kuris skaičiuotų žmogaus kūno masės indeksą, readFromConsole, kuris leistų įvesti žmogaus aukštį ir svorį iš komandinės eilutės ir printbmi, kuris atspausdintų žmogaus BMI komandinėje eilutėje ir įrašytų šią informaciją į failą vardu bmi.txt. Klasės metodų veikimą pademonstruokite Main metode.

### 

### 3.4. Mokymosi rezultatas. Testuoti programinę įrangą naudojant su Java programavimo kalba suderinamus testavimo įrankius ir metodus.

#### 

#### *1 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ VARDU JAVATESTING.

#### 

#### Projekte sukurkite Java paketą javajunit. Jame sukurkite klasę JavaForTesting.java. Klasė turi mokėti priimti vardą ir jį grąžinti. Prie projekto pridėkite junit testavimo karkasą. Naudokite 5 JUnit versiją.

#### *2 užduotis*. PARAŠYKITE *1 užduotyje* SUKURTAM PROJEKTUI VIENĄ PAPRASTĄ TESTĄ, KURIS PATIKRINA AR KLASĖ JAVAFORTESTING VEIKIA TEISINGAI.

#### 

#### *3 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ VARDU TESTING.

#### Jame sukurkite klasę Calculator.java. Klasė turi būti patalpinta į paketą vardu javatesting. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasėje realizuokite paprasto skaičiuotuvo funkcionalumą (+, -, \*, /), skaičiuotuvas turi mokėti atlikti paprastas aritmetines operacijas su skaičiais. Prie projekto prijunkite JUnit testavimo biblioteką. Parašykite testus, kurie patikrintų ar skaičiuotuvas dirba be klaidų.

#### 

#### *4 užduotis.* 3 KOMPETENCIJOS 3.3. MOKYMOSI REZULTATO *9 užduotyje* REALIZUOTAI PARDUOTUVEI SU PREKIŲ KREPŠELIU PARAŠYKITE JUNIT TESTUS, KURIE PATIKRINTŲ PRODUKTŲ GRUPIŲ IR PREKIŲ KREPŠELIO FUNKCIONALUMĄ.

#### 

#### *5 užduotis*. SUKURKITE JAVA KLASĘ VARDU SIMPLEEXCEPTION.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javaexceptions. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi generuoti sveikų skaičių masyvą, užpildomą atsitiktinai. Naudojamas paprastas masyvas, kurio ilgis 10 elementų. Į masyvą bandome įrašyti 20 atsitiktinių skaičių. Jei pildant masyvą įvyks klaida, ją reikia apdoroti ir pranešti vartotojui, kad masyvas yra per mažas. Masyvas turi būti užpildytas.

#### *6 užduotis*. SUKURKITE JAVA KLASĘ VARDU FILEREADERS.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javaexceptions. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti informacijos skaitymą iš tekstinio failo. Visos klaidos ir išimtys skaitant informaciją iš failo turi būti apdorotos.

#### 

#### *7 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVALOGGER.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javalogers. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti Fibonacci skaičių mažesnių už 1000 spausdinimą. Klasėje turi būti panaudotas standartinis Java žurnalizavimo įrankis Logger. Į komandinę eilutę turi būti išvesti prasmingi pranešimai vartotojui apie programos veikimą.

#### 

#### *8 užduotis.* SUKURKITE JAVA KLASĘ VARDU JAVAFILELOGGER.JAVA.

#### Klasė turi būti patalpinta į paketą vardu javalogers. Darbą galite atlikti tekstiniame redaktoriuje arba IDE. Klasė turi realizuoti daugybos lentelės (iki 10) spausdinimą komandinėje eilutėje. Klasėje turi būti panaudotas nestandartinis Java žurnalizavimo įrankis log4j. Prasmingi pranešimai apie programos veikimą turi būti įrašyti į failą javafilelogger.log

#### 

#### *9 užduotis*. IŠBANDYKITE PATEIKTĄ JAVA PROGRAMINĮ KODĄ.

#### Naudojant standartinius IDE kodo derinimo įrankius raskite visas kode paslėptas klaidas:

public class JavaDebug {

public static void main(String[] args) {

System.out.println("Square perimeter \n and area counter.");

Scanner input = new Scanner(System.in);

input.close();

System.out.print("Input square side, a = ");

double a = input.nextDouble();

if (a >= 0) {

double area = Math.sqrt(a);

double perimeter = 6 \* 2a;

System.out.println("Square area A = " + perimeter);

System.out.println("Square perimeter P = " + area);

} else if (a = 0) {

System.out.println("Error! The length of the side is to be a positive number.");

}

}

}

#### *12 užduotis.* NORINT PATIKRINTI ŽMOGAUS KŪNO MASĖS INDEKSĄ, REIKALINGAS SKAIČIUOTUVAS.

#### Sukurkite projektą BMI, skirtą šiai užduočiai. Turi būti galimybė atlikti kūno masės skaičiavimo veiksmus nekuriant papildomų objektų. Projekte sukurti metodus: bmiCount, kuris skaičiuotų žmogaus kūno masės indeksą, readFromConsole, kuris leistų įvesti žmogaus aukštį ir svorį iš komandinės eilutės ir printbmi, kuris atspausdintų žmogaus BMI komandinėje eilutėje ir įrašytų šią informaciją į failą vardu bmi.txt. Klasės metodų veikimą pademonstruokite Main metode

## Kompetencija. Analizuoti skirtingų tipų reikalavimus, apibūdinančius kompiuterinę programą.

### 

### Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo (angl. user story) reikalavimų programinei įrangai formatą.

#### *1 užduotis*. UŽRAŠYKITE VARTOTOJO PASAKOJIMO (USER STORY) FORMATĄ.

#### Pateikite pavyzdžių.

#### *2 užduotis*. YRA PATEIKTI FUNKCINIAI PROGRAMINĖS ĮRANGOS REIKALAVIMAI.

#### Pabandykite suformuoti 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 :

a) kurti vartotojų grupes (žiūrėti FR3), priskirti joms vartotojus

b) kurti dokumentų tipus ir priskirti grupes galinčias kurti dokumentus bei juos priimti

c) registruoti naujus vartotojus sistemoje

d) keisti esamų vartotojų slaptažodžius

#### *3 užduotis*. APRAŠYKITE GERŲ VARTOTOJO PASAKOJIMO FORMATĄ (INVEST).

#### *4* *užduotis*. APRAŠYKITE REIKALAVIMŲ PERŽIŪROS ŽINGSNIUS.

#### *5 užduotis.* APRAŠYKITE REIKALAVIMŲ PARUOŠIMO SPRINTUI KRITERIJUS (DEFINITION OF READY).

#### Pateikite pavyzdžių.

#### *6 užduotis*. APRAŠYKITE REIKALAVIMŲ BAIGTUMO APIBRĖŽIMĄ.

#### Pateikite pavyzdžių.

#### *7 užduotis*. APRAŠYKITE PAGRINDINIUS SCRUM PROCESO ĮVYKIUS.

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

#### *1 užduotis.* APRAŠYKITE KAS YRA FUNKCINIAI REIKALAVIMAI.

#### Pateikite pavyzdžių.

#### *2 užduotis.* PATEIKTAM PAVYZDŽIUI *1 užduotyje* PARAŠYKITE KAIP BUS NUSTATOMA AR FUNKCINIS REIKALAVIMAS BUVO ĮVYKDYTAS AR NE.

#### *3 užduotis*. APRAŠYKITE KAS YRA NEFUNKCINIAI REIKALAVIMAI.

#### Pateikite pavyzdžių.

#### *4 užduotis.* PATEIKTAM PAVYZDŽIUI *1 užduotyje* PARAŠYKITE KAIP BUS NUSTATOMA AR NEFUNKCINIS REIKALAVIMAS BUVO ĮVYKDYTAS AR NE.

# 

# Modulis „Nesudėtingų duomenų bazių projektavimas ir kūrimas (Java)“

## 1. Kompetencija. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes.

### 

### Mokymosi rezultatas. Pateikti reliacinės duomenų bazės schemą.

#### *1 užduotis*. SUPROJEKTUOKITE 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, vardas, 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*. SUPROJEKTUOKITE 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.

1. Poetu gali būti visi norintieji, ne vien tik užfiksavę savo poeziją sistemoje arba jau sukūrė poezijos kūrinį.
2. Poezijos kūrinius galima pateikti internetu, el. paštu arba popieriuje.
3. Visi sistemoje užfiksuoti poezijos kūriniai parašyti poetų, kurių duomenys jau užfiksuoti sistemoje. Nepateikęs visos reikalaujamos informacijos, poetas negalės pateikti ir išsaugoti savo poezijos sistemoje.
4. Leidinį gali sudaryti ir vienas eilėraštis, ir poezijos rinkinys, ir literatūros kritikos kūrinys.
5. Leidiniai parduodami klientams, apie kuriuos sistemoje yra informacijos. Anonimiški pirkėjai neaptarnaujami.
6. 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.
7. 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?

### Mokymosi rezultatas. Pateikti nereliacinės (NoSQL) duomenų bazės schemą.

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:

1. Gauti parduotuvės informaciją
2. Gauti konkretaus tipo prekės, konkrečioje parduotuvėje informaciją
3. 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.

#### *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.

#### *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 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);

*2 užduotis.* ATLIKITE UŽKLAUSAS H2 APLINKOJE IMPORTUOTOJE DB (NAUDOJAMOS PRIEMONĖS/APLINKOS - H2 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.

### Mokymosi rezultatas. Kurti duomenis duomenų bazėje valdančią programinę įrangą.

#### *1 užduotis*. PANAUDOKITE TINKAMAS SĄLYGŲ IR GRUPAVIMO SQL KOMANDAS

Duota:

1. 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 H2 aplinkoje importuotoje DB (Naudojamos priemonės/aplinkos - H2 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 (Java)“

## 

## 1. Kompetencija. Naudoti tarnybinių stočių operacines sistemas.

### Mokymosi rezultatas. Administruoti skaitmenines bylas bei tarnybinės stoties vartotojus naudojant tos tarnybinės stoties operacinę sistemą.

#### *1 užduotis.* DEBESŲ KOMPIUTERIJA.

UŽDUOTIES SĄLYGA: surasti informaciją apie debesų kompiuteriją, atsakyti 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 surasti 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 parinkti sau tinkamą Linux OS.

Naudojamos priemonės (aplinkos): interneto 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.

#### *4 užduotis.* XUBUNTU OS DIEGIMAS VIRTUALIOJE APLINKOJE.

UŽDUOTIES SĄLYGA: naudojant VirtualBox įrankį ir Xubuntu OS iso atvaizdą virtualizuoti 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).

#### *5 užduotis.* XUBUNTU OS KONFIGŪRAVIMAS.

UŽDUOTIES SĄLYGA: naudojant Xubuntu grafinius konfigūracijos įrankius sukonfigūruoti 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.

### 

### Mokymosi rezultatas. Valdyti tarnybinę stotį naudojant jos komandinės eilutės sąsają ir jos pagrindines komandas.

#### *1 užduotis.* DARBAS LINUX OS KOMANDŲ EILUTĖJE.

UŽDUOTIES SĄLYGA: naudojant komandų eilutę atlikti failų ir katalogų kūrimo, kopijavimo, perkėlimo, pašalinimo veiksmus.

Naudojamos priemonės (aplinkos): Linux OS komandų eilutė.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Savo namų kataloge sukurkite katalogą Bandymas.

2. Kataloge Bandymas sukurkite failus: f1.txt, f2.txt, f3.txt, f4.txt, ff1.txt, ff2.txt, ff3.txt, ff4.txt.

3. Pažiūrėkite katalogo Bandymas turinį.

4. Savo namų kataloge sukurkite katalogą Tikslas.

5. Nukopijuokite katalogo Bandymas failus f1.txt, f3.txt, ff2.txt, ff4.txt į katalogą Tikslas.

6. Pažiūrėkite katalogo Tikslas turinį.

7. Katalogo Tikslas failų vardus pakeiskite iš f\*.txt į file\*.txt.

8. Perkėlkite pervadintus failus į katalogą Bandymas.

9. Nukopijuokite failą passwd iš katalogo /etc į katalogą Bandymas.

10. Išveskite failo passwd turinį į ekraną.

11. Suraskite faile passwd žodį root. Rezultatą išveskite į ekraną.

12. Suskaičiuokite kiek yra simbolių faile passwd.

13. Išveskite į ekraną failo passwd 5 pirmas ir paskutines failo eilutes.

14. Kataloge Tikslas sukurkite failo passwd simbolinę nuorodą.

15. Pažiūrėkite katalogo Tikslas turinį.

16. Kataloge Bandymas ištrinkite failą passwd.

17. Pažiūrėkite katalogo Tikslas turinį.

18. Ištrinkite katalogą Bandymas su visu turiniu.

#### *2 užduotis.* SUKURKITE BASH SCENARIJŲ.

UŽDUOTIES SĄLYGA: naudojant nano tekstinį redaktorių ir komandų eilutę parašyti paprastą scenarijų.

Naudojamos priemonės (aplinkos): Linux OS komandų eilutė, nano tekstinis redaktorius.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato: sukurkite bash scenarijaus failą pavadinimu pavarde.sh. Parašykite scenarijų, kuris išveda į komandų eilutę Jūsų vardą, po to suskaičiuoja kiek Jūsų vardas turi simbolių. Papildomai scenarijus turi sukurti katalogą, kurio vardas bus Jūsų pavarde ir nukopijuoti į šį katalogą failą passwd iš katalogo /etc.

### Mokymosi rezultatas. Valdyti programinius paketus.

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

UŽDUOTIES SĄLYGA: naudojant komandų eilutę arba grafinius įrankius atnaujinti Linux OS ir įdiegti 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.

### 

### Mokymosi rezultatas. Naudoti Web serverio programinę įrangą HTTP bylų viešinimui.

#### *1 užduotis.* WEB SERVERIO DIEGIMAS IR KONFIGŪRAVIMAS.

UŽDUOTIES SĄLYGA: naudojant komandų eilutę įdiegti ir sukonfigūruoti WEB serverį.

Naudojamos priemonės (aplinkos): komandų eilutė.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Paruoškie 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. Pirma 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. Antra 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.

### 

### Mokymosi rezultatas. Valdyti tarnybines stotis per nuotolinę prieigą.

#### *1 užduotis.* NUOTOLINIS PRISIJUNGIMAS PRIE LINUX OS.

UŽDUOTIES SĄLYGA: naudojant putty įrankį atlikti nuotolinį serverio valdymą.

Naudojamos priemonės (aplinkos): Windows 7, Xubuntu OS, putty arba kitty.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Diegiame Xubuntu OS openssh serverį.

2. Sukonfigūruojame SSH serverį, kad jis veiktų ant 15000 prievado.

3. SSH serverio konfigūracijoje uždraudžiame root vartotojo prisijungimą.

4. Iš Windows 7, naudojant putty įrankį prisijungiame prie savo virtualios Xubuntu OS.

## 2. Kompetencija. Taikyti aktualias programinės įrangos kūrimo metodikas.

Šioje dalyje pateikiamos užduotys skirtos žinioms, suvokimui ir gebėjimui analizuoti programinės įrangos kūrimo procesą patikrinti. Dėl ribotos Scrum terminologijos lietuvių k. ir geresnio įsisavinimo dalis klausimų ir terminų pateikiami naudojant originalias sąvokas ar frazes anglų k.

### 

### 2.1. Mokymosi rezultatas. Suprasti SCRUM proceso dalis ir komandos narių atsakomybes.

#### *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

### Mokymosi rezultatas. Analizuoti pateiktus reikalavimus, nustatant programos atitikimą reikalavimams.

#### *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).

### Mokymosi rezultatas. Naudoti projekto eigos valdymo principus.

#### 

#### *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)

## 3. Kompetencija. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą.

### 

### Mokymosi rezultatas. Diegti ir valdyti programavimo Java kalba darbo aplinką.

#### *1 užduotis.* UŽRAŠYKITE PENKIAS INTEGRUOTAS KŪRIMO APLINKAS (IDE), KURIOS GALI BŪTI NAUDOJAMOS PROGRAMUOJANT JAVA KALBA.

#### *2 užduotis.* UŽPILDYKITE PATEIKTĄ LENTELĘ (JAVA IDE PALYGINIMAS PAGAL (*1 užduotį*):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **IDE (version)** | **Syntax highlighting** | **Code Completion** | **Refactoring** | **Version control** | **Debugging** |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

#### *3 užduotis.* PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE ECLIPSE INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Rinkitės Eclipse for Java EE Developers versiją

#### *4 užduotis.* PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE INTELLIJ IDEA INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Rinkitės Intellij Idea Community versiją

#### *5 užduotis.* PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE APACHE NETBEANS INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Rinkitės paskutinę stabilią IDE versiją.

#### *6 užduotis.* APRAŠYKITE PAGRINDINES ECLIPSE IDE APLINKOS FUNKCIJAS (FEATURES).

#### *7 užduotis.* APRAŠYKITE PAGRINDINES INTELLIJ IDEA IDE APLINKOS FUNKCIJAS (FEATURES).

#### *8 užduotis.* APRAŠYKITE PAGRINDINES NETBEANS IDE APLINKOS FUNKCIJAS (FEATURES).

#### *9 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ SU ECLIPSE IDE.

#### Projekto pavadinimas Eclipse Java Project, paketo pavadinimas eclipse.java, klasės pavadinimas HelloWorld.java.

#### *10 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ SU INTELLIJ IDEA IDE.

#### Projekto pavadinimas Intellij Idea Java Project, paketo pavadinimas intellij.idea.java, klasės pavadinimas HelloWorld.java.

#### *11 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ SU NETBEANS IDE.

#### Projekto pavadinimas NetBeans Java Project, paketo pavadinimas netbeans.java, klasės pavadinimas HelloWorld.java.

#### *12 užduotis.* Sukurtam Java projektui (*9 užduotis*) sukonfigūruokite Git įrankį.

#### *13 užduotis.* Sukurtam Java projektui (*10 užduotis*) sukonfigūruokite Git įrankį.

### 

### Mokymosi rezultatas. Sekti programavimo darbų vykdymą naudojant komandinio darbų planavimo sistemas.

#### *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.

#### *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.

### 

### Mokymosi rezultatas. Vykdyti programinio kodo versijavimą naudojant programinio kodo versijavimo įrankius, tinkamus Java kalbai.

#### *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.

#### *9 užduotis.* NUKOPIJUOKITE KODO PAVYZDĮ IŠ NUOTOLINĖS GITHUB KODO SAUGYKLOS: HTTPS://GITHUB.COM/ILUWATAR/JAVA-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 Java 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?

# Modulis „Įvadas į darbo rinką“

## *TESTAS ĮSIVERTINTI GEBĖJIMAMS BAIGUS PROGRAMĄ*

1. Koks skaičius yra užrašytas dvejetainėje sistemoje 1010?

a) 101

b) 10

c) 11

2. Kiek bitų turi vienas kilobaitas (1KB)?

a) 1024

b) 2048

c) 8192

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

a) 1000

b) 1111

c) 0111

4. Koks algoritmas yra realizuotas pateiktame kode:

public class Algorithm {

public static void main(String[] args) {

int[] numbers = new int[]{8, 2, 7, 1, 4, 9, 5};

int lar = numbers[0];

for (int i = 1; i < numbers.length; i++) {

if (numbers[i] > lar){

lar = numbers[i];

}

}

System.out.println(lar);

}

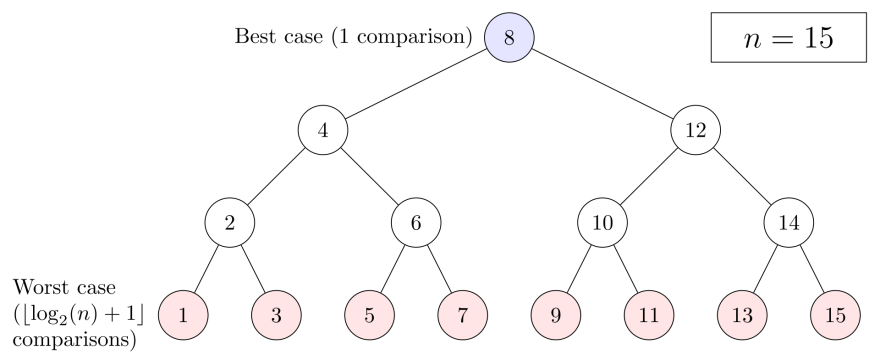
}

a) minimalaus masyvo elemento paieška

b) maksimalaus masyvo elemento paieška

c) rūšiavimo algoritmas

5. Koks algoritmas pavaizduotas?

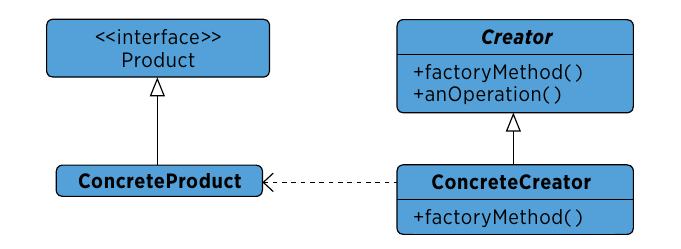


a) nuoseklios paieškos

b) dvejetainės paieškos

c) linijinės paieškos

6. Koks šablonas yra pavaizduotas?

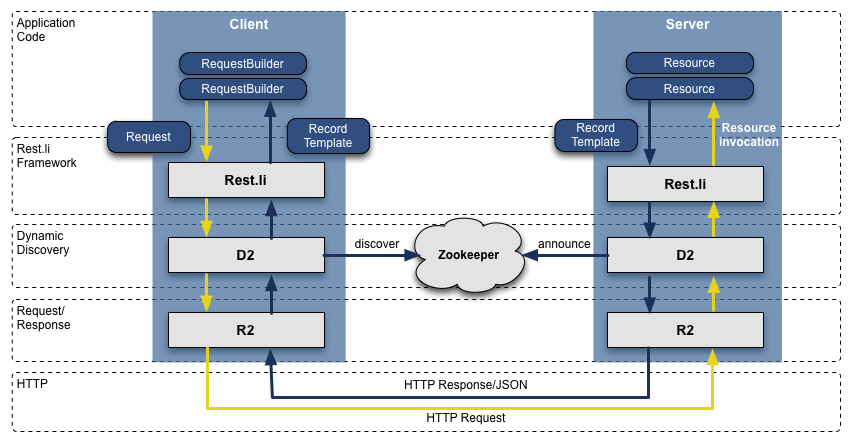


a) Singleton

b) Factory Method

c) Observer

7. Kokia architektūra yra pavaizduota?



a) Kliento - Serverio

b) Modelio - Vaizdo - Kontrolerio

c) Brokerio

8. Java tai:

a) kava

b) sala

c) programavimo kalba

9. Java įrankis javac yra skirtas:

a) surinkti (sukompiliuoti) Java kodą į baito kodą

b) paleisti Java programą

c) sukurti paleidžiamą Java archyvą

10. Kuris duomenų tipo pakeitimas yra teisingas?

a) int intNumber = 10; double doubleNumber = intNumber;

b) double doubleNumber = 25,6; char charSymbol = doubleNumber;

c) long longNumber = 12456785221L; byte byteNumber = longNumber;

11. Kuris iš nurodytų Java paketų realizuoja darbo su failų sistema funkcionalumą?

a) java.text

b) java.lang

c) java.io

12. Kuris Java sąrašas (Collection) užtikrina unikalių elementų aibę?

a) ArrayList

b) HashSet

c) TreeMap

13. Koks Java įrankis generuoja JavaDoc dokumentaciją?

a) javadoc

b) javap

c) javah

14. Inkapsuliacija tai:

a) galimybė programuoti nežinant konkrečių detalių

b) informacijos slėpimas

c) objektų organizavimas specializuojant egzistuojančius bendresnius tipus

15. TDD tai:

a) laiko padalijimo dvipusis ryšys (Time Division Duplex)

b) techninio projektavimo direktyva (Technical Design Directive)

c) testais valdomas vystymas (Test Driven Development)

16. Koks testavimo karkasas yra dažniausiai naudojamas Java programose?

a) JUnit

b) CUnit

c) JSpec

17. Kokia yra populiariausia pasaulyje darbų planavimo sistema (Java projektai)?

a) Jira

b) Workzone

c) Microsoft Project

18. GitHub tai:

a) programa darbui su Git įrankiu

b) „GitHub“ yra kūrimo platforma, skirta bendram IT projektų vystymui, naudojanti Git įrankį

c) grafinė aplinka skirta dirbti su Git įrankiu lokaliame kompiuterje

19. Kiek šakų gali būti Git repozitorijoje?

a) 20

b) priklauso nuo kompiuterio resursų

c) kiek norime

20. 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.

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

a) nefunkcinis

b) funkcinis

c) sisteminis

**Užduočių atsakymai**

# Modulis „Įvadas į profesiją“

## *TESTO ATSAKYMAI*

1. a)

2. c)

3. a)

4. b)

5. a)

6. b)

7. b)

8. c)

9. b)

10. a)

11. b)

12. c)

13. b)

14. a)

15. c)

# Modulis „Informacinių sistemų projektavimas ir kūrimas (Java)“

## 1. Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.

### Mokymosi rezultatas. Pateikti internetinio puslapio turinį naudojant kompiuterinę žymėjimo kalbą.

#### *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>

#### *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>

* 1. Placiau.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>

* 1. Poilsines.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), didelė 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 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>

1. 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;

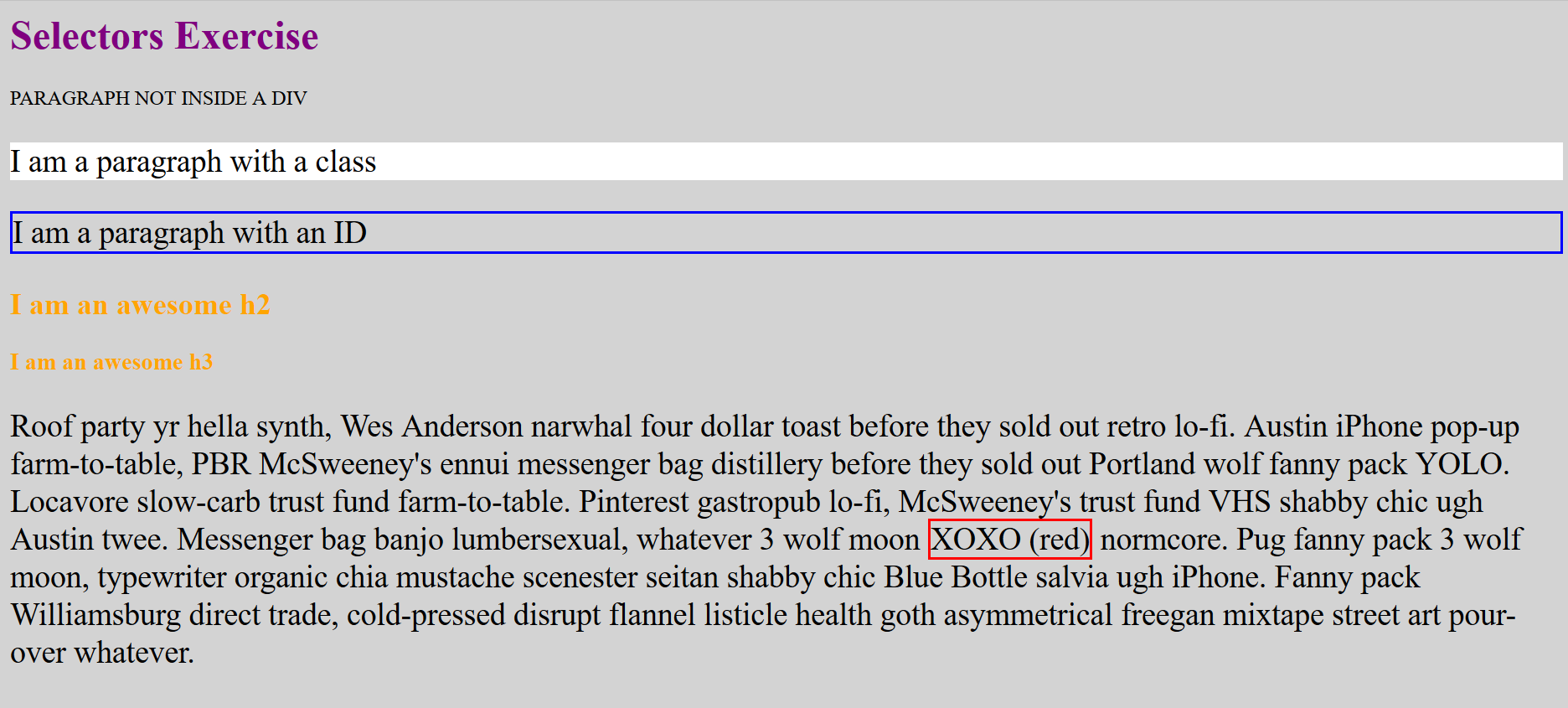
}

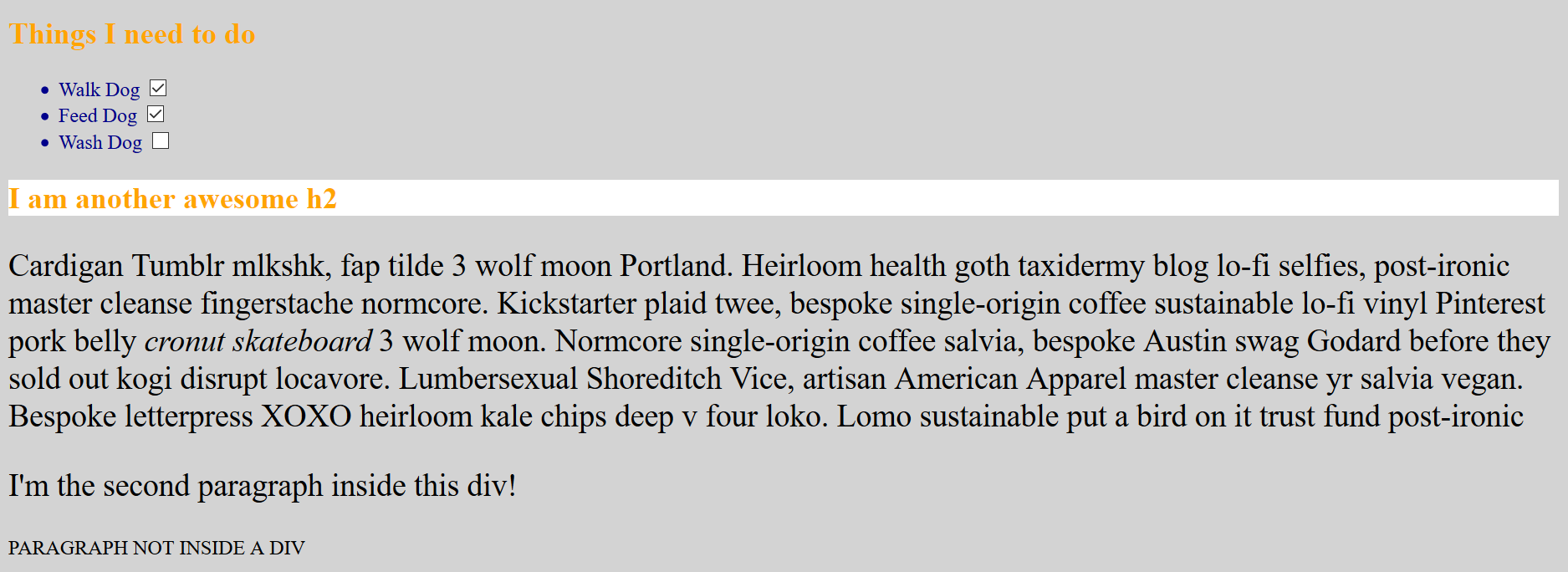
div > ul{

    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;

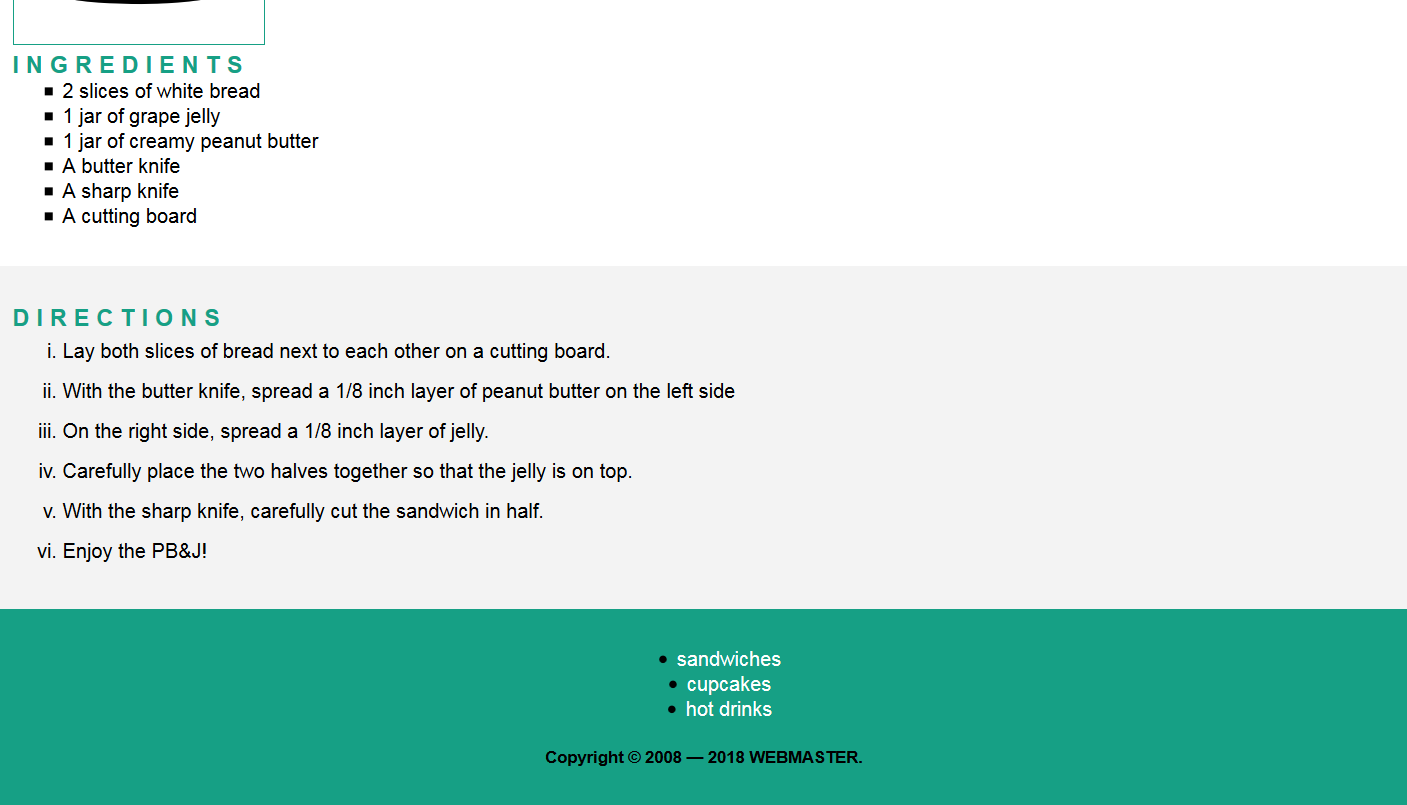
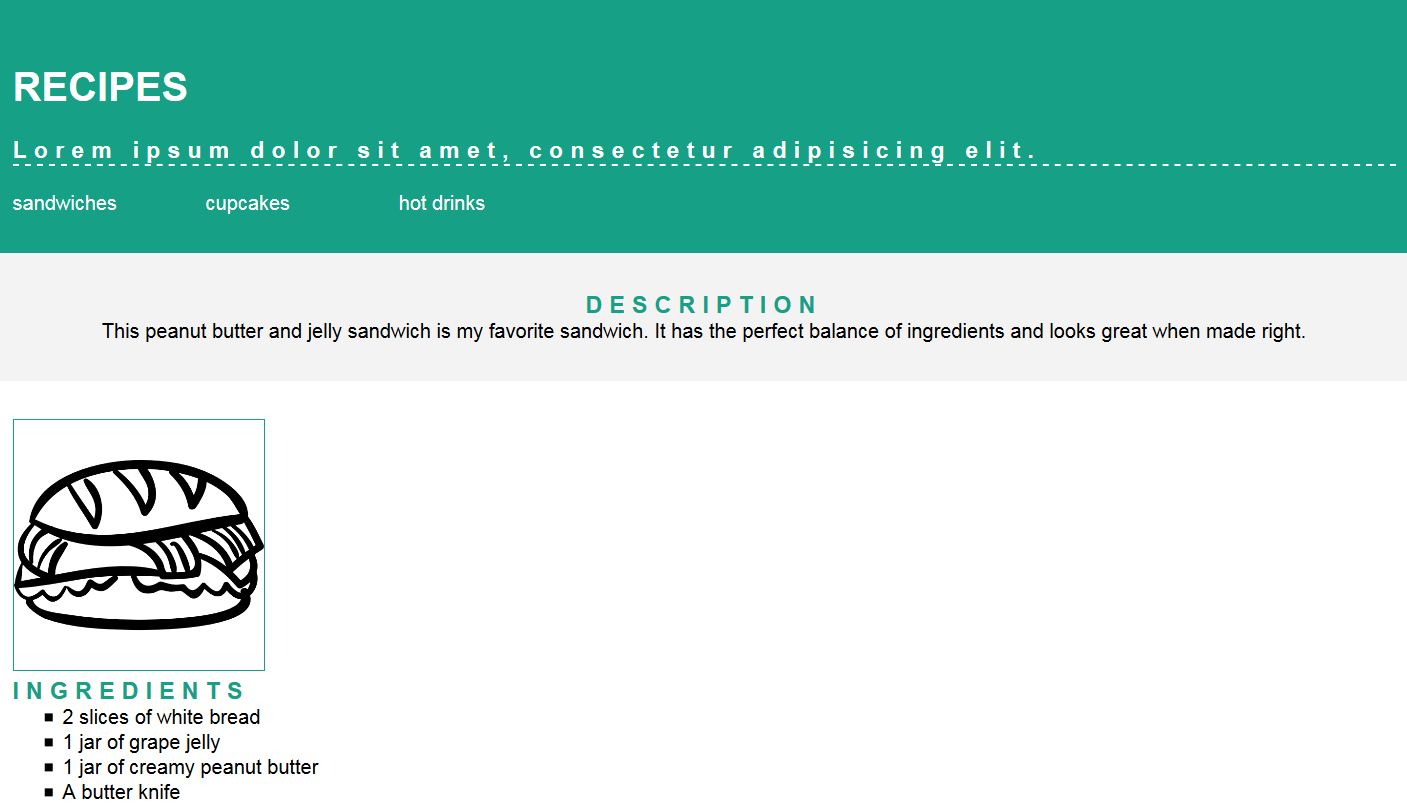
\*/

footer h5{

    padding-top: 20px;

}

Bendras rezultatas naršyklėje turi atrodyti taip:



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

Bendras rezultatas naršyklėje turi atrodyti taip:







### 1.3. Mokymosi rezultatas. Programuoti vartotojo užduočių vykdymą naudojant JavaScript kalbą.

#### *1 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>

1. 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);

})();

1. 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;

}

#### 

#### *2 užduotis.* SUKURKITE 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();

});

1. 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 ;

}

#### *3 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;

}

*4 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();

}

});

});

## 2. Kompetencija. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus.

### Mokymosi rezultatas. Suprasti skaičiavimo sistemas.

*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:

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

*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.

### Mokymosi rezultatas. Taikyti algoritmų ir logikos mokslo pagrindus programuojant.

#### *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)**

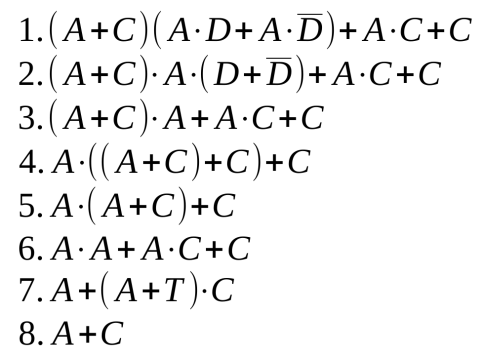
Skaitoma: “Netiesa, kad p ir q ekvivalentiška tam, kad ne-p arba ne-q”. Interpretacija. Pasižymime:

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.>

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

#### Algoritmas su Java programine 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.

#### Algoritmas su Java programine 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.

#### Algoritmas su Java programinė 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.

#### Algoritmas su Java programinė 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.

#### Algoritmas su Java programinė 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.

#### Algoritmas su Java programinė 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.

#### Algoritmas su Java programinė 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.

#### Algoritmas su Java programinė 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);

}

}

### 

### Mokymosi rezultatas. Taikyti programinio kodo dizaino modelius programuojant.

#### *1 užduotis*. UŽPILDYKITE PATEIKTĄ LENTELĘ.

#### Java 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:

1.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.Tutorial Points (2019). *Design Patterns in Java Tutorial.* [žiūrėta 2019-05-27]. Prieiga per internetą [*https://www.tutorialspoint.com/design\_pattern*](https://www.tutorialspoint.com/design_pattern/)*.*

#### *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.

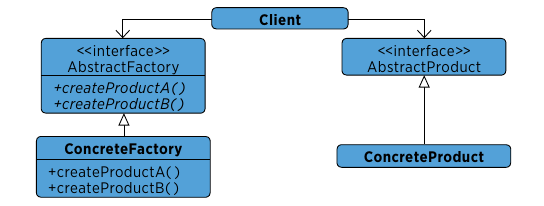
**Decorator šablonas (Dekoratorius):** naudojamas dinamiškai keisti ir pridėti funkcionalumo objektui. Objekto atsakomybė ir elgesys turėtų būti keičiami dinamiškai. Konkretūs įgyvendinimai turėtų būti atsieti nuo atsakomybės ir elgesio. Pavyzdys: daugelis įmonių savo pašto sistemas sukūrė naudodamosi 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.

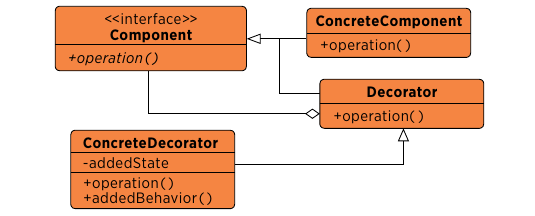
Papildomai galite paskaityti čia:

Wikipedia (2019). *Software design pattern.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_design_pattern.>

#### *3 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ ABSTRAKČIOS FABRIKOS ŠABLONUI (ABSTRACT FACTORY PATTERN).

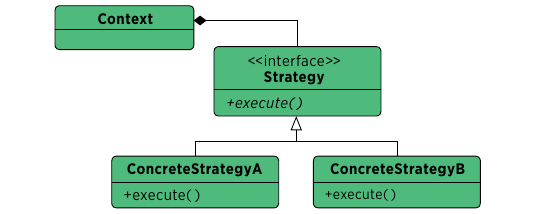


#### *4 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ DEKORATORIAUS ŠABLONUI (DECORATOR PATTERN).



#### *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 fabrikos šabloną.

Papildomai galite paskaityti:

HowToDoInJava (2019). Java Factory Pattern Explained. [žiūrėta 2019-05-27]. Prieiga per internetą <https://howtodoinjava.com/design-patterns/creational/implementing-factory-design-pattern-in-java.>

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

#### Geriausiai šioje situacijoje naudoti dekoratoriaus šabloną.

Papildomai galite paskaityti:

HowToDoInJava (2019). *Decorator Design Pattern in Java*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://howtodoinjava.com/design-patterns/structural/decorator-design-pattern.>

#### *8 užduotis.* PARAŠYKITE JAVA PROGRAMĄ, KURI REALIZUOTŲ ADAPTER ŠABLONĄ.

package adapterpattern;

public interface Bird {

public void fly();

public void makeSound();

}

package adapterpattern;

public class Sparrow implements Bird {

public void fly() {

System.out.println("Flying");

}

public void makeSound() {

System.out.println("Chirp Chirp");

}

}

package adapterpattern;

public interface ToyDuck {

public void squeak();

}

package adapterpattern;

public class PlasticToyDuck implements ToyDuck {

public void squeak() {

System.out.println("Squeak");

}

}

package adapterpattern;

public class BirdAdapter implements ToyDuck {

Bird bird;

public BirdAdapter(Bird bird) {

this.bird = bird;

}

public void squeak() {

bird.makeSound();

}

}

package adapterpattern;

public class Main {

public static void main(String[] args) {

Sparrow sparrow = new Sparrow();

ToyDuck toyDuck = new PlasticToyDuck();

ToyDuck birdAdapter = new BirdAdapter(sparrow);

System.out.println("Sparrow...");

sparrow.fly();

sparrow.makeSound();

System.out.println("ToyDuck...");

toyDuck.squeak();

System.out.println("BirdAdapter...");

birdAdapter.squeak();

}

}

#### 

#### *9 užduotis.* PARAŠYKITE JAVA PROGRAMĄ, KURI REALIZUOTŲ TEMPLATE METHOD ŠABLONĄ.

package templatemethodpattern;

abstract class OrderProcessTemplate {

public boolean isGift;

public abstract void doSelect();

public abstract void doPayment();

public final void giftWrap() {

try {

System.out.println("Gift wrap successfull");

} catch (Exception e) {

System.out.println("Gift wrap unsuccessful");

}

}

public abstract void doDelivery();

public final void processOrder(boolean isGift) {

doSelect();

doPayment();

if (isGift) {

giftWrap();

}

doDelivery();

}

}

package templatemethodpattern;

public class NetOrder extends OrderProcessTemplate {

@Override

public void doSelect() {

System.out.println("Item added to online shopping cart");

System.out.println("Get gift wrap preference");

System.out.println("Get delivery address.");

}

@Override

public void doPayment() {

System.out.println

("Online Payment through Netbanking, card or Paytm");

}

@Override

public void doDelivery() {

System.out.println

("Ship the item through post to delivery address");

}

}

package templatemethodpattern;

public class StoreOrder extends OrderProcessTemplate {

@Override

public void doSelect() {

System.out.println("Customer chooses the item from shelf.");

}

@Override

public void doPayment() {

System.out.println("Pays at counter through cash/POS");

}

@Override

public void doDelivery() {

System.out.println("Item deliverd to in delivery counter.");

}

}

package templatemethodpattern;

public class TemplateMethodPatternClient {

public static void main(String[] args) {

OrderProcessTemplate netOrder = new NetOrder();

netOrder.processOrder(true);

System.out.println();

OrderProcessTemplate storeOrder = new StoreOrder();

storeOrder.processOrder(true);

}

}

### 

### Mokymosi rezultatas. Naudoti informacinių sistemų kūrimo principus ir metodus programinės įrangos projektavime.

#### *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:

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 JAVA DARBUOTOJŲ VALDYMO PROGRAMĄ.

#### Užduoties sprendimui galite panaudoti pateiktą kodą:

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 SPRING KARKASĄ SUKURKITE PAPRASTĄ WEB VARTOTOJŲ REGISTRACIJOS PROGRAMĄ.

#### Užduoties sprendimui galite naudoti šį šaltinį:

Java Guides (2019). *Spring MVC Tutorials and Articles*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.javaguides.net/2018/10/spring-mvc-sign-up-form-handling.html.>

#### *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.

#### 

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:

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.

#### 

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.

### Mokymosi rezultatas. Naudoti Java programavimo kalbos įrankius ir sintaksę.

*1 užduotis*. WINDOWS, MAC ARBA LINUX OPERACINĖJE SISTEMOJE ĮSIDIEKITE JRE (JAVA RUNTIME ENVIRONMENT.

#### JRE diegimo paketą siunčiames iš:

Oracle (2019). Java SE Downloads. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.oracle.com/technetwork/java/javase/downloads/index.html.>

Diegimo ir konfigūravimo instrukcijos įvairioms JRE platformoms:

Oracle (2019). Java Platform, Standard Edition Documentation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/en/java/javase.>

Patikriname komandų eilutėje JRE versiją: java -version

#### *2 užduotis*. WINDOWS, MAC ARBA LINUX OPERACINĖJE SISTEMOJE ĮSIDIEKITE JDK (JAVA DEVELOPMENT KIT).

#### JDK diegimo paketą siunčiames iš:

Oracle (2019). Java SE Downloads. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.oracle.com/technetwork/java/javase/downloads/index.html.>

Diegimo ir konfigūravimo instrukcijos įvairioms JDK platformoms:

Oracle (2019). Java Platform, Standard Edition Documentation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/en/java/javase.>

Patikriname komandų eilutėje JDK versiją: javac –version

#### *3* *užduotis*. NAUDOJANT TEKSTINĮ REDAKTORIŲ (PAVYZDŽIUI: NOTEPAD++) SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE HELLOWORLD.JAVA.

#### Tekstinio redaktoriaus notepad++ diegimo paketą galite atsiųsti iš čia:

Don Ho (2019). Notepad++ download. [žiūrėta 2019-05-27]. Prieiga per internetą <https://notepad-plus-plus.org/download.>

Pilnas klasės HelloWorld turinys:

package com.oracle.javabasics;

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World! (Text editor)");

}

}

Sukurtą klasę patalpiname kataloge com/oracle/javabasics, kur com yra pagrindinis katalogas, o oracle ir javabasics yra vidiniai katalogai.

Daugiau informacijos apie Java paketų pavadinimus ir katalogų struktūrą rasite čia:

Oracle (2019). Lesson: Packages. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/package/index.html.>

4 užduotis. NAUDOJANT INTEGRUOTĄ KŪRIMO APLINKĄ (PAVYZDŽIUI: ECLIPSE) SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE HELLOWORLD.JAVA.

#### Integruotos kūrimo aplinkos Eclipse diegimo paketą galite atsiųsti iš čia:

Eclipse (2019). Download Eclipse Technology that is right for you. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/downloads.>

Java projekto kūrimo ir konfigūracijos instrukcija:

JavaTutorialHQ (2019). First Java programm. [žiūrėta 2019-05-27]. Prieiga per internetą <http://javatutorialhq.com/java/java-hello-world.>

Pilnas klasės HelloWorld turinys:

package com.oracle.javabasics;

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World! (IDE)");

}

}

#### *5 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKIUS JAVAC IR JAVA, SURINKITE IR PALEISKITE UŽDUOTYJE PARAŠYTĄ HELLOWORLD PROGRAMĄ.

#### Javac ir java įrankių aprašymą rasite čia:

Oracle (2019). JDK Tools and Utilities. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/8/docs/technotes/tools/index.html.>

Dirbame operacinės sistemos komandinėje eilutėje. Programos surinkimui naudojame komandą javac com/oracle/javabasics/HelloWorld.java (kur com/oracle/javabasics yra projekto katalogai), o surinktos programos paleidimui naudojame komandą java com.oracle.javabasics.HelloWorld.

#### *6 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKIUS JDEPS IR JAVAP, PATIKRINKITE UŽDUOTYJE PARAŠYTOS HELLOWORLD PROGRAMOS PRIKLAUSOMYBES IR PAŽIŪRĖKITE PROGRAMOS „BYTE” KODĄ.

#### Jdeps ir javap įrankių aprašymą rasite čia:

Oracle (2019). JDK Tools and Utilities. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/8/docs/technotes/tools/index.html.>

Dirbame operacinės sistemos komandinėje eilutėje. Programos priklausomybių patikrinimui naudojame komandą jdeps com/oracle/javabasics/HelloWorld.class. Programos “byte” kodą galite pažiūrėti su komanda javap -v com/oracle/javabasics/HelloWorld.

#### *7 užduotis*. NAUDOJANT JAVA KOMANDINĖS EILUTĖS ĮRANKĮ JAR SUKURKITE UŽDUOTYJE PARAŠYTOS HELLOWORLD PROGRAMOS PALEIDŽIAMĄ ARCHYVĄ.

#### Jar įrankio aprašymą rasite čia:

Oracle (2019). JDK Tools and Utilities. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/8/docs/technotes/tools/index.html.>

Dirbame operacinės sistemos komandinėje eilutėje. Surinkti paleidžiamą jar archyvą galite su komanda:

jar cfe HelloWorld.jar com.oracle.javabasics.HelloWorld com/oracle/javabasics/HelloWorld.class

Java Classpath parametras ir classloader.

#### *8 užduotis*. NAUDOJANT TEKSTINĮ REDAKTORIŲ SUKURKITE JAVA PROJEKTĄ SU VIENA JAVA KLASE JAVACLASSPATH.JAVA.

#### Kodo surinkimui naudojame komandą javac JavaClasspath.java. Kodo paleidimui naudojame komandą java -classpath <absoliutus kelias į jūsų projekto katalogą> com.javabasics.JavaClasspath.

#### Papildoma informacija apie Java Classpath:

1. Wikipedia (2019). Classpath (Java). [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Classpath_(Java).>

2. Oracle (2019). Lesson: The Platform Environment. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/essential/environment/paths.html.>

#### *9 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAELEMENTS.JAVA.

#### 

package javabasics;

public class JavaElements {

private final String NAME = "JAVA ELEMENTS";

private String elementsDescription;

public static int counter = 0;

public static void main(String[] args) {

String elementName = "Hello World";

}

public static void printAll(){

//TODO

}

}

Papildomai galite paskaityti:

Oracle (2019). Classes. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/classes.html.>

#### *10 užduotis*. PAPILDYKITE JAVAELEMENTS PROGRAMĄ KODO KOMENTARAIS.

package javabasics;

/\*

\* This class is for describing class elements:

\* 1. Local constant

\* 2. Object variables

\* 3. Static class variables

\* 4. Local method variables

\*/

public class JavaElements {

// Local constant

private final String NAME = "JAVA ELEMENTS";

// Object variable

private String elementsDescription;

// Static class variable

public static int counter = 0;

public static void main(String[] args) {

// Local method variable

String elementName = "Hello World";

}

/\*\*

\* Prints an array of characters. The characters are converted into bytes

\* according to the platform's default character encoding.

\*

\* @since 1.8

\*

\* @param charArray The array of chars to be printed

\*/

public static void printAll(char [] charArray){

// TODO

}

}

Papildomai galite paskaityti:

JavaAtPoint (2019). Java Comments. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.javatpoint.com/java-comments.>

#### *11 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAEXPRESSIONSSTATEMENTSVARIABLES.JAVA.

package javabasics;

public class JavaExpressionsStatementsVariables {

public static void main(String[] args) {

/\*Variables names: anByte and anInt\*/

byte anByte = 56;

int anInt = 10;

/\*Expressions: name = "Dave" and age = 20\*/

String name = "Dave";

int age = 20;

/\*Statements\*/

int result = age + anInt;

System.out.println(name + ": " + result);

}

}

Papildomai galite paskaityti:

1. Oracle (2019). Expressions, Statements, and Blocks. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/expressions.html.>

2. Oracle (2019). Variables. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html.>

#### *12 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAOPERATORS.JAVA.

#### 

package javabasics;

public class JavaOperators {

public static void main(String[] args) {

/\*Initialise two variables\*/

int numberOne = 100;

int numberTwo = 50;

/\*Operator + \*/

int addResult = numberOne + numberTwo;

/\*Operator - \*/

int minusResult = numberOne - numberTwo;

/\*Operator \* \*/

int multiplicationResult = numberOne \* numberTwo;

/\*Operator / \*/

int divisionResult = numberOne / numberTwo;

/\*Operator ++ \*/

int incrementResult = ++numberOne;

/\*Operator -- \*/

int decrementResult = --numberTwo;

}

}

Papildomai galite paskaityti:

Oracle (2019). Summary of Operators. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html.>

#### *13 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPRIMITIVESDATATYPES.JAVA.

package javabasics;

public class JavaPrimitivesDataTypes {

/\*Primitives data types\*/

static byte primitiveByte;

static short primitiveShort;

static int primitiveInt;

static long primitiveLong;

static float primitiveFloat;

static double primitiveDouble;

static char primitiveChar;

static boolean primitiveBoolean;

public static void main(String[] args) {

/\*Data types default size\*/

System.out.println(primitiveByte);

System.out.println(primitiveShort);

System.out.println(primitiveInt);

System.out.println(primitiveLong);

System.out.println(primitiveFloat);

System.out.println(primitiveDouble);

System.out.println(primitiveChar);

System.out.println(primitiveBoolean);

/\*Data types range\*/

System.out.println("Byte value <Min, Max>: <"

+Byte.MIN\_VALUE + ", " + Byte.MAX\_VALUE + ">");

System.out.println("Short value <Min, Max>: <"

+Short.MIN\_VALUE + ", " + Short.MAX\_VALUE + ">");

System.out.println("Int value <Min, Max>: <"

+Integer.MIN\_VALUE + ", " + Integer.MAX\_VALUE + ">");

System.out.println("Long value <Min, Max>: <"

+Long.MIN\_VALUE + ", " + Long.MAX\_VALUE + ">");

System.out.println("Float value <Min, Max>: <"

+Float.MIN\_VALUE + ", " + Float.MAX\_VALUE + ">");

System.out.println("Double value <Min, Max>: <"

+Double.MIN\_VALUE + ", " + Double.MAX\_VALUE + ">");

System.out.println("Char value <Min, Max>: <"

+Character.MIN\_VALUE + ", " + Character.MAX\_VALUE + ">");

System.out.println("Boolean value <Min, Max>: <"

+Boolean.TRUE + ", " + Boolean.FALSE + ">");

}

}

#### *14 užduotis*. SUKURKITE JAVA KLASĘ VARDU : JAVANONPRIMITIVESDATATYPES.JAVA.

package javabasics;

public class JavaNonPrimitivesDataTypes {

public static void main(String[] args) {

/\*String type object\*/

String text = "Java is a popular programming language, created in 1995. \n"

+"It is owned by Oracle, and more than 3 billion devices run Java.";

System.out.println(text);

/\*String type object reference\*/

String textObjectReference = text.getClass().getSimpleName() + "@"

+ Integer.toHexString(System.identityHashCode(text));

System.out.println("String object reference: " + textObjectReference);

/\*Array type object\*/

int [] intArray = {10, 45, 89, 65, 78, 23};

/\*Array type object reference\*/

System.out.println("Array object reference: " + intArray);

}

}

#### *15 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMETHODS.JAVA.

package javabasics;

public class JavaMethods {

public static void main(String[] args) {

int numberOne = 100;

int numberTwo = 50;

int resultOne = addResult(numberOne, numberTwo);

int resultTwo = substractResult(numberOne, numberTwo);

int resultThree = multiplyResult(numberOne, numberTwo);

int resultFour = divideResult(numberOne, numberTwo);

System.out.println("numberOne + numberTwo: " + resultOne);

System.out.println("numberOne - numberTwo: " + resultTwo);

System.out.println("numberOne \* numberTwo: " + resultThree);

System.out.println("numberOne / numberTwo: " + resultFour);

}

public static int addResult(int numberOne, int numberTwo){

return numberOne + numberTwo;

}

public static int substractResult(int numberOne, int numberTwo){

return numberOne - numberTwo;

}

public static int multiplyResult(int numberOne, int numberTwo){

return numberOne \* numberTwo;

}

public static int divideResult(int numberOne, int numberTwo){

return numberOne / numberTwo;

}

}

#### 

#### *16 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMETHODSOVERLOAD.JAVA.

package javabasics;

public class JavaMethodsOverload {

public static void main(String[] args) {

countArea(12, 12);

countArea(12);

countArea(20, 10, 15);

}

public static void countArea(double sideOne, double sideTwo){

double result = sideOne \* sideTwo;

System.out.printf("Square/rectangle area is: %.2f%n", result);

}

public static void countArea(double radius){

double result = Math.PI \* (radius \* radius);

System.out.printf("Circle area is: %.2f%n", result);

}

public static void countArea(double baseOne, double baseTwo, double height){

double result = ((baseOne + baseTwo) / 2) \* height;

System.out.printf("Trapezoid area is: %.2f%n", result);

}

}

### 

### Mokymosi rezultatas. Kurti nesudėtingą programinį kodą Java programavimo kalba.

#### 

#### *1 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPROPERTIESWRITE.JAVA.

package javasimple;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.util.Properties;

public class JavaPropertiesWrite {

public static void main(String[] args) {

Properties prop = new Properties();

OutputStream output = null;

try {

output = new FileOutputStream("config.properties");

// Set the properties value

prop.setProperty("database", "localhost");

prop.setProperty("dbuser", "java");

prop.setProperty("dbpassword", "java\_password");

// Save properties to project root folder

prop.store(output, null);

} catch (IOException io) {

io.printStackTrace();

} finally {

if (output != null) {

try {

output.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

}

}

#### *2 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAPROPERTIESREAD.JAVA.

package javasimple;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.Properties;

public class JavaPropertiesRead {

public static void main(String[] args) {

Properties prop = new Properties();

InputStream input = null;

try {

input = new FileInputStream("config.properties");

// Load a properties file

prop.load(input);

// Set the property value and print it out

System.out.println("Database address: " + prop.getProperty("database"));

System.out.println("Database user name: " + prop.getProperty("dbuser"));

System.out.println("Database user password: " + prop.getProperty("dbpassword"));

} catch (IOException ex) {

ex.printStackTrace();

} finally {

if (input != null) {

try {

input.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

}

#### *3 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVADATACONVERT.JAVA.

package javasimple;

public class JavaDataConvert {

public static void main(String[] args) {

/\*Widenning java data convert\*/

byte byteValue = 100;

System.out.println("Byte number: " + byteValue);

/\*Byte to short conversion\*/

short shortValue = byteValue;

System.out.println("Byte to short conversion: " + shortValue);

/\*Short to int conversion\*/

int intValue = shortValue;

System.out.println("Short to int conversion: " + intValue);

/\*Int to Long conversion\*/

long longValue = intValue;

System.out.println("Int to Long conversion: " + longValue);

char charValue = '@';

System.out.println("Char: " + charValue);

/\*Char to int conversion\*/

int intFromChar = charValue;

System.out.println("Char to int conversion: " + intFromChar);

/\*Char to long conversion\*/

long longFromChar = charValue;

System.out.println("Char to long conversion: " + longFromChar);

double doubleVariable = 10.45f;

/\*Long to float conversion\*/

float floatFromLong = longValue;

System.out.println("Long to float conversion: " + floatFromLong);

/\*Int to double conversion\*/

double doubleFromInt = intValue;

System.out.println("Int to double conversion: " + doubleFromInt);

/\*Float to double conversion\*/

double doubleFromFloat = doubleVariable;

System.out.println("Float to double conversion: " + doubleFromFloat);

/\*Narrowing java data convert\*/

double doubleNumber = 100.7;

long longNumber = (long) doubleNumber;

int intNumber = (int) doubleNumber;

System.out.println("Double number: " + doubleNumber);

System.out.println("Double to long conversion: " + longNumber);

System.out.println("Double to int conversion: " + intNumber);

checkInstance(doubleNumber);

checkInstance(intValue);

}

public static void checkInstance(Number number) {

if (number instanceof Double) {

System.out.println("Number is a Double");

} else if(number instanceof Integer) {

System.out.println("Number is an Integer");

}

if (number instanceof Comparable) {

System.out.println("Number is comparable");

}

}

}

Papildomai galite paskaityti:

MRBool (2019). Java Data Type Conversion. [žiūrėta 2019-05-27]. Prieiga per internetą <http://mrbool.com/java-data-type-conversion/29257.>

#### *4 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAAUTOBOXINGUNBOXING.JAVA.

package javasimple;

public class JavaAutoboxingUnboxing {

public static void main(String[] args) {

/\*Autoboxing\*/

int intPrimitiveNumber = 36;

Integer intToObject = intPrimitiveNumber;

System.out.println("Primitive int number: " + intPrimitiveNumber);

System.out.println("Boxed primitive int number: " + intToObject);

checkInstance(intToObject);

double doublePrimitiveNumber = 12.48;

Double doubleToObject = doublePrimitiveNumber;

System.out.println("Primitive double number: " + doublePrimitiveNumber);

System.out.println("Boxed primitive double number: " + doubleToObject);

checkInstance(doubleToObject);

/\*Unboxing\*/

Integer integerNumber = new Integer(5);

int number = integerNumber;

System.out.println("Uboxed Integer number: " + number);

}

public static void checkInstance(Number number) {

if (number instanceof Double) {

System.out.println("Number is a Double");

} else if(number instanceof Integer) {

System.out.println("Number is an Integer");

}

}

}

Papildomai galite paskaityti:

Oracle (2019). Autoboxing and Unboxing. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html.>

#### *5 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASTREAMS.JAVA. KLASĖ TURI BŪTI PATALPINTA Į PAKETĄ VARDU JAVASIMPLE.

package javasimple;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class JavaStreams {

private static List<String> fruits = new ArrayList<>();

public static void main(String[] args) {

scannerReader();

fruitFilter();

}

private static void scannerReader() {

Scanner scanner = new Scanner(System.in);

for (int i = 0; i < 4; i++) {

System.out.println("Input " + (i + 1) + " fruit: ");

String line = scanner.next();

fruits.add(line);

}

}

private static void fruitFilter(){

System.out.println("Filtered fruits list: ");

fruits.stream().filter(s -> s.contains("a"))

.map(String::toLowerCase).sorted()

.forEach(System.out::println);

}

}

#### *6 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVABUFFERS.JAVA.

package javasimple;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

public class JavaBuffers {

public static void main(String[] args) {

File file = new File("poem.txt");

int aCount = 0;

try (FileReader fileReader = new FileReader(file);

BufferedReader bufferedReader = new BufferedReader(fileReader)) {

StringBuilder stringBuilder = new StringBuilder();

String currentLine = bufferedReader.readLine();

while (currentLine != null) {

stringBuilder.append(currentLine);

stringBuilder.append("\n");

currentLine = bufferedReader.readLine();

}

System.out.println(stringBuilder);

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

Failo poem.txt turinys:

Advanced topics of object oriented programming.

Presenting the world as a random collection

Of objects is dubious. Far from perfection,

It hardly can pass for a decent expression

Of our senses. (With equal passion

We could be designing a storing procedure

For local morgues.) The essential feature

Of any good system of views, is consensus ndash;

Of rational thought and the deepest of senses ndash;

What “s called intuition. But let “s take a glance at

This seemingly faceless and meaningless concept.

An object in short, is a representation

Of any one thing, (or, in fact, combination

Of things). And despite what the instincts would tell you,

An object can (painfully) have no value,

Thus taking some space to declare an instance

Of nothing that can justify its existence.

#### *7 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAIO.JAVA.

#### 

package javasimple;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

public class JavaIO {

public static void main(String[] args) throws Exception{

try (FileInputStream fis = new FileInputStream("poem.txt");

FileOutputStream fos = new FileOutputStream("new\_poem.txt")){

int stack;

while ((stack = fis.read()) != -1) {

fos.write(stack);

}

} catch (FileNotFoundException e) {

System.out.println(e);

}

}

}

Failo poem.txt turinys:

But there is one thing that can cause us to toddle:

Unwittingly, we, by creating the model,

Are changing the source (or the world, which within it

Has now the model itself). It “s the limit

Of logic, that jeopardizes the mission.

The programmers call it a racing condition:

Imagine a person in front of one mirror,

While holding another hellip; the imminent fear

In face of the endless array of reflections,

The open door to infinity. Actions

Like this can effectively lead to a tumor

In sensitive brains. Let “s resort to some humor,

And follow the syntax, as well as the ethics,

Of programming languages.

#### *8 užduotis*. SUKURKITE JAVA IŠVARDINIMŲ (ENUM) KLASĘ VARDU JAVAENUM.JAVA.

package javasimple;

public enum JavaEnums {

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

}

Papildomai galite paskaityti:

Oracle (2019). Enum Types. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html.>

#### 9 *užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAWORKWITHENUM.JAVA.

package javasimple;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class JavaWorkWithEnum {

private static List<String> weekDays = new ArrayList<>();

public static void main(String[] args) {

for (JavaEnums javaEnums: JavaEnums.values()) {

weekDays.add(javaEnums.name().toLowerCase());

System.out.print(javaEnums.toString().toLowerCase());

}

for (int i = 0; i < weekDays.size(); i++){

weekDays.set(i, (i + 1) + " " + weekDays.get(i));

}

System.out.println("Days of the week: ");

System.out.println(weekDays);

}

}

#### *10 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASQUAREAREA.JAVA.

package javasimple;

public class JavaSquareArea {

private double sideOne;

private double sideTwo;

public JavaSquareArea(double sideOne, double sideTwo) {

this.sideOne = sideOne;

this.sideTwo = sideTwo;

System.out.println(this.sideOne \* this.sideTwo);;

}

}

#### *11 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAWORKWITHPACKAGE.JAVA .

package javapackage;

import javasimple.JavaSquareArea;

public class JavaWorkWithPackage {

public static void main(String[] args) {

JavaSquareArea squareArea = new JavaSquareArea(4, 4);

}

}

Papildomai galite paskaityti:

#### Oracle (2019). *Lesson: Packages.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/package/index.html.>

#### *12 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAARRAYLIST.JAVA.

package javasimple;

import java.util.ArrayList;

import java.util.Collections;

public class JavaArrayList {

public static void main(String[] args) {

ArrayList<Integer> arrayList = new ArrayList<>();

for (int i = 0; i < 20; i++) {

arrayList.add((int) (Math.random() \* 100));

}

System.out.println("Integer number array: " + arrayList);

Collections.sort(arrayList);

System.out.println("Sorted Integer number array: " + arrayList);

}

}

#### *13 užduotis.* SUKURKITE JAVA KLASĘ VARDU JAVALINKEDLIST.JAVA.

package javasimple;

import java.util.ArrayList;

import java.util.Collections;

import java.util.LinkedList;

public class JavaLinkedList {

public static void main(String[] args) {

LinkedList<Integer> linkedList = new LinkedList<>();

for (int i = 0; i < 20; i++) {

linkedList.add((int) (Math.random() \* 1000));

}

System.out.println("Integer number array: " + linkedList);

Collections.shuffle(linkedList);

System.out.println("Shuffled Integer number array: " + linkedList);

}

}

#### *14 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAMAP.JAVA.

#### 

package javasimple;

import java.util.Collections;

import java.util.HashMap;

import java.util.Random;

import java.util.Set;

public class JavaMap {

public static void main(String[] args) {

HashMap<Integer, Integer> hashMap = new HashMap<>();

for (int i = 0; i < 100; i++){

hashMap.put(i, (int) (Math.random() \* 1000));

}

Set<Integer> keys = hashMap.keySet();

for(Integer key: keys) {

System.out.println("Value of " + key + " is: " + hashMap.get(key));

}

if (hashMap.containsValue(100)) {

System.out.println("The hashmap contains number 100.");

} else {

System.out.println("The hashmap does not contains number 100.");

}

}

}

#### *15 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVASET.JAVA.

#### 

package javasimple;

import java.util.TreeSet;

public class JavaSet {

public static void main(String[] args) {

TreeSet<Integer> treeSet = new TreeSet<>();

for (int i = 0; i < 40; i++){

treeSet.add((int) (Math.random() \* 20));

}

System.out.println(treeSet);

}

}

#### *16 užduotis.* IŠNAGRINĖKITE PATEIKTĄ KODĄ IR SURASKITE VISAS KLAIDAS, KURIAS PADARĖ PROGRAMUOTOJAS, KURIS NESINAUDOJA JAVA KODAVIMO STANDARTAIS.

#### Teisingas kodas:

package javasimple;

public class JavaStandards {

public static void main(String[] args) {

String text = "We are comparing content";

StringBuffer stringBuffer = new StringBuffer("We are comparing content");

if (text.contentEquals(stringBuffer)) {

System.out.println("The content is equal");

} else {

System.out.println("The content is not equal");

}

}

}

*17 užduotis*. NAUDOJANT JAVADOC STANDARTĄ PARAŠYKITE PRASMINGUS KOMENTARUS PATEIKTAM JAVA KODUI.

package javasimple;

import java.io.IOException;

/\*\*

\* <h1>Add Two Numbers!</h1>

\* The AddNum program implements an application that

\* simply adds two given integer numbers and Prints

\* the output on the screen.

\* <p>

\* <b>Note:</b> Giving proper comments in your program makes it more

\* user friendly and it is assumed as a high quality code.

\*

\* @author author

\* @version 1.0

\* @since 2019-04-01

\*/

public class AddNum {

/\*\*

\* This method is used to add two integers. This is

\* a the simplest form of a class method, just to

\* show the usage of various javadoc Tags.

\*

\* @param numA This is the first paramter to addNum method

\* @param numB This is the second parameter to addNum method

\* @return int This returns sum of numA and numB.

\*/

public int addNum(int numA, int numB) {

return numA + numB;

}

/\*\*

\* This is the main method which makes use of addNum method.

\*

\* @param args Unused.

\* @return Nothing.

\* @throws IOException On input error.

\* @see IOException

\*/

public static void main(String args[]) throws IOException {

AddNum obj = new AddNum();

int sum = obj.addNum(10, 20);

System.out.println("Sum of 10 and 20 is :" + sum);

}

}

### 

### 3.3. Mokymosi rezultatas. Taikyti objektinio programavimo principus programuojant.

#### *1 užduotis*. SUKURKITE JAVA KLASĘ VARDU CAR.JAVA.

package javaobjects;

public class Car {

private String model;

private String color;

private double speed;

private String fuelType;

public Car(String model, String color, double speed, String fuelType) {

this.model = model;

this.color = color;

this.speed = speed;

this.fuelType = fuelType;

carDrive();

}

public String getModel() {

return model;

}

public String getColor() {

return color;

}

public double getSpeed() {

return speed;

}

public String getFuelType() {

return fuelType;

}

public void carDrive(){

if (fuelType != null){

System.out.println("The car " + this.model + " can drive!");

} else {

System.out.println("The car " + this.model + " can't drive!");

}

}

}

#### *2 užduotis*. SUKURKITE JAVA KLASĘ VARDU MAIN.JAVA .

package javaobjects;

public class Main {

public static void main(String[] args) {

Car toyota = new Car("Yaris", "Red", 210, "Gasoline");

Car mercedes = new Car("GLE", "White", 250, "Diesel");

if (toyota.getSpeed() > mercedes.getSpeed()){

System.out.println("Toyota speed > Mercedes speed");

} else {

System.out.println("Mercedes speed > Toyota speed");

}

}

}

#### *3 užduotis*. PAKEISKITE UŽDUOTYJE SUKURTĄ CAR.JAVA KLASĘ TAIP, KAD BŪTŲ PASLĖPTA PAPROGRAMĖ CANDRIVE().

#### Taip pat pakeiskite parametro „spalva” savybę taip, kad jo negalima būtų keisti.

package javaobjects;

class Car {

private String model;

private static final String COLOR = "White";

private double speed;

private String fuelType;

Car(String model, double speed, String fuelType) {

this.model = model;

this.speed = speed;

this.fuelType = fuelType;

carDrive();

}

String getModel() {

return model;

}

double getSpeed() {

return speed;

}

String getFuelType() {

return fuelType;

}

private void carDrive(){

if (fuelType != null){

System.out.println("The car " + this.model + " can drive!");

} else {

System.out.println("The car " + this.model + " can't drive!");

}

}

}

#### *4 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVAENCAPSULATION.JAVA.

package javaobjects;

public class JavaEncapsulation {

private String name;

private double price;

private double vat;

public JavaEncapsulation(String name, double price, double vat) {

this.name = name;

this.price = price;

this.vat = vat;

}

String getName() {

return this.name;

}

private double getPrice() {

return this.price;

}

private double getVat() {

return vat;

}

public final double getPriceWithVat() {

return this.getPrice() \* this.getVat();

}

}

#### *5 užduotis*. SUKURKITE JAVA KLASĘ VARDU TOYOTA.JAVA.

#### 

package javaobjects;

public class Toyota extends Car{

private String engineType;

Toyota(String model, double speed, String fuelType, String engineType) {

super(model, speed, fuelType);

this.engineType = engineType;

}

public String getEngineType() {

return engineType;

}

}

#### *6 užduotis*. SUKURKITE JAVA KLASĘ VARDU MERCEDES.JAVA.

package javaobjects;

public class Mercedes extends Car{

private int seatNumber;

Mercedes(String model, double speed, String fuelType, int seatNumber) {

super(model, speed, fuelType);

this.seatNumber = seatNumber;

}

public int getSeatNumber() {

return seatNumber;

}

}

#### *7 užduotis*. PAKEISKITE UŽDUOTYJE SUKURTĄ TOYOTA.JAVA KLASĘ TAIP, KAD AUTOMOBILIS GALĖTŲ NE TIK VAŽIUOTI BET IR SKRISTI.

package javaobjects;

public class Toyota extends Car{

private String engineType;

Toyota(String model, double speed, String fuelType, String engineType) {

super(model, speed, fuelType);

this.engineType = engineType;

this.carDrive();

}

public String getEngineType() {

return engineType;

}

private void carDrive(){

if (super.getFuelType() != null){

System.out.println("The car " + getModel() + " can fly!");

} else {

System.out.println("The car " + getModel() + " can't fly!");

}

}

}

#### *8 užduotis*. SUKURKITE KLASES DEMOPOLYMORPHISM.JAVA IR TEST.JAVA.

package javaobjects;

class DemoPolymorphism {

public int add(int x, int y) {

return x + y;

}

public int add(int x, int y, int z) {

return x + y + z;

}

public int add(double x, int y) {

return (int) x + y;

}

public int add(int x, double y) {

return x + (int) y;

}

}

package javaobjects;

public class Test {

public static void main(String[] args) {

DemoPolymorphism demoPolymorphism = new DemoPolymorphism();

System.out.println(demoPolymorphism.add(2, 3));

System.out.println(demoPolymorphism.add(2, 3, 4));

System.out.println(demoPolymorphism.add(2, 3.4));

System.out.println(demoPolymorphism.add(2.5, 3));

}

}

#### *9 užduotis.* PAGAL PATEIKTĄ PARDUOTUVĖS KLASIŲ DIAGRAMĄ SUKURKITE REIKALINGAS KLASES IR INTERFEISUS SU REIKALINGAIS KLASIŲ LAUKAIS, PAPROGRAMĖMIS.

#### 

package javaobjects;

public abstract class AbstractProduct {

private String name;

private double price;

public AbstractProduct(String name, double price) {

this.name = name;

this.price = price;

}

public String getName() {

return name;

}

public double getPrice() {

return price;

}

public final double getPriceWithVat(){

return this.getPrice() \* this.getVat();

}

public abstract double getVat();

}

package javaobjects;

public class Product extends AbstractProduct implements ShoppingCartItem{

private static final double VAT = 1.21;

public Product(String name, double price) {

super(name, price);

}

@Override

public double getVat() {

return VAT;

}

}

package javaobjects;

public class Periodical extends AbstractProduct implements ShoppingCartItem{

private static final double PERIODICAL\_VAT = 1.09;

public Periodical(String name, double price) {

super(name, price);

}

@Override

public double getVat() {

return PERIODICAL\_VAT;

}

}

package javaobjects;

public class Medicine extends AbstractProduct implements ShoppingCartItem{

private static final double MEDICINE\_VAT = 1.05;

public Medicine(String name, double price) {

super(name, price);

}

@Override

public double getVat() {

return MEDICINE\_VAT;

}

}

package javaobjects;

public interface ShoppingCartItem {

double getPrice();

double getPriceWithVat();

}

package javaobjects;

import java.util.ArrayList;

import java.util.List;

public class ShoppingCart{

private List<ShoppingCartItem> items = new ArrayList<>();

public void add(ShoppingCartItem item){

items.add(item);

}

public double getTotalPrice(){

double totalPrice = 0;

for (ShoppingCartItem item : items){

totalPrice += item.getPrice();

}

return totalPrice;

}

public double getTotalPriceWithVat(){

double totalPriceWithVat = 0;

for (ShoppingCartItem item : items){

totalPriceWithVat += item.getPriceWithVat();

}

return totalPriceWithVat;

}

}

package javaobjects;

public class Main {

public static void main(String[] args) {

Medicine panadol = new Medicine("Panadol", 12.45);

Periodical newspaper = new Periodical("Lietuvos rytas", 3.00);

ShoppingCart myCart = new ShoppingCart();

myCart.add(panadol);

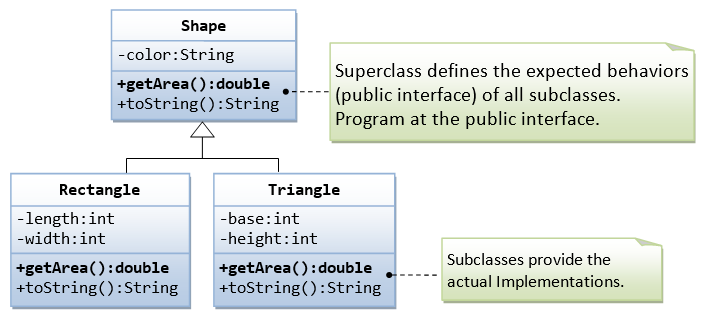
myCart.add(newspaper);

myCart.getTotalPrice();

}

}

#### *10 užduotis*. PAGAL PATEIKTAS KLASES NUBRAIŽYKITE UML JAVA KLASIŲ DIAGRAMĄ.



### 3.4. Mokymosi rezultatas. Testuoti programinę įrangą naudojant su Java programavimo kalba suderinamus testavimo įrankius ir metodus.

#### *1 užduotis*. SUKURKITE NAUJĄ JAVA PROJEKTĄ VARDU JAVATESTING.

package javajunit;

class JavaForTesting {

private String name;

public JavaForTesting(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

#### *2 užduotis*. PARAŠYKITE *1 užduotyje* SUKURTAM PROJEKTUI VIENĄ PAPRASTĄ TESTĄ, KURIS PATIKRINA AR KLASĖ JAVAFORTESTING VEIKIA TEISINGAI.

package javajunit;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class JavaForTestingTest {

@Test

public void nameReturnCorrectly(){

JavaForTesting javaForTesting = new JavaForTesting("Dave");

assertEquals(

"Dave",

javaForTesting.getName()

);

}

}

#### *3 užduotis.* SUKURKITE NAUJĄ JAVA PROJEKTĄ VARDU TESTING.

package javatesting;

public class Calculator {

private int numberOne;

private int numberTwo;

public Calculator(int numberOne, int numberTwo) {

this.numberOne = numberOne;

this.numberTwo = numberTwo;

}

public double add(){

return numberOne + numberTwo;

}

public double substract(){

return numberOne - numberTwo;

}

public double multiply(){

return numberOne \* numberTwo;

}

public double divide(){

if (numberTwo == 0){

return 0.0;

}

return (double) (numberOne / numberTwo);

}

}

package javatesting;

import javatesting.Calculator;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class CalculatorTest {

@Test

public void addTwoNumbersCorrectly(){

Calculator calculator = new Calculator(2, 2);

assertEquals(

4,

calculator.add()

);

}

@Test

public void substractTwoNumbersCorrectly(){

Calculator calculator = new Calculator(-2, -2);

assertEquals(

0,

calculator.substract()

);

}

@Test

public void multiplyTwoNumbersCorrectly(){

Calculator calculator = new Calculator(-2, -2);

assertEquals(

4,

calculator.multiply()

);

}

@Test

public void divideTwoNumbersCorrectly(){

Calculator calculator = new Calculator(2, 0);

assertEquals(

0,

calculator.divide()

);

}

}

#### *4 užduotis.* 3 KOMPETENCIJOS 3.3. MOKYMOSI REZULTATO *9 užduotyje* REALIZUOTAI PARDUOTUVEI SU PREKIŲ KREPŠELIU PARAŠYKITE JUNIT TESTUS, KURIE PATIKRINTŲ PRODUKTŲ GRUPIŲ IR PREKIŲ KREPŠELIO FUNKCIONALUMĄ.

package javaobjects;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class ShoppingCartTest {

@Test

public void calculatesTotalCartPrice() {

ShoppingCart shoppingCart = new ShoppingCart();

shoppingCart.add(new TestCartItem(2.2));

shoppingCart.add(new TestCartItem(2.0));

assertEquals(

4.2,

shoppingCart.getTotalPrice()

);

}

@Test

public void calculatesTotalCartPriceWithVat() {

ShoppingCart shoppingCart = new ShoppingCart();

shoppingCart.add(new TestCartItem(1.0));

shoppingCart.add(new TestCartItem(2.0));

assertEquals(

6.0,

shoppingCart.getTotalPriceWithVat()

);

}

private static class TestCartItem implements ShoppingCartItem {

private double price;

public TestCartItem(double price) {

this.price = price;

}

@Override

public double getPrice() {

return this.price;

}

@Override

public double getPriceWithVat() {

return this.price \* 2;

}

}

}

package javaobjects;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class ProductTest {

@Test

public void productHasNameAndPrice() {

Product product = new Product("Product Name", 1.5);

assertEquals(

"Product Name",

product.getName()

);

assertEquals(

1.5,

product.getPrice()

);

}

@Test

public void calculates21PercentVat() {

Product product = new Product("", 2.0);

assertEquals(

2.42,

product.getPriceWithVat()

);

}

}

package javaobjects;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class MedicineTest {

@Test

public void medicineReturnsPriceWith5PercentVatApplied() {

Medicine medicine = new Medicine("Aspirin", 2.0);

assertEquals(

2.1,

medicine.getPriceWithVat(),

0.01

);

}

}

package javaobjects;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class PeriodicalTest {

@Test

public void returnsPriceWith9PercentVatApplied() {

Periodical periodical = new Periodical("Java", 2.0);

assertEquals(

2.18,

periodical.getPriceWithVat(),

0.01

);

}

}

#### *5 užduotis*. SUKURKITE JAVA KLASĘ VARDU SIMPLEEXCEPTION.JAVA.

package javaexceptions;

import java.util.Arrays;

public class SimpleException {

public static void main(String[] args) {

int[] intArray = new int[10];

try {

for (int i = 0; i < 20; i++) {

intArray[i] = (int) (Math.random() \* 10);

}

} catch (ArrayIndexOutOfBoundsException ex) {

System.out.println("Array is too small! \n" + ex);

} finally {

System.out.println(Arrays.toString(intArray));

}

}

}

#### *6 užduotis*. SUKURKITE JAVA KLASĘ VARDU FILEREADERS.JAVA .

package javaexceptions;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

public class FileReaders {

private static final String FILENAME = "textfile.txt";

private static String line = null;

public static void main(String[] args) {

try {

FileReader fileReader = new FileReader(FILENAME);

BufferedReader bufferedReader = new BufferedReader(fileReader);

while((line = bufferedReader.readLine()) != null) {

System.out.println(line);

}

bufferedReader.close();

} catch (FileNotFoundException ex) {

System.out.println("Unable to open file '" + FILENAME + "'");

} catch (IOException ex) {

System.out.println("Error reading file '" + FILENAME + "'");

}

}

}

#### *7 užduotis*. SUKURKITE JAVA KLASĘ VARDU JAVALOGGER.JAVA.

package javaloggers;

import java.util.logging.Level;

import java.util.logging.Logger;

public class JavaLogger {

private static final Logger LOGGER = Logger.getLogger(JavaLogger.class.getName());

public static void main(String[] args) {

LOGGER.log(Level.INFO, "Program start.");

System.out.println("Fibonacci number < 1000:");

LOGGER.log(Level.INFO, "Fibonacci number count ...");

int a = 1, b = 1;

System.out.print(a + ", " + b);

int c = a + b;

do {

System.out.print(", " + c);

a = b;

b = c;

c = a + b;

} while (c < 1000);

System.out.println(".");

LOGGER.log(Level.INFO, "End program.");

}

}

#### *8 užduotis.* SUKURKITE JAVA KLASĘ VARDU JAVAFILELOGGER.JAVA.

package javaloggers;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class JavaFileLogger {

private final static Logger LOGGER = LoggerFactory.getLogger(JavaFileLogger.class);

public static void main(String[] args) {

LOGGER.info("Program start.");

System.out.println("Multiplication table");

LOGGER.info("StringBuilder initialisation.");

StringBuilder line = new StringBuilder();

for (int i = 0; i < 51; ++i)

line.append("-");

System.out.println(line);

System.out.print("| X |");

for (int i = 1; i < 10; ++i)

System.out.printf("%3d |", i);

System.out.println();

System.out.println(line);

for (int j = 1; j < 10; ++j) {

System.out.printf("|%3d |", j);

for (int i = 1; i < 10; ++i)

System.out.printf("%3d |", i \* j);

System.out.println();

}

System.out.println(line);

LOGGER.info("Program end.");

}

}

# Root logger option

log4j.rootLogger=INFO, file

# configuration to print into file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=javafilelogger.log

log4j.appender.file.MaxFileSize=12MB

log4j.appender.file.MaxBackupIndex=10

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n

#### *9 užduotis*. IŠBANDYKITE PATEIKTĄ JAVA PROGRAMINĮ KODĄ.

public class JavaDebug {

public static void main(String[] args) {

System.out.println("Square perimeter and area counter.");

Scanner input = new Scanner(System.in);

System.out.print("Input square side, a = ");

double a = input.nextDouble();

input.close();

if (a > 0) {

double area = Math.sqrt(a);

double perimeter = 4 \* a;

System.out.println("Square area A = " + area);

System.out.println("Square perimeter P = " + perimeter);

}

if (a <= 0) {

System.out.println("Error! The length of the side is to be a positive number.");

}

}

}

Papildomai galite paskaityti:

1. Oracle (2019). jdb. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdb.html.>

2. JetBrains (2019). Debugging Your First Java Application. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html.>

3. TutorialPoints (2019). Eclipse - Debugging Program. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_debugging_program.htm.>

#### *10 užduotis.* NORINT PATIKRINTI ŽMOGAUS KŪNO MASĖS INDEKSĄ, REIKALINGAS SKAIČIUOTUVAS.

#### Sukurkite projektą bmi, skirtą šiai užduočiai .

#### Panaudokite komandinio darbų planavimo sistemas: Jira (Atlassian), Microsoft Project (Microsoft), Trello (Atlassian), HeySpace (Time Solutions TimeCamp Inc.), Taiga (Taiga) ir pan..

## 

## 4. Kompetencija. Analizuoti skirtingų tipų reikalavimus, apibūdinančius kompiuterinę programą.

### Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo (angl. user story) reikalavimų programinei įrangai formatą.

#### *1 užduotis*. UŽRAŠYKITE VARTOTOJO PASAKOJIMO (USER STORY) FORMATĄ.

#### 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. As the HR manager, I want to create a screening quiz so that I can understand whether I want to send possible recruits to the functional manager.

2. As a manager, I want to browse my existing quizzes so I can recall what I have in place and figure out if I can just reuse or update an existing quiz for the position I need now.

3. As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

Papildomai galite paskaityti:

Wikipedia (2019). User story. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/User_story.>

#### *2 užduotis.* YRA PATEIKTI FUNKCINIAI PROGRAMINĖS ĮRANGOS REIKALAVIMAI. PABANDYKITE SUFORMUOTI TRIS VARTOTOJO PASAKOJIMUS (USER STORIES).

#### Vartotojo pasakojimo pavyzdžiai pagal funkcinius reikalavimus:

1. Aš kaip administratorius noriu registruoti naujus vartotojus sistemoje.

2. Aš kaip administratorius noriu keisti vartotojų slaptažodžius.

3. Aš kaip vartotojas noriu prisijungti prie sistemos su savo vartotojo vardu ir slaptažodžiu.

#### *3* *užduoti*s. APRAŠYKITE GERŲ VARTOTOJO PASAKOJIMO FORMATĄ (INVEST)INVEST.

#### Vartotojo pasakojimai turintys šias savybes:

1. Nepriklausomi (independent) - pasakojimai turėtų būti nepriklausomi vienas nuo kito.

2. Aptariami (negotiable) - pasakojimai aptariami ir diskutuojami.

3. Vertingi (valuable) - turi būti vertingi klientui arba pirkėjui.

4. Įvertinami (estimable) - turi būti įmanoma pasakojimus įvertinti.

5. Pakankamai maži (small enough) - rekomenduojamas dydis 1-5 dienos.

6. Patikrinami (testable) - kiekvieną pasakojimą turi būti įmanoma patikrinti (ištestuoti).

#### *4* *užduotis*. APRAŠYKITE REIKALAVIMŲ PERŽIŪROS ŽINGSNIUS.

#### Reikalavimų peržiūra (Requirements Review):

1. Vykdoma sprinto eigoje ruošiant vartotojo pasakojimus kitam sprintui.

2. Peržiūros tikslas, kad kiekvienas peržiūrėtas pasakojimas proceso gale atitiktų reikalavimų parengimo kriterijus (angl. Definition of Ready).

3. Siekiama kad būtų peržiūrėta tiek pasakojimų, kad komanda turėtų darbo visą ateinantį sprintą.

4. Esant poreikiui, vartotojo pasakojimai detalizuojami ir papildomi.

5. Komanda įvertina ar įmanoma pasakojimą realizuoti bei įvertina reikalaujamo darbo apimtį (angl. estimate).

6. Per dideli pasakojimai padalinami į mažesnius.

7. Komanda užduoda klausimus ir teikia pasiūlymus kaip aiškiau ir tiksliau užrašyti vartotojo pasakojimus.

#### *5 užduotis.* APRAŠYKITE REIKALAVIMŲ PARUOŠIMO SPRINTUI KRITERIJUS (DEFINITION OF READY). PATEIKITE PAVYZDŽIŲ.

#### Definition of Ready - tai sąlygos kurias privalo atitikti kokybiškai parašytas vartotojo pasakojimas ir jo priėmimo kriterijai. **Pavyzdys:**

1. vartotojo pasakojimas aprašyta pagal sutartą formatą (nurodomas konkretus sutartas formatas),

2. vartotojo pasakojimas peržiūrėtas ir reikalaujamo darbo apimtis įvertinta komandos,

3. visi su vartotojo pasakojimu susiję klausimai atsakyti ir pastebėti neatitikimai ištaisyti,

4. vartotojo pasakojimo apimties įvertinimas neviršija 8 pasakojimo vienetų (storypoints).

#### *6 užduotis*. APRAŠYKITE REIKALAVIMŲ BAIGTUMO APIBRĖŽIMĄ. PATEIKITE PAVYZDŽIŲ.

#### Definition of Done tai sąlygos kada vartotojo pasakojimas gali būti laikomas užbaigtu. Vartotojo pasakojimas: “Kaip naudotojas aš noriu sukurti užduotį, tam kad nepamirščiau svarbių darbų.”.

#### **Priėmimo/baigtumo kriterijai (Definition of Done):**

1. Galiu sukurti užduotį tik tada, kai nurodau užduoties aprašymą.

2. Užduoties aprašymas gali būti ne ilgesnis negu 300 simbolių.

3. Galiu taip pat nurodyti užduoties baigimo datą.

4. Datos formatas yyyy-mm-dd.

5. Mano sukurtas priminimas matomas tik man.

#### *7 užduotis.* APRAŠYKITE PAGRINDINIUS SCRUM PROCESO ĮVYKIUS.

#### Scrum proceso įvykiai:

1. Išleidimo planavimo susitikimas (Release Planning Meeting) arba produkto darbų sąrašo tvarkymo susitikimas (Backlog Grooming Meeting).

2. Sprinto planavimo susitikimas (Sprint Planning Meeting).

3. Kasdienis scrum susitkimas (Daily Scrum).

4. Sprinto peržiūra (Sprint Review).

5. Sprinto retrospektyva (Sprint Retrospective).

Aprašant kiekvieną įvykį naudokite šį šaltinį:

ScrumGuides (2019). Scrum Events. [žiūrėta 2019-05-27]. Prieiga per internetą https://www.scrumguides.org/scrum-guide.html#events

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

#### *1 užduotis*. APRAŠYKITE KAS YRA FUNKCINIAI REIKALAVIMAI.

#### Funkciniai reikalavimai tai programinės įrangos ar sistemos atliekamų paslaugų aprašymas, sistemos reakcijos į tam tikrus įvedimo ar išvedimo duomenis aprašymas, sistemos elgesys tam tikroje situacijoje, programos reakcijos į tam tikrus vartotojo veiksmus.

#### Pavyzdys: **Sistemoje galima pateikti įvairių tipų dokumentus.**

Sistemos vartotojai gali pateikti dokumentus. Kiekvienas pateikiamas dokumentas yra tam tikro tipo. Dokumentų tipus kuria administratorius. Administratorius nustato, kokių tipų dokumentus vartotojų grupėms leidžiama kurti. Grupė gali turėti teises kuri neribotą kiekį skirtingų tipų dokumentų. Dokumentai pateikiami nurodant (ar sistemai generuojant) atitinkamas savybes. Vartotojas gali sukurti tam tikro tipo dokumentą, jei priklauso bent vienai grupei, kuriai leidžiama sukurti to tipo dokumentą. Pavyzdžiui, organizacijoje yra trys grupės: darbuotojai, konsultantai, administracija. Darbuotojams leidžiama pateikti atostogų prašymus, konsultantams leidžiama pateikti prašymus padengti komandiruotės išlaidas, administracija gali tvirtinti abiejų tipų dokumentus. Vartotojas priklausantis tiek darbuotojų, tiek konsultantų grupėms gali pateikti atostogų prašymus ir prašymus padengti komandiruotės išlaidas. Direktorius priklausantis visoms trims grupėms, gali teikti abiejų tipų dokumentus ir juos tvirtinti.

#### *2 užduotis*. PATEIKTAM PAVYZDŽIUI PARAŠYKITE KAIP BUS NUSTATOMA AR FUNKCINIS REIKALAVIMAS BUVO ĮVYKDYTAS AR NE.

#### Aprašant reikalavimų įvykdymo būseną naudokite šiuos šaltinius:

1. Wikipedia (2019). Software verification and validation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_verification_and_validation.>

2. Wikipedia (2019). Software testing. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_testing.>

#### *3 užduotis*. APRAŠYKITE KAS YRA NEFUNKCINIAI REIKALAVIMAI.

#### Pateikite pavyzdžių.

#### Nefunkciniai reikalavimai tai apribojimai sistemos atliekamoms paslaugoms ar funkcijoms, dažniausiai laiko apribojimai, programavimo proceso apribojimai, standartai ir pan.

#### 

#### Pavyzdys: **Visi veiksmai sistemoje turi atsakyti greičiau nei per 1 sekundę.**

Reikalavimas taikomas laikantis prielaidos, kad sistemoje yra iki 10 tūkstančių vartotojų, kurių kiekvienas yra vidutiniškai pateikęs 20 dokumentų. Sistema yra pasiekiama internetu klientui turint ne platesnį nei 10Mbps ryšio kanalą bei vidutiniškai 50 ms atsako laiką iš serverio.

*4 užduotis*. PATEIKTAM PAVYZDŽIUI PARAŠYKITE KAIP BUS NUSTATOMA AR NEFUNKCINIS REIKALAVIMAS BUVO ĮVYKDYTAS AR NE.

#### Aprašant reikalavimų įvykdymo būseną naudokite šiuos šaltinius:

1. Wikipedia (2019). Software verification and validation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_verification_and_validation.>

2. Wikipedia (2019). Software testing. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_testing.>

# Modulis „Nesudėtingų duomenų bazių projektavimas ir kūrimas (Java)“

## 

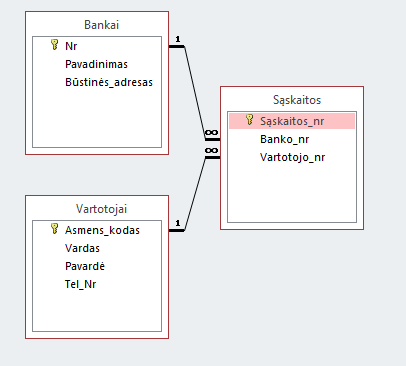
## 1. Kompetencija. Projektuoti tipines reliacines ir objektines duomenų bazes.

### 

### Mokymosi rezultatas. Pateikti reliacinės duomenų bazės schemą.

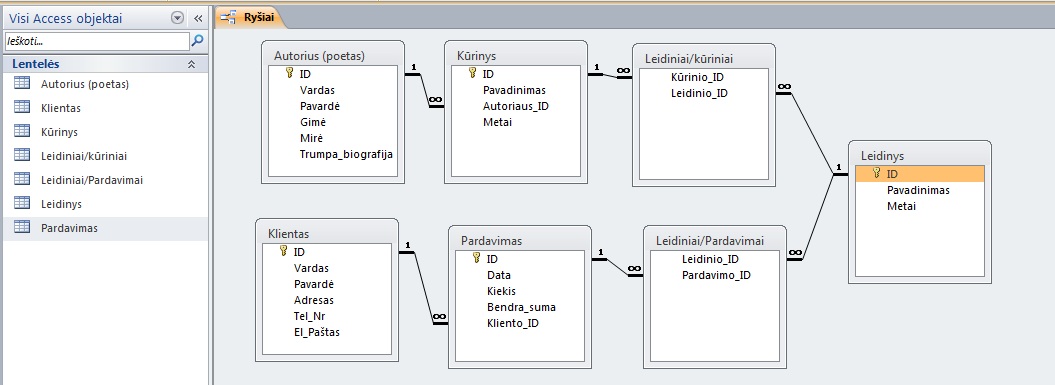
#### *1 užduotis*. SUPROJEKTUOTI RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LIETUVOS BANKAS).

Schema:



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

Schema:



### Mokymosi rezultatas. Pateikti nereliacinės (NoSQL) duomenų bazės schemą.

#### *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;

#### *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;

Alternatyviai 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.

### 

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

#### *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.

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

#### *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’;

### 2.3. Mokymosi rezultatas. Kurti duomenis duomenų bazėje valdančią programinę įrangą.

#### *1 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 (Java)“

## 

## Kompetencija. Naudoti tarnybinių stočių operacines sistemas.

### Mokymosi rezultatas. Administruoti skaitmenines bylas bei tarnybinės stoties vartotojus naudojant tos tarnybinės stoties operacinę sistemą.

#### *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.

#### *4 užduotis*. XUBUNTU OS DIEGIMAS VIRTUALIOJE APLINKOJE.

Užduotis atlikta pagal šias instrukcijas:

* https://itsfoss.com/install-xfce-desktop-xubuntu
* https://help.ubuntu.com/community/Installation?action=show&redirect=InstallingXubuntu
* https://virtualboxes.org/doc/installing-guest-additions-on-ubuntu

#### *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.

### 

### Mokymosi rezultatas. Valdyti tarnybinę stotį naudojant jos komandinės eilutės sąsają ir jos pagrindines komandas.

#### *1 užduotis.* DARBAS LINUX OS KOMANDŲ EILUTĖJE.

Sprendimas:

$ cd ~ , $ mkdir Bandymas

$ touch f{1..4}.txt , $ touch ff{1..4}.txt

$ ls Bandymas/

$ cd ~ , $ mkdir Tikslas

$ cp Bandymas/f1.txt Tikslas/ ir t.t.

$ ls Tikslas/

$ mv Tikslas/f1.txt Tikslas/file1.txt ir t.t.

$ mv Tikslas/file1.txt Bandymas/ ir t.t.

$ cp /etc/passwd Bandymas/

$ less Bandymas/passwd

$ less Bandymas/passwd | grep root

$ wc -m Bandymas/passwd

$ head -n 5 Bandymas/passwd , $ tail -n 5 Bandymas/passwd

$ ln -s Bandymas/passwd Tikslas/passwd

$ ls Tikslas/

$ rm Bandymas/passwd

$ ls Tikslas/

$ rm -dfr Bandymas

#### *2 užduotis.* SUKURKITE BASH SCENARIJŲ.

Sprendimas:

#! /bin/sh

echo “vardas”

echo vardas | wc -m

mkdir Pavarde

cp /etc/passwd Pavarde/

### 

### Mokymosi rezultatas. Valdyti programinius paketus.

#### *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

### Mokymosi rezultatas, Naudoti Web serverio programinę įrangą HTTP bylų viešinimui.

#### *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

### Mokymosi rezultatas. Valdyti tarnybines stotis per nuotolinę prieigą.

#### *1 užduotis*. NUOTOLINIS PRISIJUNGIMAS PRIE LINUX OS.

Sprendimas – veiksmai atlikti pagal šias instrukcijas:

1. https://help.ubuntu.com/lts/serverguide/openssh-server.html
2. https://help.ubuntu.com/lts/serverguide/openssh-server.html
3. https://help.ubuntu.com/lts/serverguide/openssh-server.html
4. https://www.youtube.com/watch?v=tiPZGba2qJg

## Kompetencija. Taikyti aktualias programinės įrangos kūrimo metodikas.

### 

### 2.1. Mokymosi rezultatas. Suprasti SCRUM proceso dalis ir komandos narių atsakomybes.

#### *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*. WHAT IS THE RECOMMENDED SIZE OF A SCRUM TEAM?

#### (4) 7 plus or minus 2

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

#### (3) 15 minučių

#### *5 užduotis*. IN AN ORGANIZATION THAT EMBRACES AGILE VALUES, WHO WOULD BE RESPONSIBLE FOR TOOL SELECTION AND CONFIGURATION?

#### (1) The teams, who would have to coordinate with each other.

### 

### Mokymosi rezultatas. Analizuoti pateiktus reikalavimus, nustatant programos atitikimą reikalavimams.

#### *1 užduotis*. WHAT HAPPENS DURING THE REQUIREMENTS GATHERING AND ANALYSIS PHASE OF THE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)?

#### (2) The customer gives the expectations of the project

#### *2 užduotis.* TRACEABILITY IS NOT CONSIDERED IN REQUIREMENT ANALYSIS.

#### (2) False.

#### 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*. REQUIREMENTS ANALYSIS IS AN ITERATIVE PROCESS .

#### (1) True

#### 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.* HOW IS PRODUCT BACKLOG ARRANGED?

#### (4) Most important items at the top, least important items at the bottom

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

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

#### *6 užduotis*. WHAT DO YOU INFER FROM THE FOLLOWING USER STORY? „AS A SALES AGENT, I WANT A CLIENT SEARCH FEATURE SO THAT I CAN FIND MY PREFERRED CLIENTS QUICKLY AND EASILY“.

#### (2) Good. Need clarification on “quick and easy” for UI testing.

### 

### Mokymosi rezultatas. Naudoti projekto eigos valdymo principus.

#### *1 užduotis*. WHAT IS A GOOD SIZE FOR A SPRINT TASK?

#### (1) One person per day or less, so other team members can easily detect when a task is stuck

#### *2 užduotis*. HOW OFTEN SHOULD BACKLOG GROOMING OCCUR?

#### (3) Every Sprint

#### *3 užduotis.* WHAT IS A SCRUM TEAM EXPECTED TO DO DURING THE FIRST SPRINT? (galimi keli teisingi atsakymai)

#### (1) Test the product ir (4) Build a thin slice of potentially shippable functionality

#### *4 užduotis.* THE CEO ASKS A TEAM MEMBER TO DO SOME WORK OUTSIDE THE GOALS OF THE CURRENT SPRINT IN PROGRESS. WHAT SHOULD THE TEAM MEMBER DO?

#### (3) Inform the Product Owner so he can work with the CEO

#### *5 užduotis.* KADA BAIGIAMAS SPRINTO VYKDYMAS?

#### (4) Kai baigiasi sprintui skirtas laikas (time-box expires)

## Kompetencija. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą.

### 

### 3.1. Mokymosi rezultatas. Diegti ir valdyti programavimo Java kalba darbo aplinką.

#### *1 užduotis.* UŽRAŠYKITE PENKIAS INTEGRUOTAS KŪRIMO APLINKAS (IDE), KURIOS GALI BŪTI NAUDOJAMOS PROGRAMUOJANT JAVA KALBA.

#### 

#### Programuojant Java kalba galime naudoti: NetBeans, Eclipse, Intellij Idea, BlueJ, jEdit, Eclipse Che, DrJava, Geany, Zeus it t.t..

#### *2 užduotis*. UŽPILDYKITE PATEIKTĄ LENTELĘ (JAVA IDE PALYGINIMAS PAGAL *1* *užduotį)*:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **IDE (version)** | **Syntax highlighting** | **Code Completion** | **Refactoring** | **Version control** | **Debugging** |
| 1 | Eclipse (4.11) | yes | yes | yes | yes | yes |
| 2 | Intellij Idea (2019.1) | yes | yes | yes | yes | yes |
| 3 | NetBeans (8.2) | yes | yes | yes | yes | yes |
| 4 | BlueJ (4.2.0) | yes | yes | not fully | no | not fully |
| 5 | Geany (1.34.1) | yes | yes | not fully | no | no |

#### *3 užduotis*. PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE ECLIPSE INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Parsiųsti Eclipse IDE galite iš čia:

Eclipse (2019). Eclipse IDE 2019-03 R Packages. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/downloads/packages.>

Eclipse diegimo instrukcijas rasite čia:

Eclipse (2019). Eclipse/Installation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://wiki.eclipse.org/Eclipse/Installation.>

Eclipse konfigūravimo instrukcijas rasite čia:

Eclipse (2019). Eclipse documentation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://help.eclipse.org/2019-03/index.jsp.>

#### *4 užduotis*. PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE INTELLIJ IDEA INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Parsiųsti Intellij Idea IDE galite iš čia:

JetBrain (2019). Download IntelliJ IDEA. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/idea/download.>

Intellij Idea diegimo instrukcijas rasite čia:

JetBrain (2019). Install IntelliJ IDEA. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/installation-guide.html.>

Intellij Idea konfigūravimo instrukcijas rasite čia:

JetBrain (2019). Configuring Project and IDE Settings. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/configuring-project-and-ide-settings.html.>

#### *5 užduotis*. PARSISIŲSKITE, ĮSIDIEKITE IR SUSIKONFIGŪRUOKITE APACHE NETBEANS INTEGRUOTĄ KŪRIMO APLINKĄ.

#### Parsiųsti NetBeans IDE galite iš čia:

NetBeans (2019). Apache NetBeans Releases. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.apache.org/download/index.html.>

NetBeans diegimo instrukcijas rasite čia:

NetBeans (2019). NetBeans IDE 8.2 Installation Instructions. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/community/releases/82/install.html.>

NetBeans konfigūravimo instrukcijas rasite čia:

Oracle (2019). NetBeans Developing Applications with NetBeans IDE. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/netbeans/nb82/netbeans/NBDAG/working_nbeans.htm.>

#### *6 užduotis*. APRAŠYKITE PAGRINDINES ECLIPSE IDE APLINKOS FUNKCIJAS (FEATURES)

#### Aprašant pagrindines Eclipse IDE aplinkos funkcijas (Features) naudokite:

Eclipse (2019). Platform and Equinox. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/eclipse/news/4.11/platform.php.>

#### *7 užduotis*. APRAŠYKITE PAGRINDINES INTELLIJ IDEA IDE APLINKOS FUNKCIJAS (FEATURES).

#### Aprašant pagrindines Intellij Idea IDE aplinkos funkcijas (Features) naudokite:

JetBrain (2019). Making Development an Enjoyable Experience. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/idea/features.>

#### *8 užduotis*. APRAŠYKITE PAGRINDINES NETBEANS IDE APLINKOS FUNKCIJAS (FEATURES)

#### Aprašant pagrindines Netbeans IDE aplinkos funkcijas (Features) naudokite:

NetBeans (2019). NetBeans IDE - The Smarter and Faster Way to Code . [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/features.>

#### *9 užduotis*. SUKURKITE NAUJĄ JAVA PROJEKTĄ SU ECLIPSE IDE.

#### Apie naujo Java projekto kūrimą Eclipse IDE aplinkoje galite paskaityti čia:

1. TutorialsPoint (2019). Eclipse - Create Java Project. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_create_java_project.htm.>

2. TutorialsPoint (2019). Eclipse - Create Java Package. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_create_java_package.htm.>

3. TutorialsPoint (2019). Eclipse - Create Java Class. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_create_java_class.htm.>

#### *10 užduotis*. SUKURKITE NAUJĄ JAVA PROJEKTĄ SU INTELLIJ IDEA IDE.

#### Apie naujo Java projekto kūrimą Intellij Idea IDE aplinkoje galite paskaityti čia:

JetBrain (2019). Create your first Java application. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html.>

#### *11 užduotis*. SUKURKITE NAUJĄ JAVA PROJEKTĄ SU NETBEANS IDE.

#### Apie naujo Java projekto kūrimą Intellij Idea IDE aplinkoje galite paskaityti čia:

NetBeans (2019). https://netbeans.org/kb/docs/java/quickstart.html. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/kb/docs/java/quickstart.html.>

#### *12 užduotis*. Sukurtam Java projektui (*9 užduotis*) sukonfigūruokite Git įrankį.

#### Git konfigūravimo instrukcija Eclipse IDE Java projektui:

EclipseSource (2019). Installing EGit in Eclipse. [žiūrėta 2019-05-27]. Prieiga per internetą <https://eclipsesource.com/blogs/tutorials/egit-tutorial.>

*13 užduotis*. Sukurtam Java projektui (10 užduotis) sukonfigūruokite Git įrankį.

#### Git konfigūravimo instrukcija Intellij Idea IDE Java projektui:

JetBrain (2019). Git. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/using-git-integration.html.>

### 

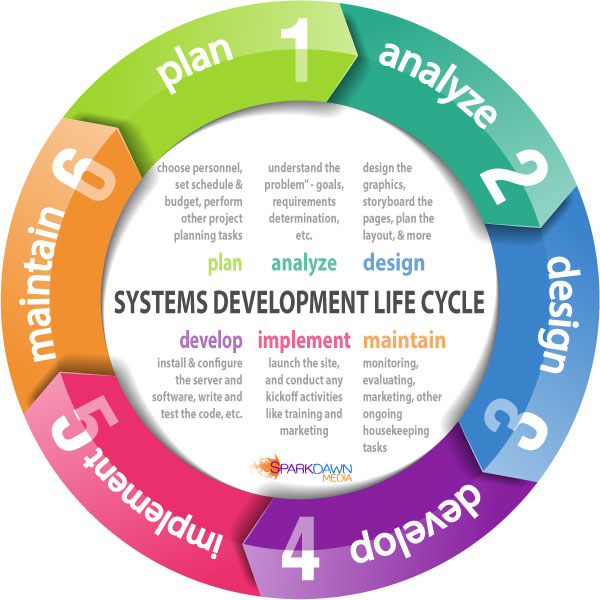
### 3.2. Mokymosi rezultatas. Sekti programavimo darbų vykdymą naudojant komandinio darbų planavimo sistemas.

#### *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:

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.>

#### *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.>

### 

### Mokymosi rezultatas. Vykdyti programinio kodo versijavimą naudojant programinio kodo versijavimo įrankius, tinkamus Java kalbai.

#### *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*](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*](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:

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.>

#### *9 užduotis.* NUKOPIJUOKITE KODO PAVYZDĮ IŠ NUOTOLINĖS GITHUB KODO SAUGYKLOS: <HTTPS://GITHUB.COM/ILUWATAR/JAVA-DESIGN-PATTERNS>.

#### Kodo kopijavimas iš nuotolinės saugyklos:

1. git clone https://github.com/iluwatar/java-design-patterns

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 komandą: git init. Toliau sukuriame Java projektą su klase Calculator.java. Parašome skaičiuotuvo aritmetines operacijas. Kiekvieną parašytą aritmetinį veiksmą įrašome į kodo saugyklą. Pavyzdys sudėties operatoriui:

public class Calculator {

private int numberOne;

private int numberTwo;

private char operator;

Calculator(int numberOne, int numberTwo, char operator) {

this.numberOne = numberOne;

this.numberTwo = numberTwo;

this.operator = operator;

switch (operator) {

case '+':

add(this.numberOne, this.numberTwo);

break;

default:

System.out.println("Not selected operator ...");

}

}

int add(int numberOne, int numberTwo) {

return numberOne + numberTwo;

}

}

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.>

**Modulis „Įvadas į darbo rinką“**

## 

## Testo atsakymai

1. b)

2. c)

3. a)

4. b)

5. b)

6. b)

7. a)

8. c)

9. a)

10. a)

11. c)

12. b)

13. a)

14. b)

15. c)

16. a)

17. a)

18. b)

19. c)

20. a)

21. b)

# Literatūros sąrašas

1. Atlassian (2019). *Atlassian Documentation*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html.>

2. 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.>

3. 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.>

4. Atlassian (2019). *Deploying a release.* [žiūrėta 2019-05-27].Prieiga per internetą<https://confluence.atlassian.com/jirasoftwarecloud/deploying-a-release-764478183.html.>

5. Atlassian (2019). *Issue types*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/adminjiracloud/issue-types-844500742.html>.

6. Atlassian (2019). *Jira Software best practices.* [žiūrėta 2019-05-27].Prieiga per internetą<https://www.atlassian.com/software/jira/guides/getting-started/best-practices.>

7. Atlassian (2019). *Jira Sprints Tutorial*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.atlassian.com/agile/tutorials/sprints.>

8. Atlassian (2019). *Project management for non-project managers*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.atlassian.com/project-management.>

9. 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.>

10. Atlassian (2019). *Working with boards*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jiracorecloud/working-with-boards-800712866.html.>

11. Atlassian (2019). *Working with issues in Jira Software*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.atlassian.com/agile/tutorials/issues.>

12. Atlassian (2019). *Working with issues*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://confluence.atlassian.com/jirasoftwarecloud/working-with-issues-764478424.html.>

13. Cheesman, J., Daniels, J. (2000). *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Professional.

14. Chacon, S., Straub, B. (2019). *Pro Git*. Apress. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2>

15. Don Ho (2019). Notepad++ download. [žiūrėta 2019-05-27]. Prieiga per internetą <https://notepad-plus-plus.org/download.>

16. Eclipse (2019). Eclipse IDE 2019-03 R Packages. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/downloads/packages.>

17. Eclipse (2019). Eclipse/Installation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://wiki.eclipse.org/Eclipse/Installation.>

18. Eclipse (2019). Eclipse documentation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://help.eclipse.org/2019-03/index.jsp>.

19. Eclipse (2019). Download Eclipse Technology that is right for you. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/downloads.>

20. Eclipse (2019). Platform and Equinox. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.eclipse.org/eclipse/news/4.11/platform.php.>

21. EclipseSource (2019). Installing EGit in Eclipse. [žiūrėta 2019-05-27]. Prieiga per internetą <https://eclipsesource.com/blogs/tutorials/egit-tutorial.>

22. GbmbOrg (2019). *Data Units Conversion.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.gbmb.org.>

23. Git (2019). Downloads. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/downloads.>

24. Git (2019). Getting Started - Installing Git. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.>

25. Pri 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.>

26. 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.>

27. Git (2019). Git format patch. [žiūrėta 2019-05-27]. Prieiga per internetą <https://git-scm.com/docs/git-format-patch.>

28. Grand, M. (2002). *Java Enterprise Design Patterns: Patterns in Java*. John Wiley & Sons.

29. Horstmann, Cay S. (2016). *Core Java: Fundamentals*. New York: Prentice Hall.

30. Horstmann, Cay S. (2017). *Core Java: Advanced Features*. New York: Prentice Hall.

31. HowToDoInJava (2019). Java Factory Pattern Explained. [žiūrėta 2019-05-27]. Prieiga per internetą <https://howtodoinjava.com/design-patterns/creational/implementing-factory-design-pattern-in-java.>

32. 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.>

33. Java Guides (2019). *Spring MVC Tutorials and Articles*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.javaguides.net/2018/10/spring-mvc-sign-up-form-handling.html.>

34. JavaTutorialHQ (2019). First Java programm. [žiūrėta 2019-05-27]. Prieiga per internetą <http://javatutorialhq.com/java/java-hello-world.>

35. JavaAtPoint (2019). Java Comments. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.javatpoint.com/java-comments.>

36. JetBrains (2019). Debugging Your First Java Application. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html.>

37. JetBrain (2019). Install IntelliJ IDEA. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/installation-guide.html>.

38. JetBrain (2019). Configuring Project and IDE Settings. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/configuring-project-and-ide-settings.html.>

39. JetBrain (2019). Making Development an Enjoyable Experience. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/idea/features.>

40. JetBrain (2019). Create your first Java application. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html.>

41. JetBrain (2019). Git. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.jetbrains.com/help/idea/using-git-integration.html.>

42. Koskela, L. (2007). *Test Driven: TDD and Acceptance TDD for Java Developers*. Manning Publications.

43. Lietuvos statistikos departamentas (2019). *Ekonominės veiklos rūšių klasifikatorius*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://osp.stat.gov.lt/static/evrk2.htm>

44. Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. New York: Prentice Hall.

45. 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://](https://klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf)

[klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf](https://klevas.mif.vu.lt/~donatas/PSArchitekturaProjektavimas/Knyga/BPD/PSAPKnyga.pdf)

46. MRBool (2019). Java Data Type Conversion. [žiūrėta 2019-05-27]. Prieiga per internetą <http://mrbool.com/java-data-type-conversion/29257.>

47. NetBeans (2019). Apache NetBeans Releases. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.apache.org/download/index.html.>

48. NetBeans (2019). NetBeans IDE 8.2 Installation Instructions. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/community/releases/82/install.html.>

49. NetBeans (2019). NetBeans IDE - The Smarter and Faster Way to Code . [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/features.>

50. NetBeans (2019). https://netbeans.org/kb/docs/java/quickstart.html. [žiūrėta 2019-05-27]. Prieiga per internetą <https://netbeans.org/kb/docs/java/quickstart.html.>

51. Oracle (2019). JDK Tools and Utilities. [žiūrėta 2019-05-27]. Prieiga per internetą Oracle (2019). Lambda Expressions. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html.>

52. Oracle (2019). Java SE Downloads. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.oracle.com/technetwork/java/javase/downloads/index.html.>

53. Oracle (2019). Java Platform, Standard Edition Documentation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/en/java/javase.>

54. Oracle (2019). Lesson: Packages. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/package/index.html.>

<https://docs.oracle.com/javase/8/docs/technotes/tools/index.html.>

55. Oracle (2019). Lesson: The Platform Environment. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/essential/environment/paths.html.>

56. Oracle (2019). Expressions, Statements, and Blocks. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/expressions.html.>

57. Oracle (2019). Variables. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html.>

58. Oracle (2019). Summary of Operators. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html.>

59. Oracle (2019). Autoboxing and Unboxing. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html.>

60. Oracle (2019). Enum Types. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html.>

61. Oracle (2019). jdb. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdb.html.>

62. Oracle (2019). NetBeans Developing Applications with NetBeans IDE. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/netbeans/nb82/netbeans/NBDAG/working_nbeans.htm.>

63. Schildt, H. (2017). Java: The Complete Reference. New York: Prentice Hall.

64. TutorialsPoint (2019). Eclipse - Create Java Project. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_create_java_project.htm.>

65. Tutorials Point (2019). *Number System Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/computer_logical_organization/number_system_conversion.htm>

66. Tutorial Points(2019). *Design Patterns in Java Tutorial.* [žiūrėta 2019-05-27].Prieiga per internetą[https://www.tutorialspoint.com/design\_pattern](https://www.tutorialspoint.com/design_pattern/).

67. TutorialPoints (2019). Eclipse - Debugging Program. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/eclipse/eclipse_debugging_program.htm.>

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. Wiegers E. K. (2005). More About Software Requirements. Microsoft Press.

70. Wikipedia (2019). Numeral system. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Numeral_system>.

71. Wikipedia (2019). Classpath (Java). [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Classpath_(Java).>

72. Wikipedia (2019). List of numeral systems. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/List_of_numeral_systems>.

73. Wikipedia (2019). Units of information. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Units_of_information>.

74. Wikipedia (2019). Data rate units. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Data-rate_units.>

75. Wikipedia (*2019). Boolean Algebra.* [žiūrėta 2019-05-27].Prieiga per internetą<https://en.wikipedia.org/wiki/Boolean_algebra.>

76. 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.)

77. Wikipedia *(2019). Software design pattern.* [žiūrėta 2019-05-27]. Prieiga per internetą<https://en.wikipedia.org/wiki/Software_design_pattern.>

78. Wikipedia (2019). Model View Controller. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Model-view-controller.>

79. Wikipedia (2019). SOLID. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/SOLID.>

80. Wikipedia (2019). Don’t repeat yourself. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Don’t_repeat_yourself.>

81. Wikipedia (2019). Separation of Concerns. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Separation_of_concerns.>

82. Wikipedia (2019). Code reuse. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Code_reuse.>

83. Wikipedia (2019). User story. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/User_story.>

84. Wikipedia (2019). Software verification and validation. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_verification_and_validation.>

85. Wikipedia (2019). Software testing. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Software_testing.>

86. Wikipedia (2019). Systems development life cycle. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Systems_development_life_cycle.>

87. 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.>