

**Php programuotojo modulinė profesinio mokymo**

**programa, IV lygis**

**Teorinių ir praktinių užduočių**

**mokinio sąsiuvinis**

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

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

Teorinių ir praktinių užduočių mokinio sąsiuvinio autoriai:

Rita Daukšienė

Laura Vaišnorė

Jaroslav Grablevski

Jolita Bačkienė

Justina Balsė

Konsultantai:

Martynas Kašelionis

# Modulis „Įvadas į profesiją“

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

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

a) Taip

b) Ne

2. Koks protokolas naudojamas komunikavimui tarp naršyklės ir serverio?

a) HTML

b) Ajax

c) HTTP

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

a) Taip

b) Ne

4. Pagrindinės WEB technologijos?

a) C++, JavaScript, TypeScript, HTML

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

c) HTML, CSS, JavaScript, PHP

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

a) HTML „dynamic“ extension biblioteka

b) Ajax

c) SASS

6. HTML, tai?

a) Programavimo kalba tinklapio interaktyvumui kurti

b) Tinklapio apipavidalinimą aprašanti kalba

c) Tinklapio struktūrą aprašanti kalba

7. Kodėl CSS vadinami pakopiniais stiliais?

a) Kodas rašomas pakopomis

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

c) Teisingo atsakymo nėra

8. Kas bus išvesta įvykdžius pateiktą žemiau esantį kodą?

class Vartotojas {

public $vardas;

public $pavarde;

public function pasisveikink()

{

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

}

}

$vartotojas1 = new User();

$vartotojas1 -> vardas = "Jonas";

$vartotojas1-> pavarde = "Jonaitis";

echo $vartotojas1 -> pasisveikink();

a) Jonas Jonaitis

b) Jonas

c) Klaida

9. Ką vadiname CSS preprocesoriumi?

a) Preprocesorius leidžia rašyti css kodą html elemento viduje

b) Preprocesorius yra procesas, kuris serveryje įvykdo CSS kodą

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

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

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

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

c) Nėra teisingo atsakymo

11. Kuris iš kintamųjų yra masyvas?

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

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

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

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

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

b) antraste.style.backgroundColor = 'tomato'

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

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

a) navigation

b) links

c) nav

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

a) 5

b) 10

c) Teisingo atsakymo nėra

15. Kas bus išvesta įvykdžius PHP kodą ?

<?php

if (3 = = = '3') {

echo 'Taip';

} else {

echo 'Ne';

}

?>

a) Taip

b) Ne

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



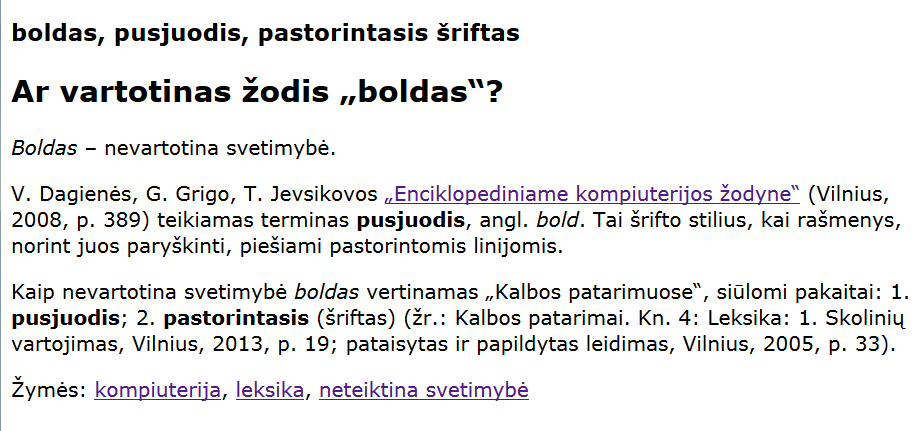
# 1. Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.

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

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

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**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Paskutinė minutė**

Ispanija, Kosta Dorada



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

**Išvykimo laikai:**

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

**Į kelionės kainą įskaičiuota:**

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

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

**Kelionės**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Poilsinės kelionės**

Ispanija, Kosta Dorada



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

Kreta



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

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

**Kelionės**

* Paskutinė minutė
* Poilsinės kelionės
* Kelionių datos

**Kelionių datos**

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

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

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

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

# 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>

2. 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.

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

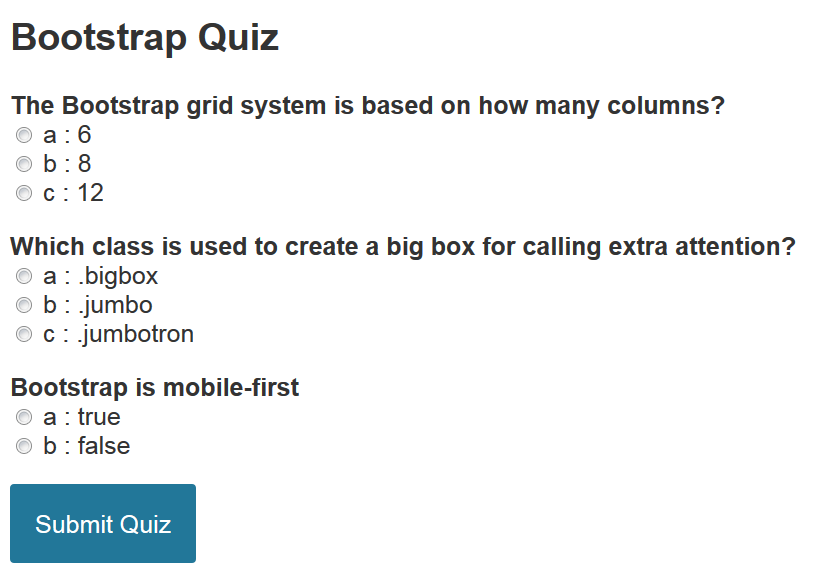
*1 užduotis.* SUKURKITE TESTĄ, TIKRINANTĮ IR PATEIKIANTĮ TEISINGUS ATSAKYMUS.

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>

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 ;

}

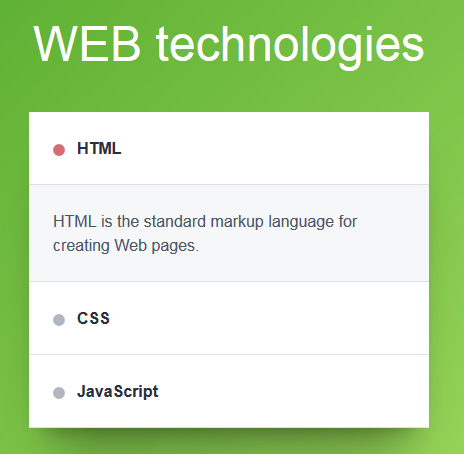
**Reikalavimai:**

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>

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;

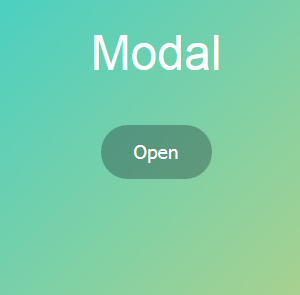
}

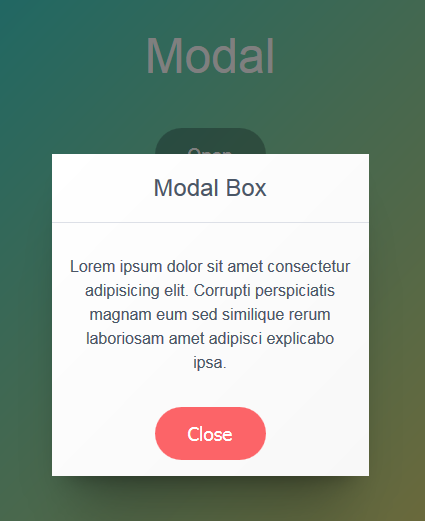
Reikalavimai:

HTML dokumente užrašyti nuorodas į stiliaus ir skripto failus.

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:

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>

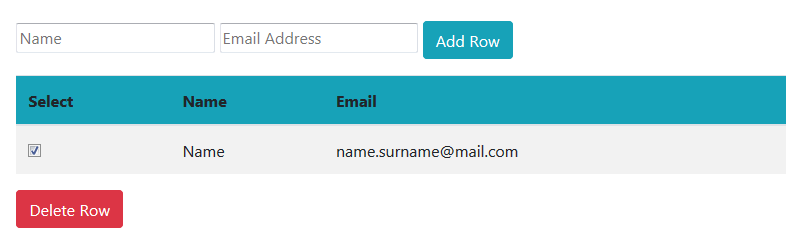
</body>

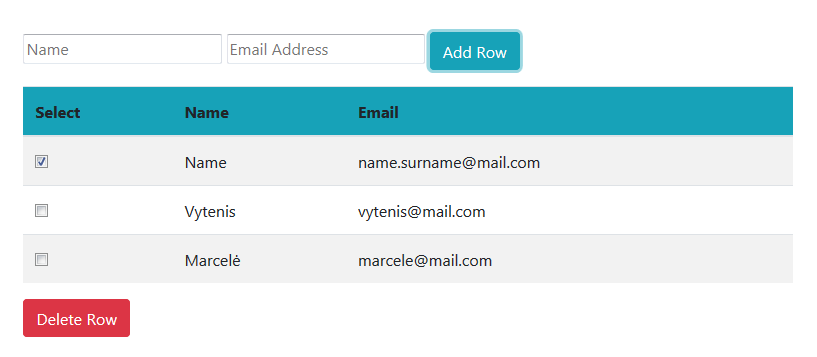
</html>

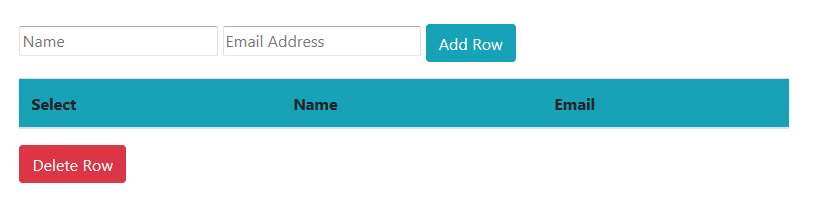
**Reikalavimai:**

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.

# 2.2. 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 RANDA MAŽIAUSIĄ SKAIČIŲ AIBĖJE.

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

# 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 GAMYKLOS“ ŠABLONUI (ABSTRACT FACTORY PATTERN).

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

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

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

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

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

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

# 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 PASIRINKTĄ KARKASĄ, PVZ. LARAVEL 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.

## Kompetencija. Kurti tipinę programinę įrangą.

## 3.1. Mokymosi rezultatas. Naudoti Php programavimo kalbos įrankius ir sintaksę.

*1 užduotis*. ĮDIEKITE PHP INTERPRETATORIŲ.

Atsisiųsti ir įdiegti PHP 7.x interpretatorių.

*2 užduotis*. ĮDIEKITE APACHE WEB SERVERĮ.

Atsisiųsti ir įdiegti Apache2 WEB serverio programinę įrangą. Sukonfigūruoti PHP, kad jis būtų kraunamas, kaip Apache modulis ir per naršyklę vykdyti PHP tipo failuose esantį kodą.

*3 užduotis*. ĮDIEKITE MYSQL DUOMENŲ BAZĖS SERVERĮ.

Atsisiųsti ir įdiegti MySQL duomenų bazės serverį.

*4 užduotis*. PARUOŠKITE PROGRAMAVIMO APLINKĄ IR KURKITE PROJEKTĄ.

Įdiegti teksto redaktorių(Sublime Text, Atom arba kt.) skirtą PHP kodo rašymui. Sukurti projekto aplanką su norima struktūra viduje, bei failu index.php. Naudojantis echo funkcija išvesti pasirinktą tekstą naršyklėje.

*5 užduotis*. PANAUDOKITE PHP KOMANDINĖJE EILUTĖJE.

Konsolėje panaudojant php komandą ir echo funkciją išvesti teksto eilutę: „PHP interpretatorius veikia“ .

*6 užduotis*. IŠBANDYKITE PHP BUILT-IN SERVERĮ.

UŽDUOTIES SĄLYGA: konsolėje sukurkite aplanką. Aplanko viduje sukurkite failą index.php su žemiau nurodytu kodu:

<?php

echo "PHP built-in WEB serveris veikia;"

?>

Paleiskite built-in serverį.

Konsolėje išvestas tekstas: "PHP built-in WEB serveris veikia“

*7 užduotis*. IŠVESKITE MOKINIO DUOMENIS NARŠYKLĖJE.

Sukurti skirtingų duomenų tipų kintamuosius, priskirti reikšmes ir juos išvesti naršyklėje pagal žemiau pateiktą pavyzdį:

Mokinio duomenys:

Vardas: "Jonas"

Pavardė: "Jonaitis"

Amžius: 39

Pomėgiai:

1. Programuoti

2. Žvejoti

3. Keliauti

*8 užduotis*. SUKURKITE MASYVĄ REMIANTIS *7 užduotyje* PATEIKTU PAVYZDŽIU, SUKURTI MASYVĄ, KURIS TALPINA 3 MOKINIŲ DUOMENIS SU SKIRTINGOMIS REIKŠMĖMIS.

*9 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT FOR CIKLĄ PANAUDOJANT *8 užduotyje* SUKURTO MASYVO DUOMENIS .

*10 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT WHILE CIKLĄ PANAUDOJANT *8 užduotyje* SUKURTO MASYVO DUOMENIS.

*11 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT DO WHILE CIKLĄ PANAUDOJANT *8 užduotyje* SUKURTO MASYVO DUOMENIS.

*12 užduotis*. SUKURKITE ASOCIATYVŲ MASYVĄ : *8 užduotyje* SUKURTĄ MASYVĄ PAKEISKITE Į MULTIDIMENSINĮ ASOCIATYVŲ , KUR MASYVO ELEMENTO RAKTAI MOKINIO YPATYBIŲ PAVADINIMAI.

*13 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT FOREACH CIKLĄ REMIANTIS *7 užduotyje* PATEIKTU PAVYZDŽIU, SUKURKITE ASOCIATYVŲ MASYVĄ, KURIS TALPINA 3 MOKINIŲ DUOMENIS SU SKIRTINGOMIS REIKŠMĖMIS.

Duomenis išvesti HTML sąraše naudojant foreach ciklą.

*14 užduotis*. IŠVESKITE MOKINIO DUOMENIS NARŠYKLĖJE NAUDOJANT FUNKCIJĄ REMIANTIS *7 užduotyje* PATEIKTU PAVYZDŽIU, SUKURKITE ASOCIATYVŲ MASYVĄ, KURIS TALPINA 3 MOKINIŲ DUOMENIS SU SKIRTINGOMIS REIKŠMĖMIS.

Sukurti funkciją, kuri surikiuoja (pagal mokinio kodą didėjančia tvarka) ir išveda mokinių duomenis HTML sąraše. Masyvas funkcijai paduodamas kaip parametras ją kviečiant.

*15 užduotis*. APRAŠYKITE DUOMENIS MASYVE IR IŠVESKITE HTML LENTELĖJE.

Aprašyti masyvą, kuriame saugoma informacija apie mokinius (mokinių skaičius 5 ), pagal žemiau pateiktą pavyzdinę struktūrą:

* Klasė: 5b
* Kodas (turi būti sugeneruotas naudojant rand funkciją)
* Duomenų formavimo data (turi būti šios dienos naudojant date funkciją, lietuvišku formatu)
* Vardas (turi būti transformuotas į didžiąsias raides)
* Pavardė (turi būti transformuota į didžiąsias raides)
* Kontrolinių darbų įvertinimai (apvalinama 2 skaičių po kablelio tikslumu):
* Matematika: 5
* Informacinės technologijos: 9
* Anglų kalba: 10

Duomenis išvesti lentelėje pagal pavyzdį :

Pastaba: HTML elementai neturi būti išvedami naudojant PHP funkcijas (echo, print).

Daugiau informacijos:

<https://www.php.net/manual/en/control-structures.alternative-syntax.php>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Klasė | Kodas | Vardas | Pavardė | Kontrolinių darbų vidurkis | Duomenų įvedimo data |
| 5b: | | | | | |
| Jonas | 123897 | Jonas | Jonaitis | 8,26 | 2019-06-20 |

*16 užduotis*. IŠVESKITE GET METODU GAUTUS DUOMENIS.

UŽDUOTIES SĄLYGA: Išvesti asmens duomenis, kurie buvo gauti GET metodu pagal žemiau pateiktą pavyzdį:

http://localhost/{jusu\_projekto\_pavadinimas}/index.php?vardas=jonas&pavarde=jonaitis&amzius=32metai

Asmens amžių konvertuoti į sveiką skaičių.

*17 užduotis*. IŠVESKITE GET METODU GAUTUS DUOMENIS BE HTML.

UŽDUOTIES SĄLYGA: GET metodu (nuoroda pateikta žemiau) buvo gauti duomenys su HTML kodu. Panaudokite funkciją, kuri HTML kodą konvertuotų į paprastą tekstą.

http://localhost/projektas /index.php?komentaras=<a href="http://www.fake.com">Naujienų portalas</a>

*18 užduotis*. IŠVESKITE POST METODU IŠ FORMOS GAUTUS DUOMENIS.

Sukurkite HTML formą, kurią sudaro laukai:

* Vardas
* El. paštas
* Telefono numeris
* Siūloma kaina
* Žinutė

Užpildžius formos duomenis ir paspaudus mygtuką siųsti, duomenys turi būti išvedami HTML sąraše.

Privalu užtikrinti formos laukų validavimą iš PHP pusės (naudojant regex).

Validacijos taisyklės:

* Vardas – tekstas iki 100 simbolių
* El. paštas – turi atitikti el. pašto formatą
* Telefono numeris - turi atitikti lietuviško telefono numerio formatą su šalies kodu
* Siūloma kaina - turi būti pateikta slankaus kablelio formatu
* Žinutė - tekstas iki 500 simbolių

Jeigu validacijos taisyklės nėra tenkinamos, klaidos pranešimai nurodant klaidingą lauką ar laukus išsaugomi masyve ir išvedami virš formos.

**3.2. Mokymosi rezultatas. Kurti nesudėtingą programinį kodą PHP programavimo kalba.**

*1 užduotis.* APRAŠYKITE ASOCIATYVŲ MASYVĄ, KURIAME SAUGOMA INFORMACIJA APIE AVIAKOMPANIJOS VYKDOMUS SKRYDŽIUS : (NE MAŽIAU KAIP 5 SKRYDŽIAI). Masyvo elementai:

skrydžio numeris, į kur skrenda, kaina.

*2 užduotis. 1 užduotyje* SUKURTĄ MASYVĄ SURIKIUOKITE PAGAL KAINĄ DIDĖJANČIA TVARKA.

*3 užduotis.* SUKURKITE FORMĄ SU ŽEMIAU PATEIKTAIS LAUKAIS:

Skrydžio nr. (pasirenkamos reikšmės, kurios nuskaitomos iš masyvo aprašyto *1 užduotyje*)

asmens kodas

vardas

pavardė

bagažas (pasirenkamos reikšmės (10 kg., 15 kg. , 30 kg, 50 kg.)

pastabos

*4 užduotis*. SUKURKITE FORMOS DUOMENIS, KURIŲ ĮVEDIMAS TURI BŪTI VALIDUOJAMAS PRIEŠ DUOMENIS IŠSIUNČIANT IŠ FORMOS (NAUDOJANT REGEX).

Validacijos taisyklės:

* Skrydžio nr.– privalomas laukas
* Asmens kodas – turi atitikti lietuviško asmens kodo formatą
* Vardas – tekstas iki 100 simbolių
* Pavardė – tekstas iki 100 simbolių
* Bagažas – privalomas laukas
* Komentaras – tekstas ilgis turi būti nuo 50 iki 1000 simbolių

Jeigu validacijos taisyklės nėra tenkinamos, klaidos pranešimai nurodant klaidingą lauką ar laukus išsaugomi masyve ir išvedami virš formos.

*5 užduotis.* UŽPILDYKITE *3 užduotyje* NURODYTĄ FORMĄ IR PASPAUSKITE FORMOS MYGTUKĄ „FORMUOTI UŽSAKYMĄ“ DUOMENIS IŠSAUGANT MASYVE.

Duomenys išvedami HTML lentelėje iš sukurto masyvo formos apdorojimo metu.

Iš formos gauta laukelio bagažas reikšmė konvertuojama į sveiką skaičių.

*6 užduotis.* PAPILDYKITE *1 užduotyje* KODĄ TAIP, KAD PASPAUDUS „FORMUOTI UŽSAKYMĄ“ VISA UŽSAKYMO INFORMACIJA IŠ FORMOS BŪTŲ IŠSAUGOTA TEKSTINIAME FAILE DUOMENYS.TXT.

*7 užduotis.* ĮDIEKITE PAPILDOMŲ BIBLIOTEKŲ DIEGIMUI ĮRANKĮ COMPOSER, KURĮ GALITE ATSISIŲSTI IŠ: <https://getcomposer.org/download/>

*8 užduotis*. PRIDĖKITE NUORODĄ SPAUSDINTI VIRŠ *5 užduotyje* SUGENERUOTOS HTML LENTELĖS SU SKRYDŽIO INFORMACIJA IŠ FORMOS.

Paspaudus nuorodą spausdinti turi būti sugeneruotas pdf failas su visa informaciją apie skrydį iš *6 užduotyje* užpildyto failo: duomenys.txt.

PDF failo generavimui naudokite biblioteką TCPDF, kurią galite įdiegti naudodami composer.

*9 užduotis*. KOMENTUOKITE IR DOKUMENTUOKITE PDF FAILO GENERAVIMO KODĄ IŠ *8 užduoties* NAUDOJANT PHPDOC STANDARTĄ.

# *10 užduotis*. ATLIKITE PROGRAMAVIMO STANDARTŲ TESTĄ.

Pažymėkite, ar pateikti kodo fragmentai atitinka PHP programavimo standartus:

1.

<?php print\_welcome\_message();

a) Taip

b) Ne

2.

<div>

<h1><?php print\_welcome\_message(); ?></h1>

</div>

a) Taip

b) Ne

3.

<div>

<p><?= get\_welcome\_message(); ?></p>

</div>

a) Taip

b) Ne

4.

<?php

/\* Labai svarbus komentaras \*/

a) Taip

b) Ne

5.

<?php

$movies = [

'slumdog\_millionaire' => 'When somebody asks me a question, I tell them the answer.',

'silver\_linings\_playbook' => 'I opened up to you, and you judged me.',

'the\_lives\_of\_others' => 'To think that people like you ruled a country.',

'the\_shawshank\_redemption' => 'Get busy living, or get busy dying.'

];

a) Taip

b) Ne

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

*1 užduotis.*SUKURKITE KLASĘ PAVADINIMU VARTOTOJAS.

Klasė turi turėti savybes: id, vardas, el. paštas, slaptažodis, prieigos teisės. Klasėje turi būti realizuoti metodai: prisijungti ir atsijungti, kurie išveda pranešimą priklausomai nuo atlikto veiksmo. Pvz.: Vartotojui prisijungus išvedamas pranešimas: „ {vartotojo vardas} prisijungė prie sistemos “.

*2 užduotis.*SUKURTOS *1 užduotyje* KLASĖS PAGRINDU SUKURKITE 3 SKIRTINGUS VARTOTOJUS SU KLASĖJE APRAŠYTŲ SAVYBIŲ REIKŠMĖMIS, TAIP PAT REALIZUOKITE KLASĖJE SUKURTUS METODUS.

*3 užduotis*. PAKEISKITE *1 užduotyje* SUKURTOS KLASĖS SAVYBES: JOS NETURI BŪTI SKAITOMOS/PERRAŠOMOS TIESIOGIAI KREIPIANTIS Į JAS KLASĖS IŠORĖJE.

*4 užduotis.*SUKURKITE „SET“ METODĄ PAVADINIMU REGISTRACIJA (*1 užduotyje* SUKURTOJE KLASĖJE) , KURIS INICIJUOTOMS VARTOTOJO OBJEKTO SAVYBĖMS LEISTŲ PRISKIRTI REIKŠMES.

*5 užduotis.*SUKURKITE „GET“ METODĄ (*1 užduotyje* SUKURTOJE KLASĖJE) PAVADINIMU PROFILIS, KURIS VISAS VARTOTOJO SAVYBIŲ REIKŠMES GRĄŽINTŲ KAIP ASOCIATYVŲ MASYVĄ.

*6 užduotis.*SUKURKITE KONSTRUKTORIŲ *(1 užduotyje* SUKURTOJE KLASĖJE ), KURIS LEISTŲ SUKURTI NAUJĄ OBJEKTĄ NURODANT SAVYBIŲ (ID, VARDAS) REIKŠMES.

*7 užduotis.*IŠTAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS.

<?php

class Pasisveikinimas

{

public $public = "Public";

private $private = "Private";

protected $protected = "Protected";

public function sakykLabas

{

echo $this->public;

echo $this->private;

echo $this->protected;

}

}

$start = new Pasisveikinimas();

echo $start->protected;

*8 užduotis.*SUKURKITE KLASĘ PAVADINIMU „RENDER“.

Klasėje turi būti sukurtas statinis metodas pavadinimu „Print“, kuris gautą masyvą išveda kaip HTML lentelę.

*9 užduotis*. *1 užduotyje* SUKURTOS KLASĖS PAGRINDU SUKURKITE NAUJĄ OBJEKTĄ. PRISKIRKITE REIKŠMES VISOMS OBJEKTO YPATYBĖMS. PANAUDOKITE RENDER KLASĖS METODĄ PRINT DUOMENŲ APIE VARTOTOJĄ IŠVEDIMUI.

*10 užduotis*. SUKURDAMI NAUJĄ KLASĘ DARBUOTOJAS PRAPLĖSKITE *1 užduotyje* SUKURTĄ KLASĘ VARTOTOJAS.

Klasė darbuotojas turi paveldėti klasę vartotojas pridedant naujas savybes (pareigos, darbo užmokestis, įdarbinimo data). Sukurkite naują objektą darbuotojas klasės pagrindu ir panaudodami render klasės metodą print išveskite informaciją apie darbuotoją.

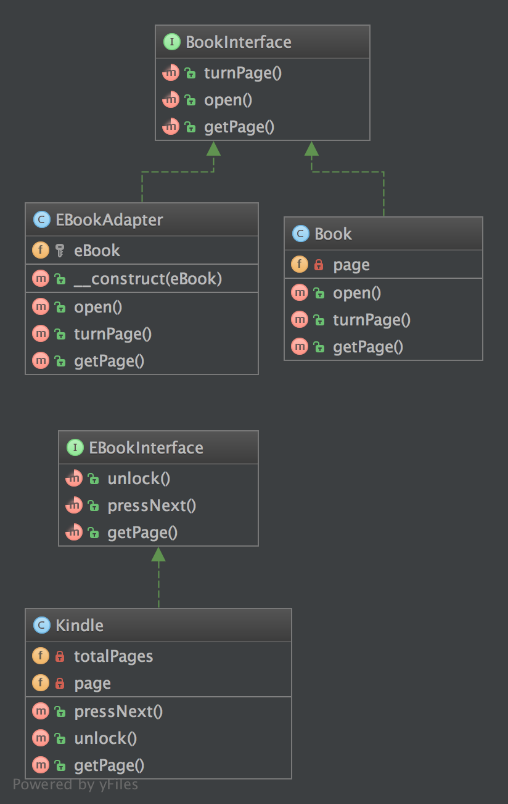
*11 užduotis*. *1 užduotyje* SUKURTĄ KLASĘ VARTOTOJAS PASKELBKITE ABSTRAKČIA.

Klasė turi turėti du abstrakčius metodus: fillportfolio (leidžia papildyti informaciją apie vartotoją), showportfolio (grąžina vartotojo duomenis).

*12 užduotis.*SUKURKITE KLASĘ STUDENTAS SU YPATYBĖMIS: MOKYMOSI PROGRAMA, KURSAS, KURI PAVELDI ABSTRAKČIĄ KLASĘ VARTOTOJAS.

Klasės studentas pagrindu sukurti 3 objektus ir išvesti informaciją.

*13 užduotis.*PAGAL ŽEMIAU PATEIKTĄ KLASIŲ DIAGRAMĄ SUKURKITE KLASES IR INTERFEISUS SU METODAIS.



*14 užduotis*. PAGAL ŽEMIAU PATEIKTĄ KODĄ NUBRAIŽYKITE UML KLASIŲ DIAGRAMĄ.

FormatterInterface.php

<?php

interface FormatterInterface

{

public function format(string $text);

}

PlainTextFormatter.php

<?php

class PlainTextFormatter implements FormatterInterface

{

public function format(string $text)

{

return $text;

}

}

HtmlFormatter.php

<?php

class HtmlFormatter implements FormatterInterface

{

public function format(string $text)

{

return sprintf('<p>%s</p>', $text);

}

}

Service.php

<?php

abstract class Service

{

/\*\*

\* @var FormatterInterface

\*/

protected $implementation;

/\*\*

\* @param FormatterInterface $printer

\*/

public function \_\_construct(FormatterInterface $printer)

{

$this->implementation = $printer;

}

/\*\*

\* @param FormatterInterface $printer

\*/

public function setImplementation(FormatterInterface $printer)

{

$this->implementation = $printer;

}

abstract public function get();

}

HelloWorldService.php

<?php

class HelloWorldService extends Service

{

public function get()

{

return $this->implementation->format('Hello World');

}

}

# *15 užduotis*. ATLIKITE PHP OBJEKTINIO PROGRAMAVIMO ŽINIŲ PATIKRINIMO TESTĄ.

1. Pasirinkite teisingą teiginį:

a) Objektinis programavimas OOP (angl. Object-Oriented Programming) yra programavimo stilius, leidžiantis programuotojui grupuoti duomenis ir užduotis į konteinerius (modulius)

b) Objektinis programavimas OOP yra programavimo stilius, leidžiantis programuotojui grupuoti duomenis ir užduotis į funkcinius karkasus

c) Objektinis programavimas OOP yra programavimo stilius, leidžiantis programuotojui grupuoti duomenis ir užduotis į vidines procedūras

2. Kiek skirtingų objektų galima sukurti klasės pagrindu ?

a) 1

b) 2

c) 3, jeigu tai jungtinė klasė

d) Neribotą skaičių

3. Kas gali būti aprašoma klasėje ?

a) Tik metodai

b) Tik abstraktūs metodai

c) Ypatybės ir metodai

4. Kas yra metodas iš OOP perpsektyvos?

a) tai objekto elgsenos tobulėjimas, kiekvienoje iteracijoje išreiškiamas dirbtinio intelekto vienetu

b) Tai yra veiksmai (funkcijos), kuriuos gali atlikti objektas

c) Tai yra objekto savybių sąsaja su išoriniais interfeisais

d) Teisingo atsakymo nėra

5. Kam naudojamas "magic" metodas \_\_construct ?

a) Tiesinių algoritmų konstravimui

b) Naujo objekto sukūrimui pagal nustatytą tvarką

c) Objekto pašalinimui

d) Tokio metodo nėra

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

*1 užduotis*. ĮDIEKITE PHP KODO TESTAVIMO ĮRANKĮ PHPUNIT IR, NAUDOJANT DOKUMENTACIJĄ, JĮ IŠBANDYKITE.

*2 užduotis.* SUKURKITE KLASĘ SKAIČIUOTUVAS.

Parašykite testą, kuris patikrina naują objekto sukūrimą.

*3 užduotis.* *1 užduotyje* SUKURTAI KLASEI REALIZUOKITE SKAIČIUOTUVO FUNKCIONALUMĄ, KURIS ATLIEKA ARITMETINES OPERACIJAS (+, -, \*, /).

Parašykite testą, kuris patikrina skaičiuotuvo funkcionalumą.

*4 užduotis*. PARAŠYKITE TESTĄ, KURIS PATIKRINA NAUJO VARTOTOJO REGISTRACIJĄ IR JO PROFILIO DUOMENŲ IŠVEDIMĄ 3. Kompetencijos „Kurti tipinę programinę įrangą“ mokymosi rezultato 3.3. „Taikyti objektinio programavimo principus programuojant“ *1 užduotyje* SUKURTOS KLASĖS „VARTOTOJAS“ REALIZUOTAIS METODAIS.

*5 užduotis.* *2 užduotyje* SUKURTAM SKAIČIUOTUVO METODUI, KURIS ATLIEKA SKAIČIAVIMUS,VIETOJ SKAIČIŲ ĮRAŠYKITE TEKSTINIUS SIMBOLIUS.

Įvykus klaidai skaičiuojant, ją apdorokite (naudojant php exception) ir išveskite pranešimą vartotojui, kad panaudoti neleistini parametrai.

*6 užduotis**.* 3. KOMPETENCIJOS„ KURTI TIPINĘ PROGRAMINĘ ĮRANGĄ“ 3.3. MOKYMOSI REZULTATO„ TAIKYTI OBJEKTINIO PROGRAMAVIMO PRINCIPUS PROGRAMUOJANT“ *8 užduotyje* SUKURTAME TEKSTINIAME FAILE ĮRAŠYTI TIK SKRYDŽIUS Į BARSELONĄ.

Jeigu skrydžių saraše yra kitokių skrydžių, juos traktuoti kaip klaidą ir apdoroti (naudojant php exception). Vartotojui turi būti išvestas pranešimas su skrydžiais, kurie neatitiko sąlygos.

*7 užduotis*. APDOROJUS KLAIDAS *6 užduotyje*, SĄLYGOS NEATITIKUSIUS SKRYDŽIUS NAUDOJANT ERROR\_LOG FUNKCIJĄ ĮRAŠYKITE Į TEKSTINĮ FAILĄ: KLAIDOS.LOG.

*8 užduotis*. NAUDODAMI VAR\_DUMP FUNKCIJĄ, IŠVESKITE GAUTUS DUOMENIS IŠ FORMOS (sukurtos *4 užduotyje*).

*9 užduotis*. PAKEISKITE PHP NUSTATYMUS, KAD NEBŪTŲ RODOMI ĮSPĖJIMAI VYKDANT PHP APLIKACIJĄ.

Klaidos turi būti rodomos.

*10 užduotis*. PANAUDOKITE XDEBUG SAVO PROJEKTE HTTP UŽKLAUSŲ ANALIZEI.

1. **Kompetencija. Analizuoti skirtingų tipų reikalavimus, apibūdinančius kompiuterinę programą.**

4. 1. **Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo reikalavimų programinei įrangai formatą.**

*1* *užduotis.* ATLIKITEPROGRAMINĖS ĮRANGOS FORMATŲ REIKALAVIMUS.

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

UŽDUOTIES TIKSLAS: Gebėti atskirti funkcinių reikalavimų formatus bei mokėti apibrėžti testavimo apimtis remiantis funkcinių reikalavimų dokumentais (nepriklausomai nuo formato).

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

**Panaudojimo atvejis Nr. 1**

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

*2* *užduotis.*  ATLIKITE REIKALAVIMŲ PERŽIŪROS PROCESĄ.

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

UŽDUOTIES TIKSLAS: Suprasti, kam reikalinga reikalavimų peržiūra, kada ji yra vykdoma, kokie yra peržiūros privalumai ir naudos komandai.

**Panaudojimo atvejis Nr. 2**

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

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

*1* *užduotis.* PATIKRINKITE ŽINIAS APIE FUNKCINIUS REIKALAVIMUS.

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

UŽDUOTIES TIKSLAS: Išmokti naudoti funkcinius reikalavimus testuojant programinę įrangą. Išmokti nustatyti neatitikimus tarp reikalavimuose apibrėžtų ir realizuotų sistemos funkcijų.

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

Opencart – Paieška (supaprastinti paieškos funkcionalumo reikalavimai)

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

- jei įvedamas raktažodis pagal produkto kategoriją ir spaudžiamas "Paieška" mygtukas, sistema pateikia atitinkamos kategorijos produktų sarašą ;

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

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

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

- Paieškos rezultatai yra surūšiuoti abėcėlės tvarka.

*2* *užduotis.* PATIKRINTI ŽINIAS APIENEFUNKCINIUS(TECHNINIAI, SAUGUMO) REIKALAVIMUS.

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

UŽDUOTIES TIKSLAS: Gebėti skirti funkcinius sistemos reikalavimus nuo nefunkcinių. Suprasti ir gebėti paaiškinti nefunkcinių reikalavimų rūšis.

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

|  |
| --- |
| Atsakymas: |

1. Pateikite bent penkis nefunkcinių reikalavimų testavimo tipus.

|  |
| --- |
| Atsakymas: |

1. Kodėl svarbu žiniatinklio programas testuoti su skirtingomis naršyklėmis?

|  |
| --- |
| Atsakymas: |

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

|  |
| --- |
| Atsakymas: |

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

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

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

|  |
| --- |
| Atsakymas: |

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

|  |
| --- |
| Atsakymas: |

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

1. Ką tikrina suderinamumo testavimas? (pasirinkite geriausiai tinkanti apibūdinimą)
   1. Ar kuriama programa yra suderinama su duomenų baze, operacine sistema, kita programine bei technine įranga
   2. Ar kuriama programa gali dirbi sinchronizuotai su keliomis duomenų bazėmis vienu metu
   3. Ar kuriama programa gali pilnavertiškai veikti tiek su Windows, tiek su Linux operacinėmis sistemomis
   4. Ar kuriama programa nelūžta po atnaujinimo

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

# 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.

- Poetu gali būti visi norintieji, ne vien tik užfiksavę savo poeziją sistemoje arba jau sukūrė poezijos kūrinį.

- Poezijos kūrinius galima pateikti internetu, el. paštu arba popieriuje.

- Visi sistemoje užfiksuoti poezijos kūriniai parašyti poetų, kurių duomenys jau užfiksuoti sistemoje. Nepateikęs visos reikalaujamos informacijos, poetas negalės pateikti ir išsaugoti savo poezijos sistemoje.

- Leidinį gali sudaryti ir vienas eilėraštis, ir poezijos rinkinys, ir literatūros kritikos kūrinys.

- Leidiniai parduodami klientams, apie kuriuos sistemoje yra informacijos. Anonimiški pirkėjai neaptarnaujami.

- Vieno pardavimo metu galima parduoti ir vieną leidinį, ir kelis. Tačiau jei parduodama keliems klientams, leidykla tai traktuoja kaip pardavimą kelis kartus. Kiekvienas klientas aptarnaujamas atskirai.

- Parduodami ne visi leidiniai. Sunkiau parduoti, pavyzdžiui, kokius nors ypatingus leidimus. O kai kurių leidinių nepavyksta parduoti nei egzemplioriaus.

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

# 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.

*3 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?
18. Punkto Nr. 17 užklausą pataisykite taip, kad rodytų tik tuos projektus, kur dirba bent 3 darbuotojai.

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

*1 užduotis.*SUKURTI PHP (OBJEKTIŠKAI ORIENTUOTĄ) APLIKACIJĄ SKIRTĄ VARTOTOJO UŽDUOČIŲ SAUGOJIMUI, RODYMUI, ŠALINIMUI IR ATNAUJINIMUI.

* Aplikacijos aplankų ir failų struktūra turi būti tvarkinga atskirianti aplikacijos loginę dalį nuo prezentacinės.
* Klasių realizacijai turi būti naudojamas PSR-4 (autoloader).
* Privalomos klasės:
* Database (connect metodas, skirtas prisijungimui prie duomenų bazės)
* QueryBuilder (metodai skirti vykdyti DB užklausas (select, insert, delete, update)
* Duomenys saugomi MySQL duomenų bazėje *TODO*.
* Duomenų bazėje turi būti sukurta lentelė: *tasks*, kurios struktūra pateikta žemiau.
* Prisijungimo prie duomenų bazės duomenys turi būti saugomi atskirame faile.
* PHP sąsajai su duomenų base turi būti naudojama PHP Data Objects (PDO).
* Vartotojo sąsaja turi būti tvarkinga ir leidžianti vartotojui lengvai naudoti aplikacija.

DB lentelės tasks struktūra:

CREATE TABLE `tasks` (

`id` int(11) NOT NULL,

`name` varchar(255) COLLATE utf8mb4\_bin NOT NULL,

`description` varchar(255) COLLATE utf8mb4\_bin NOT NULL,

`isComplete` tinyint(1) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_bin;

# Modulis „Programavimo aplinkos ir kūrimo proceso valdymas (Php)“

# 1. Kompetencija. Naudoti tarnybinių stočių operacines sistemas.

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

*1 užduotis.* SUSIPAŽINTI SUDEBESŲ 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.* SUSIPAŽINTI SU SERVERIO IŠTEKLIAIS.

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.* SUSIPAŽINTI SU LINUX OS PARINKIMU.

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.* SUSIPAŽINTI SU XUBUNTU OS DIEGIMU 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.* SUSIPAŽINTI SU XUBUNTU OS KONFIGŪRAVIMU.

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.* DIRBTI 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*. ADMINISTRUOTI LINUX OS PAKETUS.

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.* DIEGTI IR KONFIGŪRUOTI WEB SERVERĮ.

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

Naudojamos priemonės (aplinkos): komandų eilutė.

Veiksmų eiga (žingsniai), kaip bus siekiama rezultato:

1. Paruoškite WEB serverį darbui:
   1. Įdiekite WEB serverį Apache.
   2. Paleiskite WEB serverį Apache.
   3. Patikrinkite jo veikimą.
   4. Jei reikia, WEB serverį sukonfigūruokite.
2. Sukonfigūruokite WEB serverį taip, kad jis galėtų talpinti 2 svetaines:
   1. 1-ma svetainė: linux.org. Svetainė turi talpinti tekstą: „Sveiki aš esu Linux OS!“ ir turėti nuorodą į svetainę windows.org (index.html – naudojame tik HTML ir CSS)
   2. 2-ra svetainė: windows.org. Svetainė turi talpinti tekstą: „Sveiki aš esu Windows OS!“ ir turėti nuorodą į svetainę linux.org (index.html – naudojame tik HTML ir CSS)
3. Paleiskite abi svetaines ir patikrinkite jų veikimą.
4. Įjungite savo WEB serveryje PHP palaikymą, parašykite trumpą php testą (<?php phpinfo();?>) ir patikrinkite ar PHP palaikymas veikia.

# Mokymosi rezultatas. Valdyti tarnybines stotis per nuotolinę prieigą.

*1 užduotis.* PRISIJUNGTI PRIE LINUX OS NUOTOLINIU BŪDU.

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.

# 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.

# 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)

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



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

*1 užduotis.*ĮSIDIEKITE PHP INTEGRUOTA PROGRAMAVIMO APLINKĄ PHP STORM.

*2 užduotis*. PALEISKITE PHP STORM IR SUKURKITE NAUJĄ PROJEKTĄ.

*3 užduotis*. IDE SUKONFIGŪRUOKITE INTERPRETATORIŲ, KAD KODO LANGE PASPAUDUS NARŠYKLĖS IKONĄ, JIS BŪTŲ ĮVYKDYTAS JOJE.

*4 užduotis*. SUKONFIGŪRUOKITE GIT KODO VERSIJAVIMO SISTEMĄ IDE.

*5 užduotis*. NAUDOJANT TERMINALO LANGĄ, ESANTĮ IDE, PANAUDOKITE COMPOSER PHP MAILER BIBLIOTEKOS ĮDIEGIMUI PROJEKTE.

# 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.

* 1. **Mokymosi rezultatas. Vykdyti programinio kodo versijavimą naudojant programinio kodo versijavimo įrankius, tinkamus Php kalbai.**

*1 užduotis*. TEORINIŲ ŽINIŲ APIE VERSIJAVIMĄ TESTAS.

1. Kodo versijavimo sistemos:

a) CVS, CMS, MVC

b) MVC,CVS, SCM, GIT

c) SVN, GIT, CVS, SCM

2. Ar sutinkate su teiginiu, kad GIT centralizuota kodo versijavimo sistema ?

a) Taip

b) Ne

3. Naudojant GIT kodo versijavimo sistemą kodas saugomas:

a) Lokalioje repositorijoje

b) Nuotolinėje repositorijoje

c) Nuotolinėje ir lokalioje repositorijose

d) Nuotolinėje, lokalioje ir file shared serveryje

*2 užduotis*. PARUOŠKITE GIT VERSIJAVIMO ĮRANKĮ.

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

*4 užduotis*. SUKURKITE NAUJĄ PROJEKTO APLANKĄ .

Jame sukurkite lokalią repositoriją.

*5 užduotis.*PROJEKTO APLANKE PRIDĖKITE FAILĄ INDEX.PHP.

Jame išveskite šios dienos datą. Pridėkite sukurtą failą į repositoriją. Užfiksuokite pakeitimus kodo versijavimo sistemoje su prasminga informacija apie pridėtą kodą.

*6 užduotis*. PROJEKTO APLANKE PRIDĖKITE FAILĄ INDEX.PHP.

Jame išveskite šios dienos datą. Pridėkite sukurtą failą į repositoriją. Užfiksuokite pakeitimus kodo versijavimo sistemoje su prasminga informacija apie pridėtą kodą.

*7 užduotis*. PATIKRINKITE GIT ŽURNALE AR *6 užduotyje* DARYTI PAKEITIMAI BUVO UŽFIKSUOTI.

*8 užduotis*. INDEX.PHP FAILE PARAŠYKITE KODĄ, KURĮ ĮVYKDŽIUS BUS IŠVESTAS DIENŲ SKAIČIUS IKI PROGRAMUOTOJŲ DIENOS.

Užfiksuokite pakeitimus kodo versijavimo sistemoje su prasminga informacija apie pridėtą kodą.

*9 užduotis.*INDEX.PHP FAILE PARAŠYKITE KODĄ, KURĮ ĮVYKDŽIUS BUS IŠVESTAS DIENŲ SKAIČIUS IKI KALĖDŲ.

Užfiksuokite pakeitimus kodo versijavimo sistemoje su prasminga informacija apie pridėtą kodą.

*10 užduotis*. NAUDOJANT KODO VERSIJAVIMO SISTEMOS PRIEMONES, SUGRĄŽINKITE *8* *užduoties* KODO VERSIJĄ.

*11 užduotis.*PATIKRINKITE, KURIOJE ŠAKOJE (BRANCH) ŠIUO METU ESATE.

*12 užduotis*. SUKURKITE NAUJĄ ŠAKĄ PAVADINIMU BIRHDAY.

Pereikite į naujai sukurtą šaką (branch). Jame parašykite kodą, kurį įvykdžius – programos vartotojui leistų įvesti vardą, gimimo dieną ir pagal įvestus duomenis apskaičiuotų dienų skaičių iki gimtadienio.

*13 užduotis*. ŠAKĄ (BRANCH) BIRHDAY SUJUNKITE SU ŠAKA (BRANCH) MASTER.

*14 užduotis*. PAŠALINKITE BIRHDAY ŠAKĄ (BRANCH).

*15 užduotis*. GITHUB.COM SUKURKITE NUOTOLINĘ REPOSITORIJĄ.

*16 užduotis*. PRIE SAVO PROJEKTO PRIDĖKITE NUOTOLINĘ REPOSITORIJĄ IR IŠSIŲSKITE VISUS KODO PAKEITIMUS Į JĄ.

*17 užduotis.* GITHUB.COM SURASKITE JUMS PATINKANTĮ PHP PROJEKTĄ IR JĮ NUKLONUOKITE.

**Modulis „Įvadas į darbo rinką“**

## TESTAS ĮSIVERTINTI GEBĖJIMAMS BAIGUS PROGRAMĄ

1. PHP - tai:

a) Paprastojo hiperteksto puslapis, grįstas vidiniu preprocesoriumi

b) Hiperteksto žymėjimo kalba

c) Hiperteskto preprocesorius

d) Hiperteskto perdavimo protokolas

2. Kuris iš komentarų tipų naudojamas PHP

a) // komentuotas kodas iki eilutės pabaigos

b) /\* komentuotas kodas \*/

c) #komentuotas kodas iki eilutės pabaigos

d) Visi aukščiau paminėti

3. Kaip pasiekti masyvo reikšmę elemento reikšmę d?

$a = [

'a',

3 => 'b',

1 => 'c',

'd'

];

a) $a[0]

b) $a[1]

c) $a[2]

d) $a[3]

e) $a[4]

4. Kas bus išvesta:

<?php

$a = 1;

function Test()

{

echo "a = $a";

}

Test();

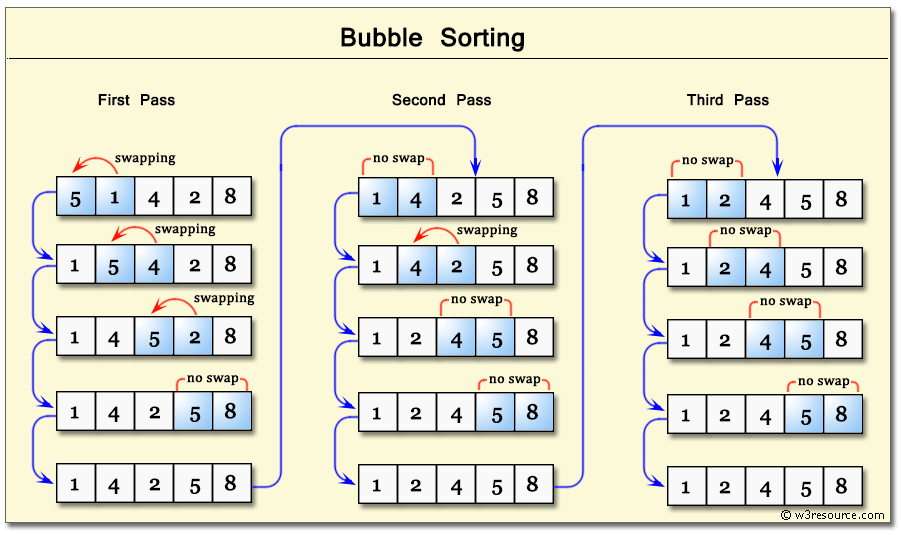
a) 1

b) 2

c) 3

d) nebus išvesta jokia reikšmė

5. Koks rikiavimo algoritmas pavaizduotas?



a) išrinkimo

b) burbulo

c) įterpimo

6. Koks šablonas naudojamas žemiau pateiktame kode?

<?php

class Automobile

{

private $vehicleMake;

private $vehicleModel;

public function \_\_construct($make, $model)

{

$this->vehicleMake = $make;

$this->vehicleModel = $model;

}

public function getMakeAndModel()

{

return $this->vehicleMake . ' ' . $this->vehicleModel;

}

}

class AutomobileCompany

{

public static function create($make, $model)

{

return new Automobile($make, $model);

}

}

$veyron = AutomobileCompany::create('Bugatti', 'Veyron');

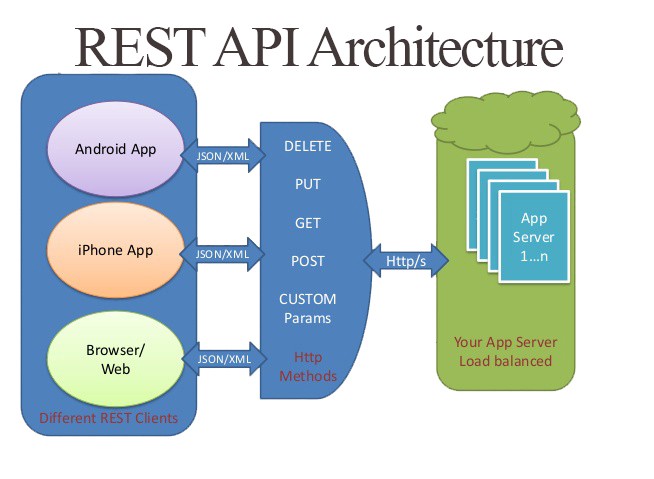
print\_r($veyron->getMakeAndModel());

a) Singleton

b) Factory Method

c) Observer

7. Kokia architektūra yra pavaizduota?



a) Kliento - Serverio

b) REST API

8. Kaip paveldinčioje klasėje pasiekti tėvinis klasės konstruktorių:

a) parent::construct()

b) parent:\_\_construct()

c) parent:construct()

d) parent::\_ \_construct()

9. Kiek parametrų, kaip argumentų galima naudoti PHP funkcijoje:

a) 64

b) 60

c) 68

c) 72

10. Jeigu klasė gali paveldėti kitą klasę, tai vadinama ?

a) Poliformizmu

b) Abstrakcija

c) Inkapsuliacija

11. Kuri funkcija naudojama masyvo konvertavimui į eilutę (string)?

a) exclude

b) implode

c) glue

12. Kuris funkcija skirta failo skaitymui?

a) fopen

b) fread

c) fileREad

13. Kuri funkcija naudojama masyvo elementų kovertavimui į JSON struktūrą?

a) array\_to\_son

b) json\_convert

c) json\_encode

14. Inkapsuliacija tai:

a) galimybė programuoti nežinant konkrečių detalių

b) informacijos slėpimas

c) objektų organizavimas specializuojant egzistuojančius bendresnius tipus

15. $\_POST tai:

a) masyvas

b) objektas

c) konstanta

16. Koks testavimo karkasas yra dažniausiai naudojamas testuojant PHP aplikacijas?

a) PUnit

b) PHPUnit

c) Selenium

17. Ar galima sukurti PHP aplikaciją, kuri operacinės sistemos log įrašus saugotų MySQL duomenų bazėje ir būtų naudojama terminale ?

a) Taip

b) Ne

18. Kuri komanda sukuria lokalią GIT repositoriją?

a) git new repo

b) git init

c) git checkout

19. Kuri GIT komanda parodo visus užfiksuotus pakeitimus?

a) git commit log

b) git log

c) git stash

20. Kuri GIT komanda parodo skirtumus tarp dviejų failų ?

a) git diff

b) git branch

c) git chech diff

21. Naudojant PHP objektinio programavimo principus realizuokite metodus, kurie rūšiuoja masyvo elementus naudojant rūšiavimo algoritmus pateiktus žemiau:

* Išrinkimo
* Įterpimo
* Burbulo
* Greito rikiavimo
* Sąlajos
* Skaitmeninį

<?php

/\*\*

\* Implementuoti rūšiavimo algoritmus.

\*

\* @method static array bubble(array $input) Burbulo rūšiavimas

\* @method static array selection(array $input) Išrinkimo rūšiavimas

\* @method static array insertion(array $input) Įterpimo rūšiavimas

\* @method static array merge(array $input) Apjungimo rūšiavimas

\* @method static array quick(array $input) Burbulo rūšiavimas

\* @method static array radix(array $input) Skaitmeninis (radix) rūšiavimas

\*/

final class Sort

{

}

Rūšiavimo metodus padengti Unit testais.

# Užduočių atsakymai

# Modulis „Įvadas į profesiją“

# *TESTO ATSAKYMAI*

1. b)

2. c)

3. b)

4. c)

5. b)

6. c)

7. b)

8. b)

9. c)

10. b)

11. b)

12. b)

13. c)

14. c)

15. b)

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

# Kompetencija. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.

* 1. **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>

2.2. Plačiau.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

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

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

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

</head>

<body>

<h2>Kelionės</h2>

<ul>

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

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

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

</ul>

<h3>Polsinės kelionės</h3>

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

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

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

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

<h3>Išvykimo laikai</h3>

<ul>

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

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

</ul>

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

<ol>

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

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

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

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

maitinimo tipu.</li>

</ol>

<a href="poilsines.html">Grįžti</a>

</body>

</html>

2.3. Poilsinės.html

<!DOCTYPE html>

<html>

<head>

<title>www praktinis darbas</title>

<meta charset="UTF-8" />

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

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

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

</head>

<body>

<h2>Kelionės</h2>

<ul>

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

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

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

</ul>

<h3>Poilsinės kelionės</h3>

<h4>Ispanija, Kosta Dorada</h4>

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

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

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

<h4>Kreta</h4>

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

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

</body>

</html>

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

Rezultatai:

* 1. Index.html

<!DOCTYPE HTML>

<html lang="lt">

<head>

<meta charset="utf-8" />

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

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

</head>

<body>

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

<h2>Pagrindinės temos</h2>

<ul>

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

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

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

</ul>

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

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

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

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

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

<ol>

<li>Lavina emocinį intelektą</li>

<li>Skatina kurti</li>

<li>Ugdo kantrybę</li>

<li>Lavina smulkiąją motoriką</li>

</ol>

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

<h2 id="rusys">Skrebinimo rūšys</h2>

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

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

<h3>Tradicinis skrebas</h3>

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

<h3>Project Life skrebas</h3>

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

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

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

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

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

<ul>

<li>Nuotraukos</li>

<li>Žirklės</li>

<li>Popierius</li>

<li>Klijai</li>

<li>Ir pagrindinis - VAIZDUOTĖ!</li>

</ul>

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

</body>

</html>

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

**šablonus ir karkasus.**

*1 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;

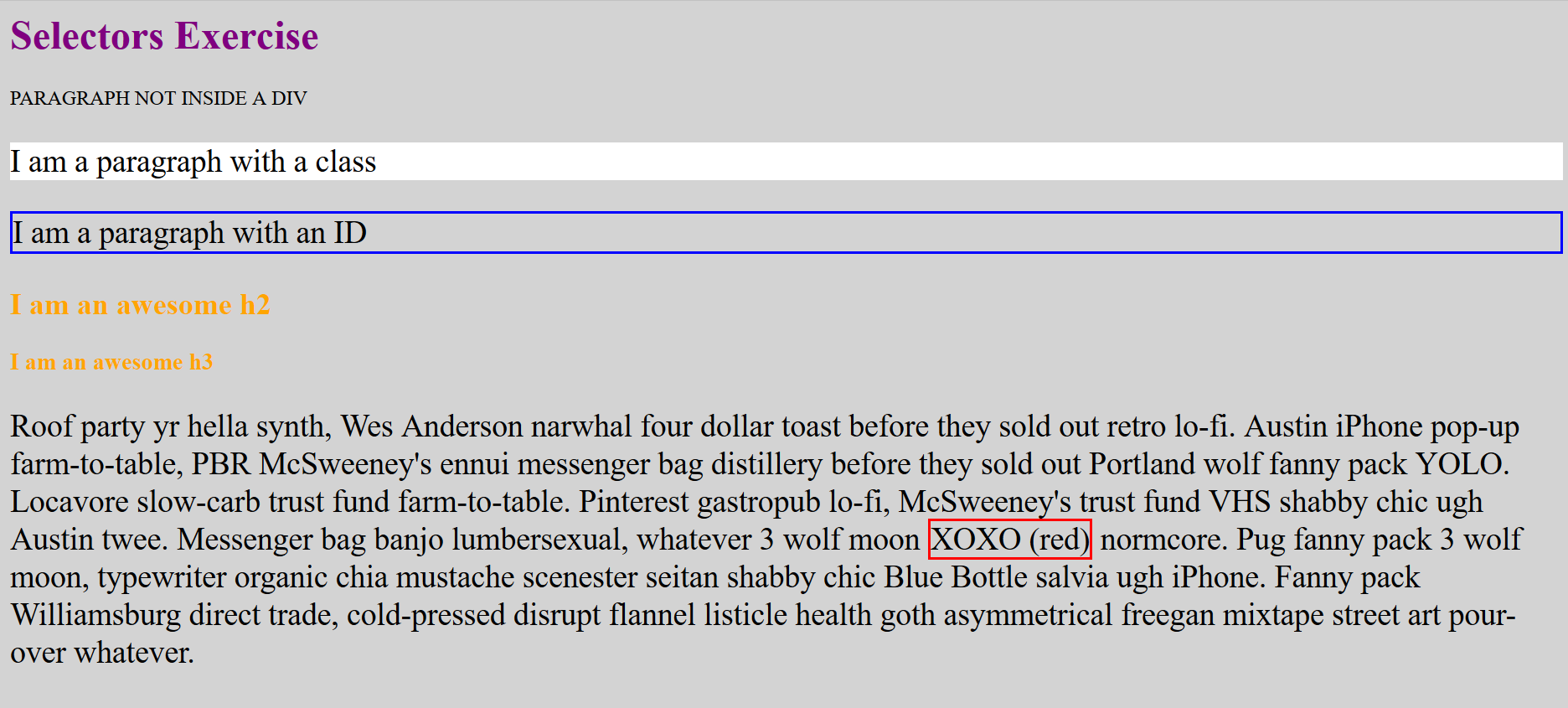
}

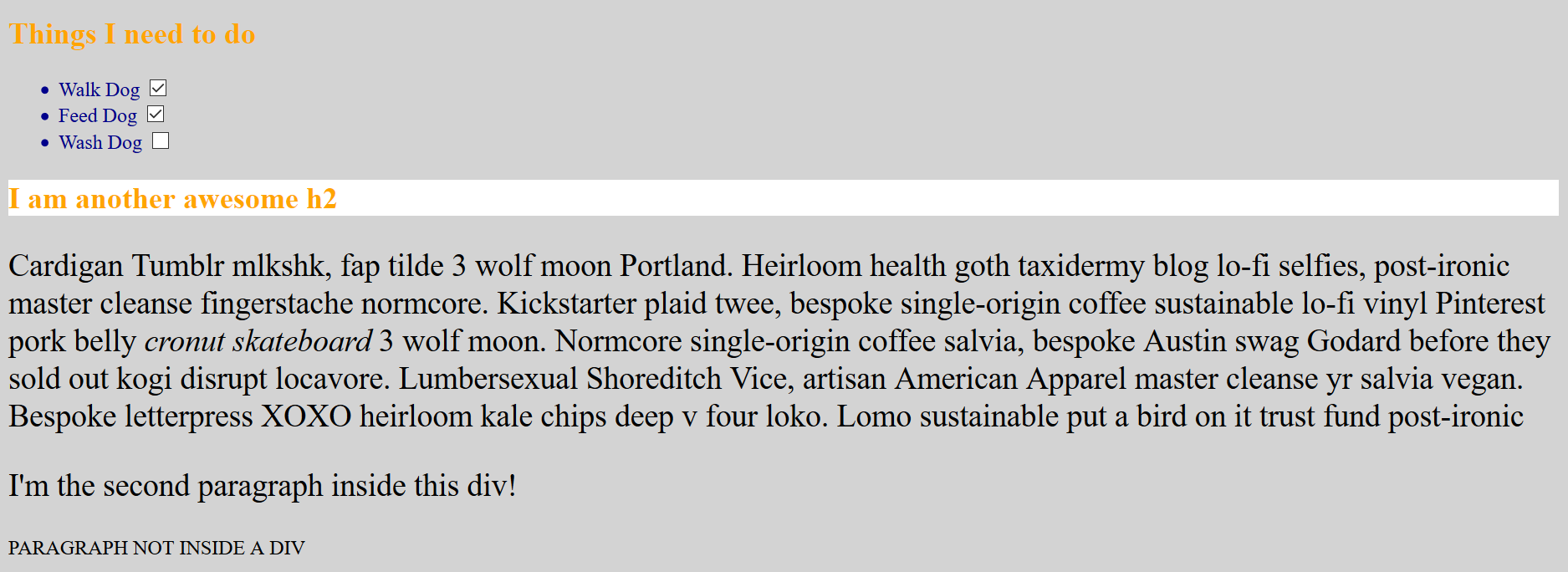
div > ul{

    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;

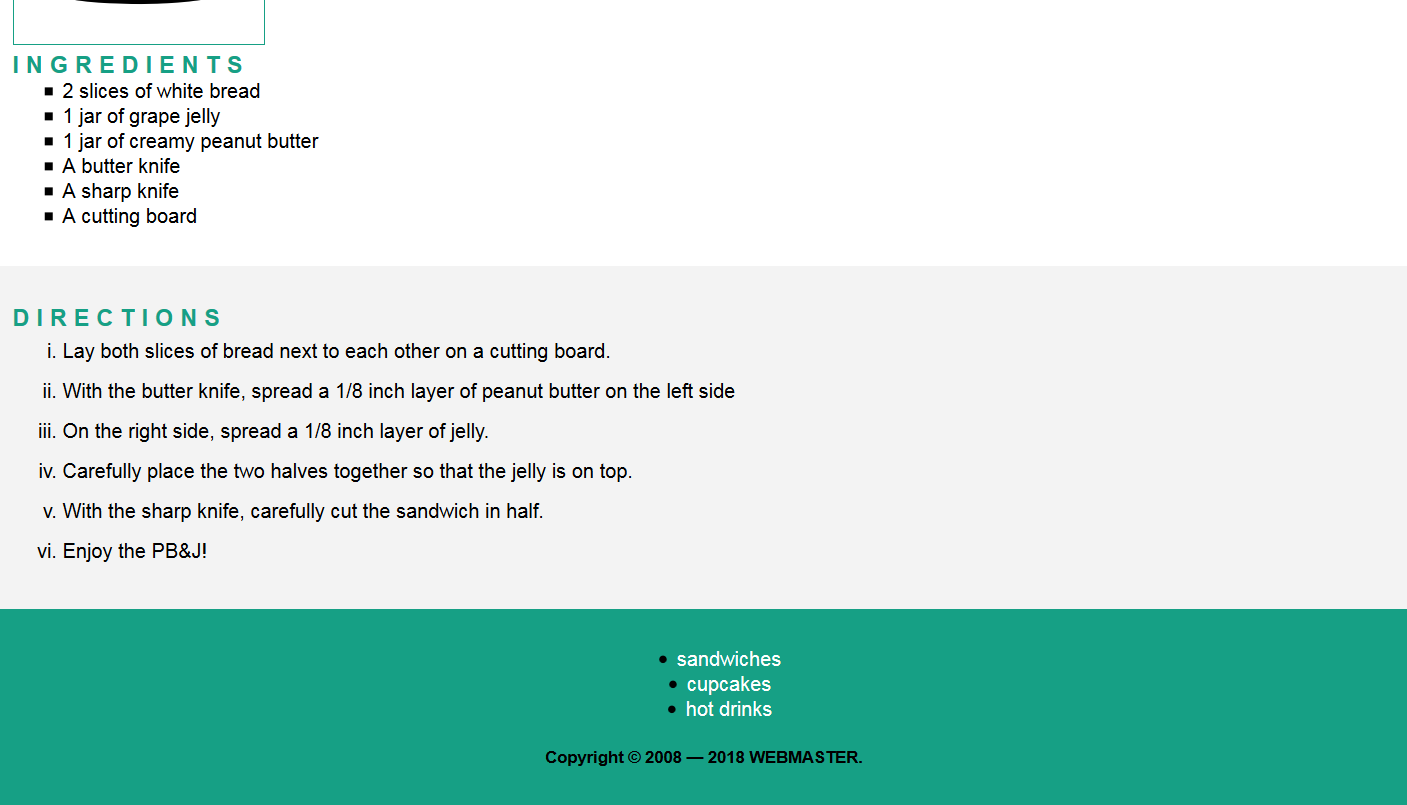
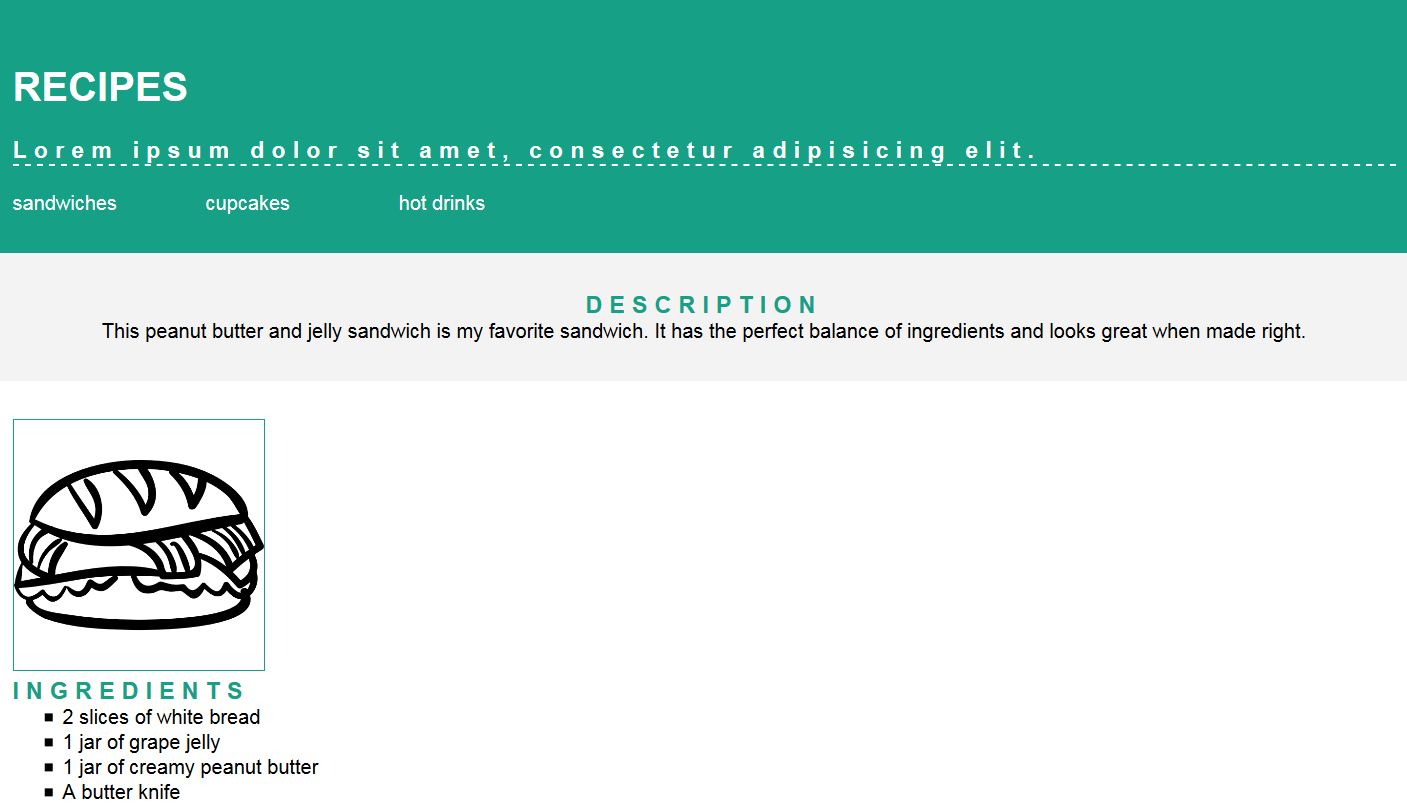
\*/

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. **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>

2. Script.js failas:

(function() {

function buildQuiz() {

// we'll need a place to store the HTML output

const output = [];

// for each question...

myQuestions.forEach((currentQuestion, questionNumber) => {

// we'll want to store the list of answer choices

const answers = [];

// and for each available answer...

for (letter in currentQuestion.answers) {

// ...add an HTML radio button

answers.push(

`<label>

<input type="radio" name="question${questionNumber}" value="${letter}">

${letter} :

${currentQuestion.answers[letter]}

</label>`

);

}

// add this question and its answers to the output

output.push(

`<div class="question"> ${currentQuestion.question} </div>

<div class="answers"> ${answers.join("")} </div>`

);

});

// finally combine our output list into one string of HTML and put it on the page

quizContainer.innerHTML = output.join("");

}

function showResults() {

// gather answer containers from our quiz

const answerContainers = quizContainer.querySelectorAll(".answers");

// keep track of user's answers

let numCorrect = 0;

// for each question...

myQuestions.forEach((currentQuestion, questionNumber) => {

// find selected answer

const answerContainer = answerContainers[questionNumber];

const selector = `input[name=question${questionNumber}]:checked`;

const userAnswer = (answerContainer.querySelector(selector) || {}).value;

// if answer is correct

if (userAnswer === currentQuestion.correctAnswer) {

// add to the number of correct answers

numCorrect++;

// color the answers green

answerContainers[questionNumber].style.color = "lightgreen";

} else {

// if answer is wrong or blank

// color the answers red

answerContainers[questionNumber].style.color = "red";

}

});

// show number of correct answers out of total

resultsContainer.innerHTML = `${numCorrect} out of ${myQuestions.length}`;

}

const quizContainer = document.getElementById("quiz");

const resultsContainer = document.getElementById("results");

const submitButton = document.getElementById("submit");

const myQuestions = [

{

question: "The Bootstrap grid system is based on how many columns?",

answers: {

a: "6",

b: "8",

c: "12"

},

correctAnswer: "c"

},

{

question: "Which class is used to create a big box for calling extra attention?",

answers: {

a: ".bigbox",

b: ".jumbo",

c: ".jumbotron"

},

correctAnswer: "c"

},

{

question: "Bootstrap is mobile-first",

answers: {

a: "true",

b: "false",

},

correctAnswer: "a"

}

];

// display quiz right away

buildQuiz();

// on submit, show results

submitButton.addEventListener("click", showResults);

})();

3. Style.css failas:

body{

font-size: 20px;

font-family: sans-serif;

color: #333;

}

.question{

font-weight: 600;

}

.answers {

margin-bottom: 20px;

}

.answers label{

display: block;

}

#submit{

font-family: sans-serif;

font-size: 20px;

background-color: #279;

color: #fff;

border: 0px;

border-radius: 3px;

padding: 20px;

cursor: pointer;

margin-bottom: 20px;

}

#submit:hover{

background-color: #38a;

}

*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");

}

});

3. 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>

2. 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.

**2.1. 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:

1. Lietuvos statistikos departamentas (2019). *Ekonominės veiklos rūšių klasifikatorius*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://osp.stat.gov.lt/static/evrk2.htm>

*3 užduotis.* SURAŠYKITE VISAS JUMS ŽINOMAS SKAIČIAVIMO SISTEMAS.

(POZICINĖS)

Dešimtainė, dvejetainė, aštuntainė, šešioliktainė, N-ainė skaičiavimo sistemos. Papildomai galite paskaityti:

1. 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:

1. 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.

5 minutės tai 300 sekundžių. 600MB / 300 sekundžių = 2MB/s arba 16Mbps.

**2.2. 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)**

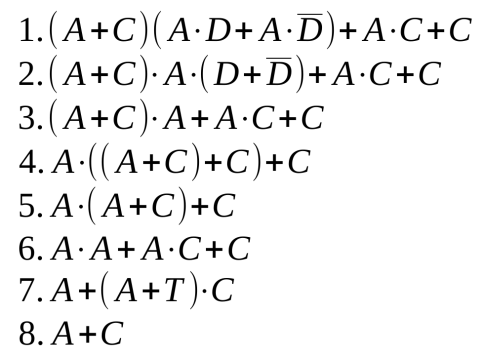
Skaitoma: “Netiesa, kad p ir q ekvivalentiška tam, kad ne-p arba ne-q”. Interpretacija. Pasižymime:

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.

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

public class IterativeAlgorithm {

public static void main(String[] args) {

int[] intArray = {54, 17, 1, 21, 85, 3, 18, 42};

int result = 0;

for(int i = 0; i < intArray.length; i++){

result += intArray[i];

}

System.out.println("Sum of the array elements: " + result);

System.out.println("Average of the array elements: " + (double) result / intArray.length);

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

public class RecursiveAlgorithm {

public static void main(String[] args) {

int[] intArray = {27, 32, 19, -7, 41, 38, 99, -36, 54, 17, 1, 21, 85, 3, -18, 42};

int arrayLength = intArray.length;

System.out.println(findMinNumber(intArray, arrayLength));

}

private static int findMinNumber(int[] intArray, int arrayLength) {

if (arrayLength == 1) {

return intArray[0];

} else {

return Math.min(intArray[arrayLength - 1],

}

findMinNumber(intArray, arrayLength - 1));

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.math.BigInteger;

public class JavaDynamicProgrammingFibonacci {

public static void main(String[] args) {

int n = 120;

System.out.println(fibonacciNumberCounting(n));

}

private static BigInteger fibonacciNumberCounting(int n) {

BigInteger a = BigInteger.valueOf(0);

BigInteger b = BigInteger.valueOf(1);

BigInteger c = BigInteger.valueOf(1);

for (int j = 2; j <= n; j++) {

c = a.add(b);

a = b;

b = c;

}

return c;

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

public class JavaDynamicProgrammingSubsetSum {

public static void main(String[] args) {

int[] set = {3, 34, 4, 12, 5, 2};

int sum = 170;

int setLength = set.length;

if (isSubsetSum(set, setLength, sum)) {

System.out.println("Found a subset with given sum");

} else {

System.out.println("No subset with given sum");

}

}

private static boolean isSubsetSum(int[] set, int n, int sum) {

boolean[][] subset = new boolean[sum + 1][n + 1];

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

subset[0][i] = true;

}

for (int i = 1; i <= sum; i++) {

subset[i][0] = false;

}

for (int i = 1; i <= sum; i++) {

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

subset[i][j] = subset[i][j - 1];

if (i >= set[j - 1]) {

subset[i][j] = subset[i][j] || subset[i - set[j - 1]][j - 1];

}

}

}

return subset[sum][n];

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.util.Arrays;

public class JavaQuickSort {

private int[] array;

private int length;

public static void main(String[] args) {

JavaQuickSort sorter = new JavaQuickSort();

int[] arrayOne = {24, -2, 45, 20, 56, -75, 2, 56, 99, 53, -12};

System.out.println("Unsorted array: \n" + Arrays.toString(arrayOne));

System.out.println("Sorted array: ");

sorter.sort(arrayOne);

for (int i : arrayOne) {

System.out.print(i);

System.out.print(", ");

}

}

private void sort(int[] inputArr) {

if (inputArr == null || inputArr.length == 0) {

return;

}

this.array = inputArr;

length = inputArr.length;

quickSort(0, length - 1);

}

private void quickSort(int lowerIndex, int higherIndex) {

int i = lowerIndex;

int j = higherIndex;

int pivot = array[lowerIndex + (higherIndex - lowerIndex) / 2];

while (i <= j) {

while (array[i] < pivot) {

i++;

}

while (array[j] > pivot) {

j--;

}

if (i <= j) {

exchangeNumbers(i, j);

i++;

j--;

}

}

if (lowerIndex < j) {

quickSort(lowerIndex, j);

}

if (i < higherIndex) {

quickSort(i, higherIndex);

}

}

private void exchangeNumbers(int i, int j) {

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

public class JavaBubbleSort {

public static void main(String[] args) {

int[] inputNumber = {4, 2, 9, 6, 23, 12, 34, 0, 1};

bubble\_srt(inputNumber);

}

private static void bubble\_srt(int[] array) {

int n = array.length;

int k;

for (int m = n; m >= 0; m--) {

for (int i = 0; i < n - 1; i++) {

k = i + 1;

if (array[i] > array[k]) {

swapNumbers(i, k, array);

}

}

printNumbers(array);

}

}

private static void swapNumbers(int i, int j, int[] array) {

int temp;

temp = array[i];

array[i] = array[j];

array[j] = temp;

}

private static void printNumbers(int[] input) {

for (int inp : input) {

System.out.print(inp + " ");

}

System.out.println("\n");

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

import java.util.Stack;

public class JavaDataStructuresStack {

public static void main(String[] args) {

String str = "Java for all";

reverseWords(str);

}

private static void reverseWords(String str) {

Stack<Character> characterStack = new Stack<>();

for (int i = 0; i < str.length(); ++i) {

if (str.charAt(i) != ' ') {

characterStack.push(str.charAt(i));

} else {

while (!characterStack.empty()) {

System.out.print(characterStack.pop());

System.out.print(" ");

}

}

while (!characterStack.empty()) {

System.out.print(characterStack.pop());

}

}

}

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javaalgorithms;

class Node {

int data;

Node left, right;

Node(int d) {

data = d;

left = right = null;

}

}

package javaalgorithms;

public class JavaDataStructuresBinaryTree {

public static void main(String[] args) {

JavaDataStructuresBinaryTree binaryTree = new JavaDataStructuresBinaryTree();

Node root = null;

root = binaryTree.insert(root, 4);

binaryTree.insert(root, 23);

binaryTree.insert(root, -12);

binaryTree.insert(root, 3);

binaryTree.insert(root, 65);

binaryTree.insert(root, 54);

System.out.println("Minimum value of binary binaryTree is "

+ binaryTree.minvalue(root));

}

private Node insert(Node node, int data) {

if (node == null) {

return (new Node(data));

} else {

if (data <= node.data) {

node.left = insert(node.left, data);

} else {

node.right = insert(node.right, data);

}

return node;

}

}

private int minvalue(Node node) {

Node current = node;

while (current.left != null) {

current = current.left;

}

return (current.data);

}

}

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

*1 užduotis*. UŽPILDYKITE PATEIKTĄ LENTELĘ.

Dizaino šablonai:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Dizaino šablono rūšys** | | |
|  | **Creational** | **Structural** | **Behavioral** |
| **Šablonas** | Abstract Factory | Adapter | Chain of Responsibility |
| Builder | Bridge | Command |
| Factory Method | Composite | Interpreter |
| Prototype | Decorator | Iterator |
| Singleton | Facade | Mediator |
|  | Flyweight | Memento |
|  | Proxy | Observer |
|  |  | State |
|  |  | Strategy |
|  |  | Template Method |
|  |  | Visitor |

Papildomai galite paskaityti čia:

Wikipedia (2019). *Projektavimo pavyzdys.* [žiūrėta 2019-05-27].Prieiga per internetą[https://lt.wikipedia.org/wiki/Projektavimo\_pavyzdys*.*](https://lt.wikipedia.org/wiki/Projektavimo_pavyzdys.)

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.

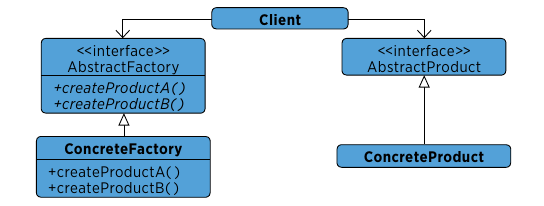
**Decorator šablonas (Dekoratorius):** naudojamas dinamiškai keisti ir pridėti funkcionalumo objektui. Objekto atsakomybė ir elgesys turėtų būti keičiami dinamiškai. Konkretūs įgyvendinimai turėtų būti atsieti nuo atsakomybės ir elgesio. Pavyzdys: daugelis įmonių savo pašto sistemas sukūrė naudodamosi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

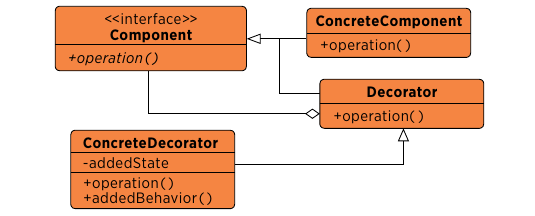
Papildomai galite paskaityti čia:

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

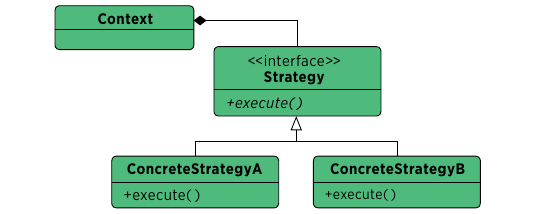
*3 užduotis.* NAUDOJANT UML MODELIAVIMO KALBĄ SUKURKITE KLASIŲ DIAGRAMĄ ABSTRAKČIOS FABRIKOS ŠABLONUI (ABSTRACT FACTORY PATTERN).



*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 fabriko (Factory) 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.>

**2.4. 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 DARBUOTOJŲ VALDYMO PROGRAMĄ.

Pavyzdys. Modelis realizuotas Java programavimo kalba:

package javamvc;

class EmployeeModel {

private String id;

private String name;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

package javamvc;

public class EmployeeController {

private EmployeeModel model;

private EmployeeView view;

public EmployeeController(EmployeeModel model, EmployeeView view) {

this.model = model;

this.view = view;

}

public void setEmployeeName(String name) {

model.setName(name);

}

public String getEmployeeName() {

return model.getName();

}

public void setEmployeeId(String rollNo) {

model.setId(rollNo);

}

public String getEmployeeId() {

return model.getId();

}

public void updateView() {

view.printEmployeeDetails(model.getName(), model.getId());

}

}

package javamvc;

class EmployeeView {

public void printEmployeeDetails(String name, String no) {

System.out.println("EmployeeModel: ");

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

System.out.println("ID: " + no);

}

}

package javamvc;

public class Main {

public static void main(String[] args) {

EmployeeModel employee = new EmployeeModel();

EmployeeView view = new EmployeeView();

EmployeeController controller = new EmployeeController(employee, view);

employee.setId("1");

employee.setName("Tom");

controller.updateView();

controller.setEmployeeName("Dave");

controller.updateView();

}

}

Papildomai galite paskaityti:

Wikipedia (2019). *Model View Controller*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Model-view-controller.>

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

Užduoties sprendimui galite naudoti šį šaltinį:

1. How Laravel implements MVC and how to use it effectively (MAY 24, 2018). *Pusher Tutorials and Articles*. [žiūrėta 2019-05-27]. Prieiga per internetą [https://blog.pusher.com/laravel-mvc-use/.](https://www.javaguides.net/2018/10/spring-mvc-sign-up-form-handling.html.)

2. How to make Login System in Laravel. Published on Mar 5, 2018. [žiūrėta 2019-05- 27]. Prieiga per internetą <https://www.youtube.com/watch?v=OUFmwAnFclo>

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

Užduoties sprendimui naudokite šiuos resursus:

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

2. Wikipedia (2019). *Don’t repeat yourself*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Don’t_repeat_yourself.>

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

Užduoties sprendimui naudokite šiuos resursus:

1. Wikipedia (2019). *Separation of Concerns*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Separation_of_concerns.>

2. Wikipedia (2019). *Code reuse*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Code_reuse.>

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javafunctional;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.function.Function;

import java.util.function.Predicate;

import java.util.stream.Collectors;

import java.util.stream.Stream;

public class JavaQuickSortFunctional {

private static Function<Integer, Predicate<Integer>> smallerThan = x -> y -> y < x;

public static void main(String[] args) {

List<Integer> integerList = Arrays.asList(5, 6, 7, 23, 4, 5645, 6, 1223,

44453, 60182, 2836, 23993, 1);

System.out.println(quickSort(integerList));

}

private static List<Integer> quickSort(List<Integer> integerList) {

if (integerList.isEmpty()) return new ArrayList<>();

return Stream.concat(Stream.concat(quickSort(integerList.stream().skip(1)

.filter(smallerThan.apply(integerList.get(0)))

.collect(Collectors.toList())).stream(),

Stream.of(integerList.get(0))), quickSort(integerList.stream().skip(1)

.filter(smallerThan.apply(integerList.get(0)).negate())

.collect(Collectors.toList())).stream()).collect(Collectors.toList());

}

}

Papildomai apie funkcinį programavimą galite paskaityti čia:

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

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

Pavyzdys. Algoritmas Java programavimo kalba:

package javafunctional;

import java.util.Arrays;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

import java.util.function.Function;

import java.util.stream.Collectors;

public class JavaGroupCountSortFunctional {

public static void main(String[] args) {

List<String> items = Arrays.asList("Tom", "Dave", "John", "Mary",

"Steve", "Tom", "Mary");

Map<String, Long> result = items.stream()

.collect(Collectors.groupingBy(Function

.identity(), Collectors.counting()));

Map<String, Long> finalMap = new LinkedHashMap<>();

result.entrySet().stream().sorted(Map.Entry.<String, Long>comparingByKey())

.forEachOrdered(e -> finalMap.put(e.getKey(), e.getValue()));

System.out.println(finalMap);

}

}

**3. Kompetencija. Kurti tipinę programinę įrangą**.

## 3.1. Mokymosi rezultatas. Naudoti Php programavimo kalbos įrankius ir sintaksę.

*1 užduotis*. ĮDIEGTI PHP INTERPRETATORIŲ.

PHP diegimo paketą siunčiamės iš:

Binaries sources and releases. [žiūrėta 2019-06-28]. Prieiga per internetą <https://windows.php.net/download/>

Diegimo ir konfigūravimo instrukcijos:

Install PHP in Windows 10. [žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jeffgeerling.com/blog/2018/installing-php-7-and-composer-on-windows-10>

Patikriname komandų eilutėje PHP versiją: php –v

*2 užduotis.* ĮDIEGTI APACHE WEB SERVERĮ.

PHP diegimo paketą siunčiamės iš:

Apache 2.4 binaries VS16. [žiūrėta 2019-06-28]. Prieiga per internetą <https://www.apachelounge.com/download/>

Diegimo ir konfigūravimo instrukcijos:

PHP 7 - Installation on Windows with Apache. [žiūrėta 2019-06-28]. Prieiga per internetą <https://www.tutorialspoint.com/php7/php7_installation_windows_apache.htm>

Sukurti failą vardu testas.php su žemiau pateiktu kodu ir jį įvykdyti naršyklėje.

<?php

echo "PHP veikia naršyklėje;"

?>

*3 užduotis*. ĮDIEGTI MYSQL DUOMENŲ BAZĖS SERVERĮ .

MySQL diegimo paketą siunčiamės iš:

MySQL server. [žiūrėta 2019-06-28]. Prieiga per internetą

<https://dev.mysql.com/downloads/mysql/5.7.html#downloads>

Diegimo ir konfigūravimo instrukcijos:

Installing MySQL on Microsoft Windows. [žiūrėta 2019-06-28]. Prieiga per internetą <https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html>

*4 užduotis*. PARUOŠTI PROGRAMAVIMO APLINKĄ IR KURTI PROJEKTĄ.

Sublime diegimo paketą siunčiamės iš:

Download. [žiūrėta 2019-06-28]. Prieiga per internetą

<https://download.sublimetext.com/Sublime%20Text%20Build%203207%20Setup.exe>

Diegimo ir konfigūravimo instrukcijos:

Paleidžiame atsiųstą diegimo failą ir sekame diegimo vedlio nurodymus.

Po diegimo paleidžiame Sublime redaktorių.

Projekto sukūrimas:

* Sukuriame projekto aplanką pavadinimu: projektas1
* Pridedame projekto aplanką redaktoriaus meniu pasirinkę: Project -> Add folder to project
* Išsaugome projektą redaktoriaus meniu pasirinkę: Project -> Save Project As...
* Projekto aplanke sukuriame failą vardu: index.php
* Atidarome failą index.php ir parašome kodą, kuris išveda tekstą:

<?php

echo "PHP projektas pradėtas";

?>

*5 užduotis*. PANAUDOKITE PHP KOMANDINĖJE EILUTĖJE.

$ php -a

$ php > echo "PHP interpretatorius veikia";

Daugiau informacijos:

<https://www.php.net/manual/en/features.commandline.interactive.php>

*6 užduotis*. IŠBANDYKITE PHP BUILT-IN SERVERĮ.

$ mkdir projektas2

$ cd projektas2

$ touch index.php

$ php -S localhost:8000

Atidarykite index.php Sublime redaktoriuje ir įrašykite kodą:

<?php

echo "PHP built-in WEB serveris veikia";

?>

*7 užduotis*. IŠVESKITE MOKINIO DUOMENIS NARŠYKLĖJE.

<?php

$vardas = "Jonas";

$pavarde = "Jonaitis";

$amzius = 39;

$pomegiai = "programuoti, žvejoti, keliauti";

echo "Mokinio duomenys: \n";

echo "Vardas: $vardas\n";

echo "Pavardė: $pavarde\n";

echo "Amžius: $amzius\n";

echo "Pomėgiai: $pomegiai\n";

*8 užduotis*. SUKURKITE MASYVĄ REMIANTIS *7 užduotyje* PATEIKTU PAVYZDŽIU, SUKURKITE MASYVĄ, KURIS TALPINA 3 MOKINIŲ DUOMENIS SU SKIRTINGOMIS REIKŠMĖMIS.

<?php

$duomenys = [

0 => [

'Petras',

'Petraitis',

22,

'Programuoti Medžioti Keliauti',

false

],

1=> [

'Ieva',

'Ievaitė',

19,

'Skaityti Žvejoti Keliauti',

true

],

2=> [

'Inga',

'Ingauskaitė',

40,

'Programuoti Raizgyti Keliauti',

false

],

3 => [

'Rytis',

'Rytauskas',

45,

'Megzti Žvejoti Keliauti',

true

]

];

*9 užduotis.* IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT FOR CIKLĄ.

<?php

echo "Mokinio duomenys:\n";

for($i=0; $i<count($duomenys); $i++){

$ypatybiuSkaicius = count($duomenys[$i]);

$ypatybes = $duomenys[$i];

echo "Mokinio nr. : $i\n";

for($p=0; $p<$ypatybiuSkaicius; $p++){

echo "$ypatybes[$p]\n";

}

}

*10 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT WHILE CIKLĄ.

<?php

$i=0;

$p=0;

$mokiniuskaicius = count($duomenys);

while($i<$mokiniuskaicius) {

echo "Mokinio nr. : $i\n";

$ypatybiuSkaicius = count($duomenys[$i]);

$ypatybes = $duomenys[$i];

$i++;

while($p<$ypatybiuSkaicius) {

echo "$ypatybes[$p]\n";

echo $p;

$p++;

}

}

*11 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT DO WHILE CIKLĄ.

<?php

$i = 0;

$mokiniuskaicius = count($duomenys[$i]);

$mokiniai = $duomenys[$i];

do {

echo "$mokiniai[$i]\n";

$i++;

} while ($i<$mokiniuskaicius);

*12 užduotis*. SUKURKITE ASOCIATYVŲ MASYVĄ.

<?php

$duomenys = [

'mokinys1' => [

'vardas' => 'Jonas',

'pavarde' => 'Jonaitis',

'amzius' => 32,

'pomegiai' => 'Programuoti', 'Žvejoti', 'Keliauti',

],

'mokinys2' => [

'vardas' => 'Petras',

'pavarde' => 'Petraitis',

'amzius' => 22,

'pomegiai' => 'Programuoti', 'Medžioti', 'Keliauti',

],

'mokinys3' => [

'vardas' => 'Ieva',

'pavarde' => 'Ievaitė',

'amzius' => 19,

'pomegiai' => 'Skaityti', 'Žvejoti', 'Keliauti',

],

'mokinys4' => [

'vardas' => 'Inga',

'pavarde' => 'Ingauskaitė',

'amzius' => 40,

'pomegiai' => 'Programuoti', 'Raizgyti', 'Keliauti',

],

'mokinys5' => [

'vardas' => 'Rytis',

'pavarde' => 'Rytauskas',

'amzius' => 45,

'pomegiai' => 'Megzti', 'Žvejoti', 'Keliauti',

]

];

*13 užduotis*. IŠVESKITE MOKINIŲ DUOMENIS NARŠYKLĖJE NAUDOJANT FOREACH CIKLĄ.

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Mokiniai</title>

</head>

<body>

<ul>

<?php foreach ($duomenys as $kodas =>$mokinys):?>

<li>

<?=$kodas ?>

<ul>

<?php foreach($duomenys[$kodas] as $ypatybe =>$ypatybesReiksme):?>

<li><?=$ypatybe.':'.$ypatybesReiksme;?></li>

<?php endforeach;?>

</ul>

</li>

<?php endforeach; ?>

</ul>

</body>

</html>

*14 užduotis*. IŠVESKITE MOKINIO DUOMENIS NARŠYKLĖJE NAUDOJANT FUNKCIJĄ REMIANTIS *7 užduotyje* PATEIKTU PAVYZDŽIU, SUKURKITE ASOCIATYVŲ MASYVĄ, KURIS TALPINA 3 MOKINIŲ DUOMENIS SU SKIRTINGOMIS REIKŠMĖMIS.

<!doctype html>

<?php

$duomenys = [

'mokinys1' => [

'vardas' => 'Jonas',

'pavarde' => 'Jonaitis',

'amzius' => 32,

'pomegiai' => 'Programuoti', 'Žvejoti', 'Keliauti',

],

'mokinys2' => [

'vardas' => 'Petras',

'pavarde' => 'Petraitis',

'amzius' => 22,

'pomegiai' => 'Programuoti', 'Medžioti', 'Keliauti',

],

'mokinys4' => [

'vardas' => 'Ieva',

'pavarde' => 'Ievaitė',

'amzius' => 19,

'pomegiai' => 'Skaityti', 'Žvejoti', 'Keliauti',

],

'mokinys3' => [

'vardas' => 'Inga',

'pavarde' => 'Ingauskaitė',

'amzius' => 40,

'pomegiai' => 'Programuoti', 'Raizgyti', 'Keliauti',

],

'mokinys5' => [

'vardas' => 'Rytis',

'pavarde' => 'Rytauskas',

'amzius' => 45,

'pomegiai' => 'Megzti', 'Žvejoti', 'Keliauti',

]

];

function mokiniai($data)

{

ksort($data);

echo "<ul>";

foreach ($data as $kodas => $mokinys) {

echo "<li>$kodas";

echo "<ul>";

foreach ($data[$kodas] as $ypatybe => $ypatybesReiksme) {

echo "<li>$ypatybe : $ypatybesReiksme</li>";

}

echo "</ul>";

echo "</li>";

}

echo "</ul>";

}

?>

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Mokiniai</title>

</head>

<body>

<?php mokiniai($duomenys); ?>

</body>

</html>

*15 užduotis***.** APRAŠYKITE DUOMENIS MASYVE IR IŠVESKITE HTML LENTELĖJE.

<?php

$mokiniai = [

"5b" => [

round(rand(3,5),2) => [

'ivedimo\_data' => date('y-m-d'),

'vardas' => 'Petras',

'pavarde' => 'Petraitis',

'ivertinimai' => [

'matematika' => 5,

'it' => 9,

'anglu' => 10

]

],

round(rand(3,5),2) => [

'ivedimo\_data' => date('y-m-d'),

'vardas' => 'Jonas',

'pavarde' => 'Jonaitis',

'ivertinimai' => [

'matematika' => 5,

'it' => 8,

'anglu' => 6

]

],

round(rand(3,5),2) => [

'ivedimo\_data' => date('y-m-d'),

'vardas' => 'Ieva',

'pavarde' => 'Ievaka',

'ivertinimai' => [

'matematika' => 5,

'it' => 8,

'anglu' => 6

]

]

],

"5a" => [

round(rand(3,5),2) => [

'ivedimo\_data' => date('y-m-d'),

'vardas' => 'Jurgis',

'pavarde' => 'Jurgaitis',

'ivertinimai' => [

'matematika' => 10,

'it' => 9,

'anglu' => 10

]

],

round(rand(3,5),2) => [

'ivedimo\_data' => date('y-m-d'),

'vardas' => 'Zenonas',

'pavarde' => 'Zenauskas',

'ivertinimai' => [

'matematika' => 9,

'it' => 8,

'anglu' => 6

]

]

]

];

?>

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

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

<title>Mokiniai</title>

</head>

<body>

<table>

<tr>

<th>Klasė</th>

<th>Kodas</th>

<th>Duomenų įvedimo data</th>

<th>Vardas</th>

<th>Pavardė</th>

<th>Kontrolinių darbų vidurkis</th>

</tr>

<?php foreach($mokiniai as $group => $data):?>

<tr>

<td><?=$group;?></td>

</tr>

<?php foreach ($data as $code => $student):?>

<tr>

<td></td>

<td><?=$code; ?></td>

<?php foreach ($student as $key =>$value):?>

<?php if($key == 'ivertinimai'):?>

<td><?php

$average = array\_sum($value)/count($value);

echo round($average,2);?>

</td>

<?php else:?>

<td><?=strtoupper($value);?></td>

<?php endif;?>

<?php endforeach;?>

</tr>

<?php endforeach;?>

<?php endforeach; ?>

</table>

</body>

</html>

*16 užduotis*. IŠVESKITE GET METODU GAUTUS DUOMENIS.

<?php

echo $\_GET['vardas'];

echo $\_GET['pavarde'];

echo intval($\_GET['amzius']);

*17 užduotis*. IŠVESKITE GET METODU GAUTUS DUOMENIS BE HTML.

<?php

echo htmlspecialchars($\_GET['komentaras']);

*18 užduotis*. IŠVESKITE POST METODU IŠ FORMOS GAUTUS DUOMENIS.

<?php

<?php

if(isset($\_POST['submit'])){

$validation\_errors = [];

if (!preg\_match('/^[\w\d]{4,100}$/',

$\_POST['vardas']))

{

$validation\_errors[] = "Vardas negali būti ilgesnis už 100 simbolių ir trumpesnis už 4";

}

else {

echo '<li>'.$\_POST['vardas'].'</li>';

}

if (!preg\_match('/^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/',

$\_POST['email'])){

$validation\_errors[] = "El. paštas turi atitikti el. pašto formatą";

}

else {

echo '<li>'.$\_POST['email'].'</li>';

}

if (!preg\_match('/^[+0-9. ()-]\*$/ui',

$\_POST['phone'])){

$validation\_errors[] = "Telefonas turi atitikti lietiviško telefono formatą su šalies kodu";

}

else {

echo '<li>'.$\_POST['phone'].'</li>';

}

if (!preg\_match('/^[0-9]+(\\.[0-9]+)?$/',

$\_POST['price'])){

$validation\_errors[] = "Kaina turi atitikti slankaus kablelio skaičiaus formatą";

}

else {

echo '<li>'.$\_POST['price'].'</li>';

}

if (!preg\_match('/^[\w\d]{4,500}$/',

$\_POST['message'])){

$validation\_errors[] = "Žinutės tekstas, negali viršyti 500 simbolių ir trumpesnis už 4";

}

else {

echo '<li>'.$\_POST['message'].'</li>';

}

}

?>

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Forma</title>

</head>

<body>

<?php if($validation\_errors):?>

<h2>Formoje yra klaidų</h2>

<ul>

<?php foreach($validation\_errors as $error):?>

<li><?=$error;?></li>

<?php endforeach; ?>

</ul>

<?php endif;?>

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

<input type="text" name="vardas" placeholder="Vardas" required>

<input type="text" name="email" placeholder="El. paštas" required>

<input type="tel" name="phone" placeholder="Telefono numeris">

<input type="text" name="price" placeholder="Kaina" required>

<input type="text" name="message" placeholder="Žinutė" required>

<button type="submit" name="submit">Siųsti</button>

</form>

</body>

</html>

### 3.2. Mokymosi rezultatas. Kurti nesudėtingą programinį kodą PHP programavimo kalba.

*1 užduotis*. APRAŠYKITE ASOCIATYVŲ MASYVĄ, KURIAME SAUGOMA INFORMACIJA APIE AVIAKOMPANIJOS VYKDOMUS SKRYDŽIUS (NE MAŽIAU, KAIP 5 SKRYDŽIAI). inc/flights.php

<?php

$flights = [

'OSL-567' => [

'direction' => "Oslas, Norvegija",

'price' =>1000.56

],

'BUD-787' => [

'direction' => "Budapeštas, Vengrija",

'price' =>690.67

],

'TEN-562' => [

'direction' => "Tenerifė, Ispanija",

'price' =>890.45

],

'ROM-593' => [

'direction' => "Roma, Italija",

'price' =>1200

],

'PAR-129' => [

'direction' => "Paryžius, Prancūzija",

'price' =>457.67

],

];

*2 užduotis. 1 užduotyje* SUKURTĄ MASYVĄ SURIKIUOKITE PAGAL KAINĄ DIDĖJANČIA TVARKA.

inc/flights.php

<?php

$keys = array\_column($flights, 'price');

array\_multisort($keys, SORT\_ASC, $flights);

*3 užduotis*. SUKURKITE FORMĄ SU ŽEMIAU PATEIKTAIS LAUKAIS.

index.php

<?php

include 'inc/flights.php';

?>

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Skrydžio forma</title>

</head>

<body>

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

<select name="flight" required>

<option selected disabled>--- Skrydžio nr. ---</option>

<?php foreach ($flights as $flightNumber => $flightDetails):?>

<option value="<?=$flightNumber; ?>"><?=$flightNumber; ?></option>

<?php endforeach; ?>

</select>

<input type="text" name="pid" placeholder="Asmens kodas" required>

<input type="text" name="name" placeholder="Vardas" required>

<input type="text" name="lastname" placeholder="Pavardė" required>

<select name="extra" required>

<option selected disabled>--- Bagažas ---</option>

<option value="10kg">10 kg.</option>

<option value="15kg">15 kg.</option>

<option value="30kg">30 kg.</option>

<option value="50kg">50 kg.</option>

</select>

<textarea name="note" id="" cols="30" rows="10" placeholder="Pastabos"></textarea>

<button type="submit" name="submit">Formuoti užsakymą</button>

</form>

</body>

</html>

*4 užduotis*. SUKURKITE FORMOS DUOMENIS, KURIŲ ĮVEDIMAS TURI BŪTI VALIDUOJAMAS PRIEŠ DUOMENIS IŠSIUNČIANT IŠ FORMOS (NAUDOJANT REGEX).

index.php

<?php

include 'inc/flights.php';

$validation\_errors = [];

$data = [];

if(isset($\_POST['submit'])){

if (!preg\_match('/^.{7}$/',

$\_POST['flight']))

{

$validation\_errors[] = "Nepasirinktas skrydis";

}

else {

$data[]=$\_POST['flight'];

}

$valid=false;

$ak=$\_POST['pid'];

if(strlen($ak)==11){

if($ak[0]>2 && $ak[0]<7){

if(checkdate(substr($ak,3,2),substr($ak,5,2),substr($ak,1,2))){

$str=$ak[0]\*1+$ak[1]\*2+$ak[2]\*3+$ak[3]\*4+$ak[4]\*5+$ak[5]\*6+$ak[6]\*7+$ak[7]\*8+$ak[8]\*9+$ak[9]\*1;

$str=$str%11;

if($str==10){

$str=$ak[0]\*3+$ak[1]\*4+$ak[2]\*5+$ak[3]\*6+$ak[4]\*7+$ak[5]\*8+$ak[6]\*9+$ak[7]\*1+$ak[8]\*2+$ak[9]\*3;

$str=$str%11;

if($str==10 && substr($ak,10,1)=="0")

$valid=true;

elseif($str==substr($ak,10,1))

$valid=true;

}

elseif($str==substr($ak,10,1))

$valid=true;

}

}

$data[]=$\_POST['pid'];

}

else {

$validation\_errors[] = "Neattitinka asmens kodo formato";

}

if (!preg\_match('/^[\w\d]{4,100}$/',

$\_POST['name'])){

$validation\_errors[] = "Žinutės tekstas, negali viršyti 100 simbolių ir trumpesnis už 4";

}

else {

$data[]=$\_POST['name'];

}

if (!preg\_match('/^[\w\d]{4,100}$/',

$\_POST['lastname'])){

$validation\_errors[] = "Žinutės tekstas, negali viršyti 100 simbolių ir trumpesnis už 4";

}

else {

$data[]=$\_POST['lastname'];

}

if (!preg\_match('/^[\w\d]{4}$/',

$\_POST['extra'])){

$validation\_errors[] = "Nepasirinktas bagažas";

}

else {

$data[]=$\_POST['extra'];

}

if (!preg\_match('/^.{50,1000}$/',

$\_POST['note'])){

$validation\_errors[] = "Žinutės tekstas, negali viršyti 1000 simbolių ir trumpesnis už 50";

}

else {

$data[]=$\_POST['note'];

}

}

*5 užduotis*. UŽPILDYKITE *3 užduotyje* NURODYTĄ FORMĄ IR PASPAUSKITE FORMOS MYGTUKĄ „FORMUOTI UŽSAKYMĄ“ DUOMENIS IŠSAUGANT MASYVE.

index.php

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Skrydžio forma</title>

</head>

<body>

<?php if(count($data)<=5):?>

<?php if($validation\_errors):?>

<h2>Formoje yra klaidų</h2>

<ul>

<?php foreach($validation\_errors as $error):?>

<li><?=$error;?></li>

<?php endforeach; ?>

</ul>

<?php endif;?>

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

<select name="flight">

<option selected disabled>--- Skrydžio nr. ---</option>

<?php foreach ($flights as $flightNumber => $flightDetails):?>

<option value="<?=$flightNumber; ?>"><?=$flightNumber; ?></option>

<?php endforeach; ?>

</select>

<input type="text" name="pid" placeholder="Asmens kodas">

<input type="text" name="name" placeholder="Vardas">

<input type="text" name="lastname" placeholder="Pavardė">

<select name="extra" required>

<option selected disabled>--- Bagažas ---</option>

<option value="10kg">10 kg.</option>

<option value="15kg">15 kg.</option>

<option value="30kg">30 kg.</option>

<option value="50kg">50 kg.</option>

</select>

<textarea name="note" id="" cols="30" rows="10" placeholder="Pastabos"></textarea>

<button type="submit" name="submit">Formuoti užsakymą</button>

</form>

<?php else: ?>

<h2>Duomenys</h2>

<table>

<tr>

<th>Skydžio nr.</th>

<th>Asmens kodas</th>

<th>Vardas</th>

<th>Pavardė</th>

<th>Bagažas</th>

<th>Pastabos</th>

</tr>

<tr>

<?php foreach($data as $key => $value): ?>

<td><?php

if($key ==='extra'){

$value = (int)$value;

echo $value;

} else{

echo $value;

}

?>

</td>

<?php endforeach; ?>

</tr>

</table>

<?php endif;?>

</body>

</html>

*6 užduotis*. PAPILDYKITE *1 užduotyje* KODĄ TAIP, KAD PASPAUDUS „FORMUOTI UŽSAKYMĄ“ VISA UŽSAKYMO INFORMACIJA IŠ FORMOS BŪTŲ IŠSAUGOTA TEKSTINIAME FAILE DUOMENYS.TXT.

<?php

$my\_file = 'duomenys.txt';

$handle = fopen($my\_file, 'w') or die('Negaliu atidaryti failo: '.$my\_file);

$orders = implode(";",$data);

fwrite($handle, $orders);

*7 užduotis*. ĮDIEKITE PAPILDOMŲ BIBLIOTEKŲ DIEGIMUI ĮRANKĮ COMPOSER, KURĮ GALITE ATSISIŲSTI IŠ: <https://getcomposer.org/download/>

Diegimo ir konfigūravimo instrukcijos:

Composer. [žiūrėta 2019-06-28]. Prieiga per internetą

<https://getcomposer.org/download/>

*8 užduotis*. PRIDĖKITE NUORODĄ SPAUSDINTI VIRŠ *5 užduotyje* SUGENERUOTOS HTML LENTELĖS SU SKRYDŽIO INFORMACIJA IŠ FORMOS.

TCPDF diegimas

$ composer require tecnickcom/tcpdf

print.php

<?php

// Include the main TCPDF library (search for installation path).

require\_once('vendor/tecnickcom/tcpdf/tcpdf.php');

$my\_file = 'duomenys.txt';

$handle = fopen($my\_file, 'r');

$data = fread($handle,filesize($my\_file));

$orders = explode(";",$data);

$orderData = "<ul>";

foreach ($orders as $item){

$orderData .="<li>$item";

}

$orderData .= "</ul>";

// create new PDF document

$pdf = new TCPDF(PDF\_PAGE\_ORIENTATION, PDF\_UNIT, PDF\_PAGE\_FORMAT, true, 'UTF-8', false);

// set document information

$pdf->SetCreator(PDF\_CREATOR);

$pdf->SetAuthor('PHP');

$pdf->SetTitle('Skrydžio užsakymas');

$pdf->SetSubject('Detali informacija');

// set default header data

$pdf->SetHeaderData(PDF\_HEADER\_LOGO, PDF\_HEADER\_LOGO\_WIDTH, PDF\_HEADER\_TITLE.' 001', PDF\_HEADER\_STRING, array(0,64,255), array(0,64,128));

$pdf->setFooterData(array(0,64,0), array(0,64,128));

// set header and footer fonts

$pdf->setHeaderFont(Array(PDF\_FONT\_NAME\_MAIN, '', PDF\_FONT\_SIZE\_MAIN));

$pdf->setFooterFont(Array(PDF\_FONT\_NAME\_DATA, '', PDF\_FONT\_SIZE\_DATA));

// set default monospaced font

$pdf->SetDefaultMonospacedFont(PDF\_FONT\_MONOSPACED);

// set margins

$pdf->SetMargins(PDF\_MARGIN\_LEFT, PDF\_MARGIN\_TOP, PDF\_MARGIN\_RIGHT);

$pdf->SetHeaderMargin(PDF\_MARGIN\_HEADER);

$pdf->SetFooterMargin(PDF\_MARGIN\_FOOTER);

// set auto page breaks

$pdf->SetAutoPageBreak(TRUE, PDF\_MARGIN\_BOTTOM);

// set image scale factor

$pdf->setImageScale(PDF\_IMAGE\_SCALE\_RATIO);

// set some language-dependent strings (optional)

if (@file\_exists(dirname(\_\_FILE\_\_).'/lang/eng.php')) {

require\_once(dirname(\_\_FILE\_\_).'/lang/eng.php');

$pdf->setLanguageArray($l);

}

// ---------------------------------------------------------

// set default font subsetting mode

$pdf->setFontSubsetting(true);

// Set font

// dejavusans is a UTF-8 Unicode font, if you only need to

// print standard ASCII chars, you can use core fonts like

// helvetica or times to reduce file size.

$pdf->SetFont('dejavusans', '', 14, '', true);

// Add a page

// This method has several options, check the source code documentation for more information.

$pdf->AddPage();

// set text shadow effect

$pdf->setTextShadow(array('enabled'=>true, 'depth\_w'=>0.2, 'depth\_h'=>0.2, 'color'=>array(196,196,196), 'opacity'=>1, 'blend\_mode'=>'Normal'));

// Set some content to print

$html = <<<EOD

{$orderData}

EOD;

// Print text using writeHTMLCell()

$pdf->writeHTMLCell(0, 0, '', '', $html, 0, 1, 0, true, '', true);

// ---------------------------------------------------------

// Close and output PDF document

// This method has several options, check the source code documentation for more information.

$pdf->Output('example\_001.pdf', 'I');

//============================================================+

// END OF FILE

//============================================================+

*9 užduotis.* KOMENTUOKITE IR DOKUMENTUOKITE PDF FAILO GENERAVIMO KODĄ IŠ *8 užduoties* NAUDOJANT PHPDOC STANDARTĄ.

print.php

/\*\*

\* Creates an PDF ticket order document using TCPDF

\* @package com.tecnick.tcpdf

\* @abstract TCPDF – Order Example: Default Header and Footer

\* @author Nicola Asuni

\* @since 2019-06-28

\*/

# *10 užduotis.* PHP PROGRAMAVIMO STANDARTŲ TESTAS.

# Testo atsakymai

Pažymėkite, ar pateikti kodo fragmentai atitinka PHP programavimo standartus.

1. b)
2. a)
3. a)
4. b)
5. b)

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

*1 užduotis*. SUKURKITE KLASĘ PAVADINIMU VARTOTOJAS.

<?php

class Vartotojas

{

public $id;

public $vardas;

public $elPastas;

public $slaptazodis;

public $prieigosTeises;

public function prisijungti()

{

echo "$this->vardas prisijungė prie sistemos";

}

public function atisijungti()

{

echo "$this->vardas atsijungė nuo sistemos";

}

}

*2 užduotis*. *1 užduotyje* SUKURTOS KLASĖS PAGRINDU SUKURKITE 3 SKIRTINGUS VARTOTOJUS SU KLASĖJE APRAŠYTŲ SAVYBIŲ REIKŠMĖMIS, TAIP PAT REALIZUOKITE KLASĖJE SUKURTUS METODUS.

<?php

$vartotojas1 = new Vartotojas();

$vartotojas1->id = rand(10,15);

$vartotojas1->vardas = "Jonas";

$vartotojas1->elPastas = "jonas@gmail.com";

$vartotojas1->slaptazodis = password\_hash('jonas456TXW',PASSWORD\_DEFAULT);

$vartotojas1->prieigosTeises = 'vartotojas';

$vartotojas2 = new Vartotojas();

$vartotojas2->id = rand(10,15);

$vartotojas2->vardas = "Petras";

$vartotojas2->elPastas = "petras@gmail.com";

$vartotojas2->slaptazodis = password\_hash('petras456TXW',PASSWORD\_DEFAULT);

$vartotojas2->prieigosTeises = 'administratorius';

$vartotojas3 = new Vartotojas();

$vartotojas3->id = rand(10,15);

$vartotojas3->vardas = "Paulius";

$vartotojas3->elPastas = "paulius@gmail.com";

$vartotojas3->slaptazodis = password\_hash('paulius456TXW',PASSWORD\_DEFAULT);

$vartotojas3->prieigosTeises = 'vartotojas';

$vartotojas1->prisijungti();

$vartotojas2->prisijungti();

$vartotojas3->prisijungti();

$vartotojas1->atisijungti();

$vartotojas2->atisijungti();

$vartotojas3->atisijungti();

*3 užduotis. 1 užduotyje* SUKURTOS KLASĖS SAVYBĖS NETURI BŪTI SKAITOMOS/PERRAŠOMOS TIESIOGIAI KREIPIANTIS Į JAS KLASĖS IŠORĖJE.

Pakeiskite savybes taip, kad minėtas reikalavimas būtų realizuotas.

<?php

class Vartotojas

{

private $id;

private $vardas;

private $elPastas;

private $slaptazodis;

private $prieigosTeises;

}

*4 užduotis*. *1 užduotyje* SUKURTOJE KLASĖJE SUKURKITE „SET“ METODĄ PAVADINIMU REGISTRACIJA, KURIS INICIJUOTOMS VARTOTOJO OBJEKTO SAVYBĖMS LEISTŲ PRISKIRTI REIKŠMES.

<?php

class Vartotojas

{

private $id;

private $vardas;

private $elPastas;

private $slaptazodis;

private $prieigosTeises;

public function registracija($vardas, $elPastas, $slaptazodis)

{

$this->id = rand(10,15);

$this->vardas = $vardas;

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);;

$this->preigosTeises = 'vartotojas';

}

public function prisijungti()

{

echo "$this->vardas prisijungė prie sistemos\n";

}

public function atisijungti()

{

echo "$this->vardas atsijungė nuo sistemos\n";

}

}

$vartotojas1 = new Vartotojas();

$vartotojas1->registracija("Jonas","jonas@gmail.com", "jonas456TXW");

$vartotojas2 = new Vartotojas();

$vartotojas2->regsitracija("Petras","petras@gmail.com", "petras456TXW");

$vartotojas3 = new Vartotojas();

$vartotojas3->registracija("Paulius","paulius@gmail.com", "paulius456TXW");

$vartotojas1->prisijungti();

$vartotojas2->prisijungti();

$vartotojas3->prisijungti();

$vartotojas1->atisijungti();

$vartotojas2->atisijungti();

*5 užduotis*. *1 užduotyje* SUKURTOJE KLASĖJE SUKURKITE „GET“ METODĄ PAVADINIMU PROFILIS, KURIS VISAS VARTOTOJO SAVYBIŲ REIKŠMES GRAŽINTŲ KAIP ASOCIATYVŲ MASYVĄ.

<?php

class Vartotojas

{

private $id;

private $vardas;

private $elPastas;

private $slaptazodis;

private $prieigosTeises;

public function registracija($vardas, $elPastas, $slaptazodis)

{

$this->id = rand(10,15);

$this->vardas = $vardas;

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);;

$this->preigosTeises = 'vartotojas';

}

public function profilis()

{

$data[] = $this->id;

$data[] = $this->vardas;

$data[] = $this->elPastas;

$data[] = $this->prieigosTeises;

return $data;

}

}

$vartotojas1 = new Vartotojas();

$vartotojas1->registracija("Jonas","jonas@gmail.com", "jonas456TXW");

var\_dump($vartotojas1->profilis());

$vartotojas2 = new Vartotojas();

$vartotojas2->registracija("Petras","petras@gmail.com", "petras456TXW");

var\_dump($vartotojas2->profilis());

$vartotojas3 = new Vartotojas();

$vartotojas3->registracija("Paulius","paulius@gmail.com", "paulius456TXW");

var\_dump($vartotojas3->profilis());

*6 užduotis*. *1 užduotyje* SUKURTOJE KLASĖJE SUKURKITE KONSTRUKTORIŲ, KURIS LEISTŲ SUKURTI NAUJĄ OBJEKTĄ NURODANT SAVYBIŲ (ID, VARDAS) REIKŠMES.

<?php

class Vartotojas

{

private $id;

private $vardas;

private $elPastas;

private $slaptazodis;

private $prieigosTeises;

public function \_\_construct($vardas)

{

$this->id = rand(10,15);

$this->vardas = $vardas;

$this->preigosTeises = 'vartotojas';

}

public function profilioAtnaujinimas($elPastas, $slaptazodis)

{

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);;

}

public function profilis()

{

$data[] = $this->id;

$data[] = $this->vardas;

$data[] = $this->elPastas;

$data[] = $this->prieigosTeises;

return $data;

}

}

$vartotojas1 = new Vartotojas("Jonas");

$vartotojas1->profilioAtnaujinimas("jonas@gmail.com", "jonas456TXW");

var\_dump($vartotojas1->profilis());

$vartotojas2 = new Vartotojas("Petras");

$vartotojas2->profilioAtnaujinimas("petras@gmail.com", "petras456TXW");

var\_dump($vartotojas2->profilis());

$vartotojas3 = new Vartotojas("Paulius");

$vartotojas3->profilioAtnaujinimas("paulius@gmail.com", "paulius456TXW");

var\_dump($vartotojas3->profilis());

*7 užduotis.* IŠTAISYKITE ŽEMIAU PATEIKTĄ KODĄ, KAD NEGAUTUME KLAIDOS.

<?php

class Pasisveikinimas

{

public $public = "Public";

private $private = "Private";

public $protected = "Protected";

public function sakykLabas

{

echo $this->public;

echo $this->private;

echo $this->protected;

}

}

$start = new Pasisveikinimas();

echo $start->protected;

*8 užduotis.* SUKURKITE KLASĘ PAVADINIMU „RENDER“.

<?php

class Render

{

public static function spausdinti($data)

{

echo "<table>";

echo "<tr>";

foreach ($data as $item){

echo "<td>$item</td>";

}

echo "</tr>";

echo "</table>";

}

}

*9 užduotis*. *1 užduotyje* SUKURTOS KLASĖS PAGRINDU SUKURKITE NAUJĄ OBJEKTĄ.

<?php

class Vartotojas

{

private $id;

private $vardas;

private $elPastas;

private $slaptazodis;

private $prieigosTeises;

public function \_\_construct($vardas)

{

$this->id = rand(10,15);

$this->vardas = $vardas;

$this->preigosTeises = 'vartotojas';

}

public function profilioAtnaujinimas($elPastas, $slaptazodis)

{

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);;

}

public function profilis()

{

$data[] = $this->id;

$data[] = $this->vardas;

$data[] = $this->elPastas;

$data[] = $this->prieigosTeises;

return $data;

}

}

class Render

{

public static function spausdinti($data)

{

echo "<table>";

echo "<tr>";

foreach ($data as $item){

echo "<td>$item</td>";

}

echo "</tr>";

echo "</table>";

}

}

$vartotojas1 = new Vartotojas("Jonas");

$vartotojas1->profilioAtnaujinimas("jonas@gmail.com", "jonas456TXW");

$vartotojas2 = new Vartotojas("Petras");

$vartotojas2->profilioAtnaujinimas("petras@gmail.com", "petras456TXW");

$vartotojas3 = new Vartotojas("Paulius");

$vartotojas3->profilioAtnaujinimas("paulius@gmail.com", "paulius456TXW");

Render::spausdinti($vartotojas1->profilis());

Render::spausdinti($vartotojas2->profilis());

Render::spausdinti($vartotojas3->profilis());

*10 užduotis*. SUKURDAMI NAUJĄ KLASĘ DARBUOTOJAS PRAPLĖSKITE*1 užduotyje* SUKURTĄ KLASĘ VARTOTOJAS.

<?php

class Darbuotojas extends Vartotojas

{

protected $pareigos;

protected $darboUzmokestis;

protected $idarbinimoData;

public function \_\_construct($vardas)

{

parent::\_\_construct($vardas);

}

public function profilioAtnaujinimas($elPastas, $slaptazodis,$pareigos, $darboUzmokestis, $idarbinimoData)

{

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);

$this->pareigos = $pareigos;

$this->darboUzmokestis = $darboUzmokestis;

$this->idarbinimoData = $idarbinimoData;

}

public function profilis()

{

$data[] = $this->id;

$data[] = $this->vardas;

$data[] = $this->elPastas;

$data[] = $this->prieigosTeises;

$data[] = $this->pareigos;

$data[] = $this->darboUzmokestis;

$data[] = $this->idarbinimoData;

return $data;

}

}

*11 užduotis*. *1 užduotyje* SUKURTĄ KLASĘ VARTOTOJAS PASKELBKITE ABSTRAKČIA.

<?php

abstract class Vartotojas

{

protected $id;

protected $vardas;

protected $elPastas;

protected $slaptazodis;

protected $prieigosTeises;

abstract public function fillPortfolio ($elPastas, $slaptazodis,$pareigos, $darboUzmokestis, $idarbinimoData);

abstract public function showPortfolio();

}

*12 užduotis*. SUKURKITE KLASĘ STUDENTAS SU YPATYBĖMIS (MOKYMOSI PROGRAMA, KURSAS), KURI PAVELDI ABSTRAKČIĄ KLASĘ VARTOTOJAS.

<?php

abstract class Vartotojas

{

protected $id;

protected $vardas;

protected $elPastas;

protected $slaptazodis;

protected $prieigosTeises;

public function \_\_construct($vardas)

{

$this->id = rand(10,15);

$this->vardas = $vardas;

$this->preigosTeises = 'vartotojas';

}

abstract public function fillPortfolio ();

abstract public function showPortfolio();

}

class Studentas extends Vartotojas{

private $mokymosiPrograma;

private $kursas;

public function \_\_construct($vardas)

{

parent::\_\_construct($vardas);

}

public function fillPortfolio()

{

list($elPastas, $slaptazodis,$mokymosiPrograma,$kursas) = func\_get\_args();

$this->elPastas = $elPastas;

$this->slaptazodis = password\_hash($slaptazodis,PASSWORD\_DEFAULT);

$this->mokymosiPrograma = $mokymosiPrograma;

$this->kursas = $kursas;

}

public function showPortfolio()

{

$data[] = $this->id;

$data[] = $this->vardas;

$data[] = $this->elPastas;

$data[] = $this->prieigosTeises;

$data[] = $this->mokymosiPrograma;

$data[] = $this->kursas;

return $data;

}

}

$vartotojas1 = new Studentas("Jonas");

$vartotojas1->fillPortfolio("jonas@gmail.com", "jonas456TXW","Finansai","I");

$vartotojas2 = new Studentas("Petras");

$vartotojas2->fillPortfolio("petras@gmail.com", "petras456TXW","IT","II");

$vartotojas3 = new Darbuotojas("Paulius");

$vartotojas3->fillPortfolio("paulius@gmail.com", "paulius456TXW","programuotojas","1200EUR","2019-08-12");

var\_dump($vartotojas1->showPortfolio ());

*13 užduotis*. PAGAL ŽEMIAU PATEIKTĄ KLASIŲ DIAGRAMĄ SUKURKITE KLASES IR INTERFEISUS SU METODAIS.

**BookInterface.php**

<?php

interface BookInterface

{

public function turnPage();

public function open();

public function getPage(): int;

}

**EBookAdapter.php**

<?php

/\*\*

\* This is the adapter here. Notice it implements BookInterface,

\* therefore you don't have to change the code of the client which is using a Book

\*/

class EBookAdapter implements BookInterface

{

/\*\*

\* @var EBookInterface

\*/

protected $eBook;

/\*\*

\* @param EBookInterface $eBook

\*/

public function \_\_construct(EBookInterface $eBook)

{

$this->eBook = $eBook;

}

/\*\*

\* This class makes the proper translation from one interface to another.

\*/

public function open()

{

$this->eBook->unlock();

}

public function turnPage()

{

$this->eBook->pressNext();

}

/\*\*

\* notice the adapted behavior here: EBookInterface::getPage() will return two integers, but BookInterface

\* supports only a current page getter, so we adapt the behavior here

\*

\* @return int

\*/

public function getPage(): int

{

return $this->eBook->getPage()[0];

}

}

**EBookInterface.php**

<?php

interface EBookInterface

{

public function unlock();

public function pressNext();

/\*\*

\* returns current page and total number of pages, like [10, 100] is page 10 of 100

\*

\* @return int[]

\*/

public function getPage(): array;

}

**Kindle.php**

<?php

/\*\*

\* this is the adapted class. In production code, this could be a class from another package, some vendor code.

\* Notice that it uses another naming scheme and the implementation does something similar but in another way

\*/

class Kindle implements EBookInterface

{

/\*\*

\* @var int

\*/

private $page = 1;

/\*\*

\* @var int

\*/

private $totalPages = 100;

public function pressNext()

{

$this->page++;

}

public function unlock()

{

}

/\*\*

\* returns current page and total number of pages, like [10, 100] is page 10 of 100

\*

\* @return int[]

\*/

public function getPage(): array

{

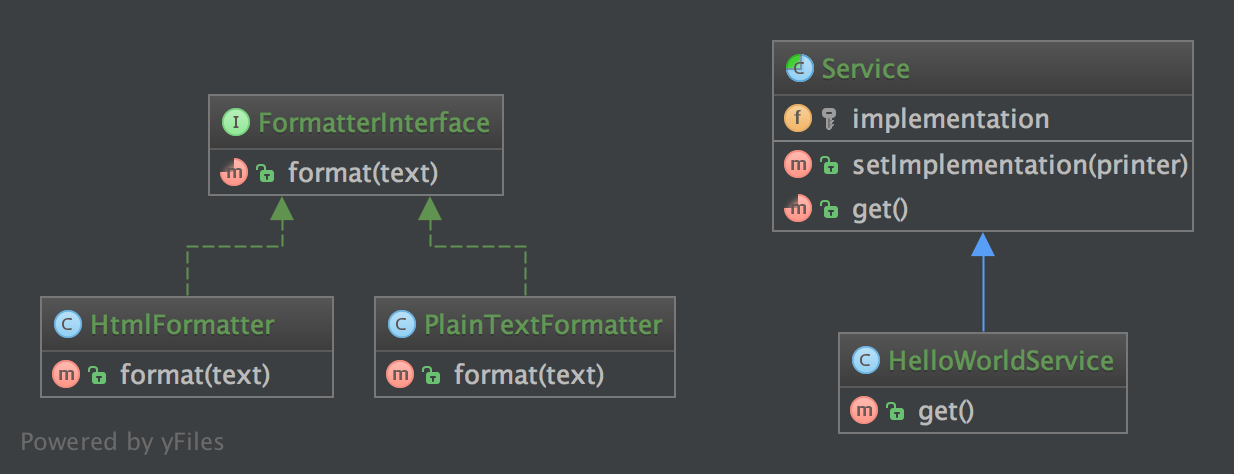
return [$this->page, $this->totalPages];

}

}

*14 užduotis*. PAGAL ŽEMIAU PATEIKTĄ KODĄ NUBRAIŽYKITE UML KLASIŲ DIAGRAMĄ.

Diagrama:



# 

# *15 užduotis.* PHP OBJEKTINIO PROGRAMAVIMO ŽINIŲ PATIKRINIMO TESTAS.

# 

# Testo atsakymai

1. a)
2. d)
3. c)
4. b)
5. b)

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

*1 užduotis*. ĮDIEKITE PHP KODO TESTAVIMO ĮRANKĮ PHPUNIT IR NAUDOKITE DOKUMENTACIJĄ JAM IŠBANDYTI.

Diegimo ir konfigūravimo instrukcijos:

Getting Started with PHPUnit 8 .[žiūrėta 2019-06-28]. Prieiga per internetą

<https://phpunit.de/getting-started/phpunit-8.html>

*2 užduotis*. SUKURKITE KLASĘ SKAIČIUOTUVAS.

<?php

declare(strict\_types=1);

use PHPUnit\Framework\TestCase;

final class SkaiciuotuvasTestTest extends TestCase

{

public function testCanBeCreatedObject(): void

{

$skaiciuotuvas = new Skaiciuotuvas();

$this->assertInstanceOf(

Skaiciuotuvas::class, $skaiciuotuvas);

}

}

*3 užduotis. 1 užduotyje* SUKURTAI KLASEI REALIZUOKITE SKAIČIUOTUVO FUNKCIONALUMĄ, KURIS ATLIEKA ARITMETINES OPERACIJAS (+, -, \*, /).

**Skaiciuotuvas.php**

<?php

/\*\*

\* Class Skaiciuotuvas

\*/

class Skaiciuotuvas {

private $a;

private $b;

private $op;

/\*\*

\* @param $a

\* @param $b

\* @param $op

\*

\*/

public function set($a, $b, $op) {

$this->validuoti($a, $b);

$this->a = $a;

$this->b = $b;

$this->op = $op;

}

/\*\*

\* @return float|int|number

\*

\* Šis metodas atlieka standartinius skaičiuotuvo veiksmus, kuriuos gauna iš metodo ->set().

\*/

public function vykdyti() {

switch ($this->op) {

case 'sudetis':

$val = $this->sudetis();

break;

case 'atimtis':

$val = $this->atimtis();

break;

case 'daugyba':

$val = $this->daugyba();

break;

case 'dalyba':

$val = $this->dalyba();

break;

default:

$val = $this->sudetis();

break;

}

return $this->rodyti($val);

}

/\*\*

\* @param $a

\* @param $b

\*/

private function validuoti($a, $b) {

foreach (array($a, $b) as $item) {

if (!is\_numeric($item)) {

echo "Argumentas turi būti skaičius";

}

}

}

/\*\*

\* @return mixed

\*/

private function sudetis() {

return $this->a + $this->b;

}

/\*\*

\* @return mixed

\*/

private function atimtis() {

return $this->a - $this->b;

}

/\*\*

\* @return mixed

\*/

private function daugyba() {

return $this->a \* $this->b;

}

/\*\*

\* @return float|int

\*/

private function dalyba() {

if ($this->b == 0) {

echo "Klaida. Dalyba iš nulio negalima";

}

return $this->a / $this->b;

}

/\*\*

\* @param $val

\* @return string

\*

\* Apvalinimas

\*/

private function rodyti($val) {

return round($val, 5);

}

}

?>

**SkaiciuotuvasTest.php**

<?php

declare(strict\_types=1);

use PHPUnit\Framework\TestCase;

final class SkaiciuotuvasTestTest extends TestCase

{

/\*\*

\* Sudeties testas

\*/

public function testBasicAddition() {

$calc = new Skaiciuotuvas();

$calc->set(2, 2, 'sudetis');

$this->assertEquals($calc->vykdyti(), 4);

}

}

*4 užduotis*. PARAŠYKITE TESTĄ, KURIS PATIKRINA NAUJO VARTOTOJO REGISTRACIJĄ IR JO PROFILIO DUOMENŲ IŠVEDIMĄ 3. Kompetencijos„ Kurti tipinę programinę įrangą“ 3.3. mokymosi rezultato„ Taikyti objektinio programavimo principus programuojant“ *1 užduotyje* SUKURTOS KLASĖS „VARTOTOJAS“ REALIZUOTAIS METODAIS.

**VartotojasTest.php**

<?php

declare(strict\_types=1);

use PHPUnit\Framework\TestCase;

final class VartotojasTest extends TestCase

{

/\*\*

\* Vartotojo registracijos ir

\*/

public function testVartotojoProfilis() {

$vartotojas = new Darbuotojas("Andrius");

$vartotojas->fillPortfolio("paulius@gmail.com", "paulius456TXW","programuotojas","1200EUR","2019-08-12");

$portfolio = $vartotojas->showPortfolio();

$portfolio = implode(",",$portfolio);

var\_dump($portfolio);

$this->assertEquals('Andrius,paulius@gmail.com,,programuotojas,1200EUR,2019-08-12', $portfolio);

}

}

*5 užduotis*. SUKURTAM *2 užduotyje* SKAIČIUOTUVO METODUI, KURIS ATLIEKA SKAIČIAVIMUS,VIETOJ SKAIČIŲ ĮRAŠYKITE TEKSTINIUS SIMBOLIUS.

Įvykus klaidai skaičiuojant, ją apdorokite (naudojant Php exception) ir išveskite pranešimą vartotojui, kad panaudoti neleistini parametrai.

**VartotojasTest.php**

<?php

/\*\*

\* Class Skaiciuotuvas

\*/

class Skaiciuotuvas {

private $a;

private $b;

private $op;

/\*\*

\* @param $a

\* @param $b

\* @param $op

\*

\*/

public function set($a, $b, $op) {

$this->validuoti($a, $b);

$this->a = $a;

$this->b = $b;

$this->op = $op;

}

/\*\*

\* @return float|int|number

\*

\* Šis metodas atlieka standartinius skaičiuotuvo veiksmus, kuriuos gauna iš metodo ->set().

\*/

public function vykdyti() {

switch ($this->op) {

case 'sudetis':

$val = $this->sudetis();

break;

case 'atimtis':

$val = $this->atimtis();

break;

case 'daugyba':

$val = $this->daugyba();

break;

case 'dalyba':

$val = $this->dalyba();

break;

default:

$val = $this->sudetis();

break;

}

return $this->rodyti($val);

}

/\*\*

\* @param $a

\* @param $b

\*/

private function validuoti($a, $b) {

foreach (array($a, $b) as $item) {

if (!is\_numeric($item)) {

throw new \InvalidArgumentException('Argumentas turi būti skaičius.');

}

}

}

/\*\*

\* @return mixed

\*/

private function sudetis() {

return $this->a + $this->b;

}

/\*\*

\* @return mixed

\*/

private function atimtis() {

return $this->a - $this->b;

}

/\*\*

\* @return mixed

\*/

private function daugyba() {

return $this->a \* $this->b;

}

/\*\*

\* @return float|int

\*/

private function dalyba() {

if ($this->b == 0) {

throw new \InvalidArgumentException('Dalyba iš nulio negalima');

}

return $this->a / $this->b;

}

/\*\*

\* @param $val

\* @return string

\*

\* Apvalinimas

\*/

private function rodyti($val) {

return round($val, 5);

}

}

?>

*6 užduotis* 3. KOMPETENCIJOS„ KURTI TIPINĘ PROGRAMINĘ ĮRANGĄ“ 3.3. MOKYMOSI REZULTATO„ TAIKYTI OBJEKTINIO PROGRAMAVIMO PRINCIPUS PROGRAMUOJANT“ *8 užduotyje* SUKURTAME TEKSTINIAME FAILE ĮRAŠYTI TIK SKRYDŽIUS Į BARSELONĄ.

<?php

$my\_file = 'duomenys.txt';

$handle = fopen($my\_file, 'w') or die('Negaliu atidaryti failo: '.$my\_file);

if($data['flight']=="BAR-567" ){

fwrite($handle, $data['flight']);

}else{

throw new Exception('Klaidingas skrydis');

}

*7 užduotis. 6 užduotyje* APDOROJUS KLAIDAS, SĄLYGOS NEATITIKUSIUS SKRYDŽIUS NAUDOJANT ERROR\_LOG FUNKCIJĄ ĮRAŠYKITE Į TEKSTINĮ FAILĄ: KLAIDOS.LOG. <?php

$my\_file = 'duomenys.txt';

$handle = fopen($my\_file, 'w') or die('Negaliu atidaryti failo: '.$my\_file);

if($data['flight']=="BAR-567" ){

fwrite($handle, $data['flight']);

}else{

error\_log("Klaidingas skrydis!", 3, "klaidos.log");

throw new Exception('Klaidingas skrydis');

}

*8 užduotis*. NAUDODAMI VAR\_DUMP FUNKCIJĄ, IŠVESKITE GAUTUS DUOMENIS IŠ FORMOS (sukurtos *4 užduotyje*).

<?php

if(isset($\_POST['submit'])){

var\_dump($\_POST);

}

*9 užduotis*. PAKEISKITE PHP NUSTATYMUS, KAD NEBŪTŲ RODOMI ĮSPĖJIMAI VYKDANT PHP APLIKACIJĄ.

Klaidos turi būti rodomos.

<?php

error\_reporting(E\_ALL & ~E\_NOTICE);

*10 užduotis*. PANAUDOKITE XDEBUG SAVO PROJEKTE HTTP UŽKLAUSŲ ANALIZEI.

Diegimo ir konfigūravimo instrukcijos:

Debugging a PHP HTTP Request .[žiūrėta 2019-06-28]. Prieiga per internetą

<https://www.jetbrains.com/help/phpstorm/debugging-a-php-http-request.html>

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

# 4.1. Mokymosi rezultatas. Vykdyti reikalavimų peržiūros procesą naudojant vartotojo pasakojimo reikalavimų programinei įrangai formatą.

*1* *užduotis.* ATLIKTI PROGRAMINĖS ĮRANGOS FORMATŲ REIKALAVIMUS.

Reikalavimas yra apibrėžtas kaip specifikacija. Panaudojimo atvejis šios specifikacijos yra pateikiamas atskirai.

Reikalavimui ištestuoti gali būti naudojami šie testavimo atvejai:

* Prisijungti prie sistemos nurodant neteisingą slaptažodį slaptažodį -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos nurodant neteisingą slaptažodį, pakartoti kelis kartus, patikrinti, ar paskyra blokuojama;
* Patikrinti, ar yra galimybė atblokuoti paskyrą atstatant slaptažodį (kokiu būdu atblokavimas vyksta, nenurodyta reikalavime);
* Patikrinti visus vartotojo registravimo formos įvedimo laukus (formatus, ilgius, būtinumo požymį);
* Prisijungti prie sistemos su neregistruotu vartotoju -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos su egzistuojančiu vartotoju, suvedant neteisingą slaptažodį -> gaunamas klaidos pranešimas, vartotojas neprijungiamas prie sistemos;
* Prisijungti prie sistemos su registruotu vartotoju -> atidaromas pagrindinis sistemos langas su priregistravimo kortele;
* Registruotam vartotojui pakoreguoti profilio duomenis (kokie konkrečiai duomenys gali būti keičiami nenurodoma):

*2* *užduotis.* ATLIKTI REIKALAVIMŲ PERŽIŪROS PROCESĄ.

**Vartotojo pasakojimas:** Kaip prisiregistravęs sistemos vartotojas noriu išsiregistruoti iš sistemos paspausdamas mygtuką „Atsijungti“, nepriklausomai nuo to, kokį darbą su sistema būčiau dirbęs.

**Reikalavimo paruošimo kriterijai:** vartotojo pasakojimas yra apibrėžtas aiškiai, yra aiškus ir vienareikšmiškai suvokiamas kontekstas. Priėmimo kriterijai yra aiškūs, testuotini.

**Priėmimo kriterijų pavyzdžiai:**

* Atsijungimo nuo sistemos mygtukas rodomas tik prisijungusiems vartotojams;
* Paspaudus “Atsijungti” mygtuką, Sistema atjungia vartotoją, uždaro jo pradėtus darbus ir atvaizduoja pradinį puslapį;
* Jei vartotojas atsijungia nuo sistemos nepabaigęs kokio nors darbo, jo pradėti, bet nepabaigti darbai nėra išsaugomi, juos reikės atlikti iš naujo;
* Atsijungęs vartotojas vėl mato “Prisijungti” mygtuką ir gali registruotis vėl;
* Vartotojas gali atsijungti nuo sistemos nepriklausomai nuo to, kokiame sistemos lange jis yra.

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

*1* *užduotis.* PASITIKRINKITE ŽINIAS APIE FUNKCINIUS REIKALAVIMUS.

Opencart – Paieška (supaprastinti paieškos funkcionalumo reikalavimai)

- Jei į paieškos lauką įvedamas produkto tikslus pavadinimas ir spaudžiamas "Paieška" mytgtukas, sistema pateikia sąrašą visų šio produkto atmainų, turimu pardavime. OK (atitinka)

- jei įvedamas raktažodis pagal produkto kategoriją ir spaudžiamas "Paieška" mygtukas, sistema pateikia atitinkamos kategorijos produktų sąrašą; NOK (Not OK, neatitinka).

Komentaras: Sistema pateikia tik vieną telefoną, nors <Phones & PDA‘s> kategorijoje yra daugiau produktų. Priežastis gali būti netinkama produkto konfigūracija.

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

Komentaras: Nieko neįvedus į paieškos laukelį ir paspaudus paieškos mygtuką, sistema nieko nepateikia. Klaida gali būti arba reikalavimuose (netinkamai aprašyti, nesuprasti), arba implementacijoje, arba CMS paieškos modulio konfigūracijoje.

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

Komentaras: Bet, jei įvesime tik fragmentą, kuris atitiks kurio nors egzistuojančio produkto pavadinimo dalį, šis produktas bus atvaizduotas paieškos rezultatuose.

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

Komentaras: Pagal nutylėjimą produktai nėra skirstomi puslapiais. Tačiau turi būti patikrinti CMS paieškos modulio nustatymai. Gali būti, kad puslapiavimas yra implementuotas, tačiau neįjungtas.

- Paieškos rezultatai yra surūšiuoti abėcėlės tvarka. OK

*2* *užduotis.* PASITIKRINKITE ŽINIAS APIE NEFUNKCINIUS(TECHNINIAI, SAUGUMO) REIKALAVIMUS.

1. Funkciniai reikalavimai testuoja sistemos funkcijas (skaičiavimus, logiką ir pan), nefunkciniai – sistemos charakteristikas (greitį, išvaizdą, saugumą ir pan.)
2. Saugumo, našumo, atsparumo, suderinamumo, panaudojamumo, t.t.
3. Iš esmės, atliekant testavimą su skirtingomis naršyklėmis, testuojame svetainės išvaizdą ir funkcionalumą įvairiose naršyklėse ir įrenginiuose. Kaip žmonės interpretuoja dalykus skirtingai, taip ir ne visos naršyklės “supranta” dalykus vienodai. Pvz. „Internet Explorer 8“ CSS stiliai gali būti skirtingi nei naujesnėse „Internet Explorer“ ir „Google Chrome“ versijose. Elegantiškas stilius ir efektai, vaizdai, šešėliai ir netgi šriftai, gali labai skirtis. Nors šiuos vaizdinius skirtumus dažnai neįmanoma panaikinti visiškai (ypač su senesnėmis naršyklėmis), testavimo su skirtingomis naršyklėmis ir įrenginiais tikslas yra užtikrinti, kad naudotojai galėtų pasiekti visą turinį ir vykdyti visas pagrindines žiniatinklio programos funkcijas be jokių svarbių problemų.
4. Galima matuoti puslapio našumą /užkrovimo greitį, testuoti turinio pasikeitimus, SEO žymas (angl. tags), atlikti nesudėtingą puslapio derinimą (ang. debugging).

|  |  |
| --- | --- |
| Funkciniai reikalavimai:  b,c, d, f, h, | Nefunkciniai reikalavimai:  a, e, g, i, j |

1. Reikalavimų peržiūra yra labiau nefunkcinis testavimas. Funkcinis testavimas yra tiesiogiai susijęs su sistemos vykdymu. Reikalavimų peržiūra yra statinės analizės dalis, kuri nėra priskiriama prie funkcinio testavimo.
2. a), b), c), d)
3. c)
4. Panaudojamumas – tai naudotojo veiklos veiksmingumas, našumas ir jaučiamas pasitenkinimas, su kuriuo konkretus naudotojas gali pasiekti konkrečių tikslų konkrečiose aplinkose, šiuo atveju, naudodamasis mūsų sistema.
5. a)

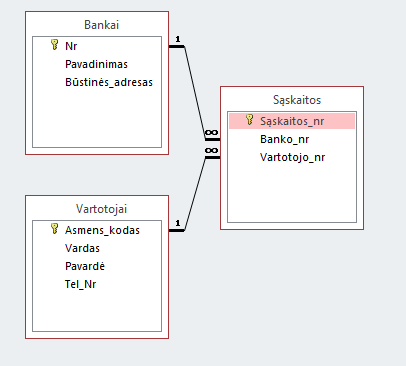
# Modulis „Nesudėtingų duomenų bazių projektavimas ir kūrimas (Php)“

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

# Mokymosi rezultatas. Pateikti reliacinės duomenų bazės schemą.

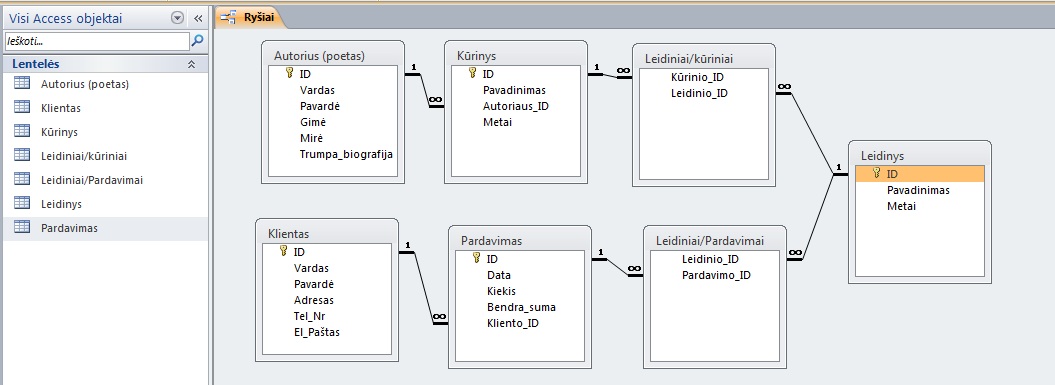
*1 užduotis*. SUPROJEKTUOKITE RELIACINĘ DUOMENŲ BAZĘ (ATVEJIS: LIETUVOS BANKAS).

Schema:



*2 užduotis*. SUPROJEKTUOKITE 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;

Alternatyva - galima panaudoti antrinius indeksus.

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

Priimtini sprendimai tiek su viena, tiek su keliomis kolekcijomis.

Pavyzdžiui sudedant į vieną kolekciją, parduotuvė su jos prekėmis galėtų atrodyti taip:

{

\_id: “V1”,

area: 2500,

address: “Linkmenu g. 3, Vilnius”,

items: [

{ code: 11559245, name: “Skalsioji duona”, unit\_weight: 1.2, unit\_price: 1.7, count: 11 },

...

]

}

*5 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS:

db.getCollection('shops').find({\_id: "V1"});

1. db.getCollection('shops').find({\_id: "V1"}, {items: 1});
2. db.getCollection('shops').find({"items.code": 23});
3. db.getCollection('shops').aggregate([{$unwind: "$items"}, {$group: {\_id: "$\_id", total: {$sum: "$items.count"}}}]);

Jei modeliuojama parduotuvė su prekėmis viename dokumente, tai tiek užklausa #1, tiek #2 grąžins visas prekes.

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

Parduotuvės ir prekės modeliuojamos kaip viršūnės su skirtingomis žymomis ir susiejamos ryšiu.

create (V1:Shop {code: 'K1', area: 3000, address: 'Kauno g. 11'})

create (i11559245:Item {code: 11559245, name: 'Skalsioji duona', unit\_weight: 1.2, unit\_price: 1.8, count: 11})

match (s:Shop),(i:Item)

where s.code = 'K1' AND i.code = 2445

create (s)-[r:HAS]->(i)

return s,i

*7 užduotis*. PARAŠYKITE ŠIAS UŽKLAUSAS:

1. match (s:Shop {code:'V1'}) return s
2. match (s:Shop {code:'V1'})-[r:HAS]-(i:Item) return i
3. match (s:Shop)-[r:HAS]-(i:Item {code: 1445}) return s
4. match (s:Shop)-[r:HAS]-(i:Item) return s,sum(i.count)

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

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

Grafas papildomas naujais elementais pažymėtais žymomis Producer, prekėms sukuriamas ryšys pažymėtas PRODUCEDBY.

create (p2:Producer {name: "Makers", country: "Germany"})

match (i:Item {code: 2445}),(p:Producer {name: "Makers"})

create (i)-[r:PRODUCEDBY]->(p)

return i,p

Užklausos:

1. match (s:Shop)-[r:HAS]-(i:Item)-[pb:PRODUCEDBY]-(p:Producer {country: 'Lithuania'}) return distinct s
2. match (s:Shop)-[r:HAS]-(i:Item)-[pb:PRODUCEDBY]-(p:Producer {country: 'Lithuania'}) where i.unit\_price > 1 return distinct s

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

# 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 užduotis*. PANAUDOKITE TINKAMAS SĄLYGŲ IR GRUPAVIMO SQL KOMANDAS

SQL komandos:

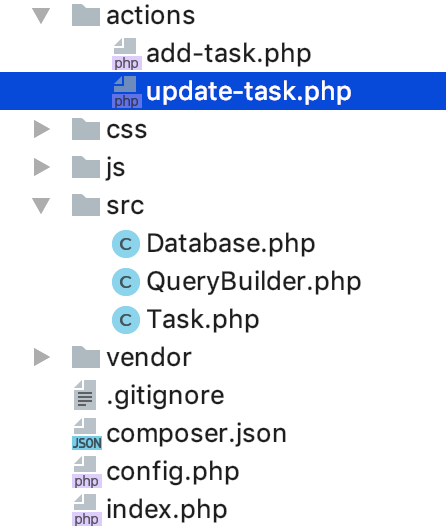
1. SELECT ASMENSKODAS, VARDAS, PAVARDE FROM DARBUOTOJAI WHERE GIMIMOMETAI=’1988-06-15’;
2. SELECT \* FROM DARBUOTOJAI WHERE GIMIMOMETAI < '1988-07-29';
3. SELECT DIRBANUO ,GIMIMOMETAI FROM DARBUOTOJAI WHERE DIRBANUO BETWEEN '2000-10-30' AND 2012-11-11';
4. SELECT VARDAS ,SKYRIAUSPAVADINIMAS ,PROJEKTONUMERIS 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;

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

*1 užduotis*. SUKURKITE PHP (OBJEKTIŠKAI ORIENTUOTĄ) APLIKACIJĄ, SKIRTĄ VARTOTOJO UŽDUOČIŲ SAUGOJIMUI, RODYMUI, ŠALINIMUI IR ATNAUJINIMUI.

Sukurti PHP (objektiškai orientuotą) aplikaciją skirtą vartotojo užduočių saugojimui, rodymui, šalinimui ir atnaujinimui.

**Aplankų ir failų struktūra:**



**PSR-4 konfiguracija faile composer.json:**

{

"autoload": {

"psr-4": {

"Tasker\\": "src"

}

}

}

**Klasių automatinis krovimas faile index.php:**

<?php

require\_once('vendor/autoload.php');

**Prisijungimo prie duomenų bazės parametrai faile config.php:**

<?php

return [

'database' => [

'db' => 'todo',

'username' => 'root',

'password' => '',

'connection' => 'mysql:host=127.0.0.1',

'options' => []

]

];

**Klasė prisijungimui prie duomenų bazės faile Database.php:**

<?php

namespace Tasker;

use PDO;

class Database {

public static function connect($config){

try{

return new PDO (

$config['connection'].';dbname='.$config['db'],

$config['username'],$config['password'],$config['options']

);

} catch (PDOExepction $e){

die($e->getMessage());

}

}

}

**Klasė užklausų vykdymui faile queryBuilder.php:**

<?php

namespace Tasker;

use PDO;

class QueryBuilder

{

private $pdo;

public function \_\_construct($pdo)

{

$this->pdo = $pdo;

}

public function selectAll($table)

{

try {

$statment = $this->pdo->prepare("SELECT \* FROM {$table}");

$statment->execute();

} catch (PDOException $e) {

echo "Duomenys negauti";

exit;

}

return $statment->fetchAll();

}

public function selectById($table, $id){

try {

$statment = $this->pdo->prepare("SELECT \* FROM {$table} WHERE id=:id");

$statment->execute([

':id'=>$id

]);

$data = $statment->fetchAll();

} catch (PDOException $e) {

echo "Duomenys negauti";

exit;

}

return $data;

}

public function addTask($table, $name, $description){

$sql = "INSERT INTO {$table} (name, description, isComplete)

VALUES (:name, :description, 0)";

try {

$statment = $this->pdo->prepare($sql);

$statment->execute([

':name'=>$name,

':description'=>$description,

]);

} catch (PDOException $e) {

echo "Duomenu iterpti nepavyko";

exit;

}

}

function deleteTask($table, $id)

{

try {

$statment = $this->pdo->prepare("DELETE FROM {$table} WHERE id = :id");

$statment->execute([

':id'=>$id

]);

} catch (\PDOException $e) {

echo "Duomenu pasalinti nepavyko";

exit;

}

}

public function updateTask($table, $id, $name, $description, $isComplete){

try {

$statment = $this->pdo->prepare("UPDATE {$table} SET name = :name, description = :description, isComplete = :isComplete WHERE id = :id");

$statment->execute([

':id'=>$id,

':name'=>$name,

':description'=>$description,

':isComplete'=>$isComplete

]);

} catch (PDOException $e) {

echo "Duomenu atnaujinti nepavyko";

exit;

}

}

}

**Naujos užduoties sukūrimas faile add-task.php:**

<div class="modal fade" id="taskModal" tabindex="-1" role="dialog" aria-labelledby="taskModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Add new task</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<!--Add task-->

<form method="POST">

<div><?php echo $error?></div>

<div class="form-group">

<label for="taskName">Task name</label>

<input type="text" class="form-control" id="taskName" aria-describedby="taskName" name="taskName" placeholder="Enter task name">

</div>

<div class="form-group">

<label for="taskDescription">Task description</label>

<input type="text" class="form-control" id="taskDescription" name="description" placeholder="Description">

</div>

<button type="submit" name="submit" class="btn btn-primary">Add</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

**Užduoties atnaujinimas faile update-task.php:**

<?php

require\_once('./vendor/autoload.php');

use Tasker\Database;

use Tasker\QueryBuilder;

$config = require './config.php';

$pdo = Database::connect($config['database']);

$query = new QueryBuilder($pdo);

# Single task item

if (isset($\_GET['taskID'])) {

$data = $query->selectById('tasks', $\_GET['taskID']);

}

# Update task

$isCompleted = null;

if (isset($\_POST['update-task'])) {

if (isset($\_POST['isCompleted'])) {

$isCompleted = 1;

} else {

$isCompleted = 0;

}

$query->updateTask('tasks', htmlspecialchars($\_GET['taskID']), htmlspecialchars($\_POST['taskNameUpdated']), htmlspecialchars($\_POST['descriptionUpdated']), $isCompleted);

header("Location:index.php");

}

?>

<div class="container">

<!--Update task-->

<div class="d-flex p-4 todo-list-wrapper">

<?php foreach ($data as $item) : ?>

<form method="POST">

<div class="form-group">

<label for="taskName">Task name</label>

<input type="text" class="form-control" id="taskNameUpdated" aria-describedby="taskNameUpdated"

name="taskNameUpdated" placeholder="Enter task name" value="<?= $item['name'] ?>">

</div>

<div class="form-group">

<label for="taskDescription">Task description</label>

<input type="text" class="form-control" id="taskDescriptionUpdated" name="descriptionUpdated"

placeholder="Description" value="<?= $item['description'] ?>">

</div>

<div class="form-group">

<label for="taskDescription">Mark as completed</label>

<input type="checkbox" id="isTaskCompleted" name="isCompleted" placeholder="Description">

</div>

<button type="submit" name="update-task" class="btn btn-primary">Update</button>

</form>

<?php endforeach; ?>

</div>

</div>

</div>

**Užduočių rodymas, šalinimas, būsenos atnaujinimas faile index.php:**

<?php

require\_once('vendor/autoload.php');

use Tasker\Database;

//use Tasker\Task;

use Tasker\QueryBuilder;

$config = require 'config.php';

$pdo = Database::connect($config['database']);

$query = new QueryBuilder($pdo);

$allTasks = $query->selectAll('tasks');

# Add new task

$error = "";

if (isset($\_POST['submit'])) {

if (empty($\_POST['taskName']) || empty($\_POST['description'])) {

$error = "Yra neuzpildytu lauku";

} else {

$query->addTask('tasks', htmlspecialchars($\_POST['taskName']), htmlspecialchars($\_POST['description']));

header("Location:index.php");

}

}

# Delete task

if (isset($\_GET['action']) && $\_GET['action'] == "delete") {

$query->deleteTask('tasks', htmlspecialchars($\_GET['id']));

header("Location:index.php");

}

?>

<!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>Todo - php-oop</title>

<!--FontAwesome-->

<script defer src="https://use.fontawesome.com/releases/v5.8.1/js/regular.js"

integrity="sha384-Uc9toywOA44owltk1MWl0lQZ+L0mBzJkLQcdif6+JtG9izvok9DLJtCZX57Uq3k2" crossorigin="anonymous">

</script>

<script defer src="https://use.fontawesome.com/releases/v5.8.1/js/fontawesome.js"

integrity="sha384-EMmnH+Njn8umuoSMZ3Ae3bC9hDknHKOWL2e9WJD/cN6XLeAN7tr5ZQ0Hx5HDHtkS" crossorigin="anonymous">

</script>

<!--Bootstrap-->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<!--Custom style-->

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

</head>

<body>

<nav class="navbar navbar-dark bg-dark justify-content-center">

<h3 class="main-title">TODO app</h3>

</nav>

<?php if (isset($\_GET['action']) && $\_GET['action'] == "update") : ?>

<?php include "./actions/update-task.php"; ?>

<?php else : ?>

<div class="container">

<div class="d-flex justify-content-center mt-4">

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#taskModal">

+ Add task

</button>

</div>

<!--List-->

<div class="d-flex p-4 todo-list-wrapper">

<ul class="list-group todo-list flex-fill">

<?php foreach ($allTasks as $item) : ?>

<?php if ($item['isComplete'] == true) : ?>

<li class="list-group-item d-flex is-complete">

<?php elseif ($item['isComplete'] == false) : ?>

<li class="list-group-item d-flex">

<?php endif; ?>

<div class="round">

<?php if ($item['isComplete'] == true) : ?>

<input type="checkbox" id="checkbox-<?= $item['id'] ?>" checked="checked" />

<?php elseif ($item['isComplete'] == false) : ?>

<input type="checkbox" id="checkbox-<?= $item['id'] ?>" />

<?php endif; ?>

<label for="checkbox-<?= $item['id'] ?>"></label>

</div>

<div class="task-content pl-3">

<h5 class="task-name" contenteditable="true"><?= $item['name'] ?></h5>

<p contenteditable="true"><?= $item['description'] ?></p>

</div>

<div class="edit-content d-flex flex-column justify-content-around pl-4">

<div class="edit">

<a class="update-task" href="?action=update&taskID=<?= $item['id'] ?>">

<i class="far fa-edit"></i>

</a>

</div>

<div class="remove">

<a class="delete-task" href="?action=delete&id=<?= $item['id'] ?>">

<i class="far fa-trash-alt"></i>

</a>

</div>

</div>

</li>

<?php endforeach ?>

</ul>

</div>

</div>

<?php endif; ?>

<!--Add new task BS4 modal-->

<?php include "./actions/add-task.php" ?>

<!--Custom script-->

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

<!--BS scripts-->

<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/1.14.7/umd/popper.min.js"

integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous">

</script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"

integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous">

</script>

</body>

</html>

# 

# Modulis „Programavimo aplinkos ir kūrimo proceso valdymas (Php)“

# 1. Kompetencija. Naudoti tarnybinių stočių operacines sistemas.

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

*1 užduotis*. SUSIPAŽINTI SU 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*. SUSIPAŽINTI SU SERVERIO IŠTEKLIAIS.

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*. SUSIPAŽINTI SULINUX OS PARINKIMU.

Teisingas atsakymas formuojamas https://distrochooser.de/en sistemos pagalba. Atsakymas turi būti pirmi tris Linux OS pavadinimai, kuriuos pasiūlė sistema.

*4 užduotis*. SUSIPAŽINTI SU XUBUNTU OS DIEGIMU VIRTUALIOJE APLINKOJE.

Užduotis atlikta pagal šias instrukcijas:

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

*5 užduotis*. SUSIPAŽINTI SU XUBUNTU OS KONFIGŪRAVIMU.

Darbas vykdomas pagal lektoriaus nurodymus. Darbas vykdomas Xubuntu OS aplinkoje. Konfigūruojame klaviatūrą: en, lt. Konfigūruojame XFce išvaizdą: pasirenkame norimą spalvinę temą.

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

*1 užduotis.* DIRBTI 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*. ADMINISTRUOTI LINUX OS PAKETUS.

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*. ĮDIEKITE IR KONFIGŪRUOKITE WEB SERVERĮ.

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*. PRISIJUNGTI PRIE LINUX OS NUOTOLINIU BŪDU.

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.

# 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*. KOKS YRA REKOMENDUOJAMAS SCRUM KOMANDOS DYDIS?

(4) 7 plius ar minus 2

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

(3) 15 minučių

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

(1) Komandos, kurios turėtų susiderinti tarpusavyje

# 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?

(2) Klientas išsako savo lūkesčius projekte

*2 užduotis.* REIKALAVIMŲ ANALIZĖ NEAPIMA JŲ ATSEKAMUMO.

(2) Ne, apima.

(Explanation: Requirements traceability is concerned with documenting the life of a requirement and providing bi-directional traceability between various associated requirements, hence requirements must be traceable.)

*3 užduotis*. REIKALAVIMŲ ANALIZĖ YRA ITERATYVUS PROCESAS.

(1) Taip, yra.

(Explanation: Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized solutions can satisfy customer requirements.)

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

(4) Svarbiausi darbai sąrašo pradžioje, mažiausiai svarbūs gale

*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*. KAIP VERTINTUMĖTE TOKĮ VARTOTOJO PASAKOJIMĄ? „KAIP PARDAVIMO AGENTAS, AŠ NORIU TURĖTI KLIENTŲ PAIEŠKOS GALIMYBĘ KAD GALĖČIAU RASTI SAVO KLIENTUS GREITAI IR LENGVAI“.

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

# Mokymosi rezultatas. Naudoti projekto eigos valdymo principus.

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

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

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

(3) Kiekvieno sprinto metu

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

(1) Testuoti produktą

ir

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

*4 užduotis.* ĮMONĖS VADOVAS PAPRAŠO KOMANDOS NARIO ATLIKTI DARBĄ, KURIS NEĮEINA Į VYKSTANČIO SPRINTO NUMATYTĄ APIMTĮ. KĄ TOKIU ATVEJU TURĖTŲ DARYTI KOMANDOS NARYS?

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

*5 užduotis.* KADA BAIGIAMAS SPRINTO VYKDYMAS?

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

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

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

*1 užduotis*. ĮSIDIEKITE PHP INTEGRUOTĄ PROGRAMAVIMO APLINKĄ PHP STORM.

Diegimo instrukcijos:

Install Php storm .[žiūrėta 2019-06-28]. Prieiga per internetą

<https://www.jetbrains.com/help/phpstorm/installation-guide.html?section=Windows>

*2 užduotis*. PALEISKITE PHP STORM IR SUKURKITE NAUJĄ PROJEKTĄ.

Instrukcijos:

Creating New Project from Scratch .[žiūrėta 2019-06-28]. Prieiga per internetą

<https://www.jetbrains.com/help/phpstorm/creating-new-project-from-scratch.html>

*3 užduotis*. IDE SUKONFIGŪRUOKITE INTERPRETATORIŲ, KAD KODO LANGE PASPAUDUS NARŠYKLĖS IKONĄ, JIS BŪTŲ ĮVYKDYTAS JOJE.

Instrukcijos:

Configuring Local PHP Interpreters.[žiūrėta 2019-06-28]. Prieiga per internetą

<https://www.jetbrains.com/help/phpstorm/configuring-local-interpreter.html>

*4 užduotis.* SUKONFIGŪRUOKITE GIT KODO VERSIJAVIMO SISTEMĄ IDE .

Instrukcijos:

Set up a GIT.[žiūrėta 2019-06-28]. Prieiga per internetą

<https://www.jetbrains.com/help/phpstorm/set-up-a-git-repository.html?section=Windows%20or%20Linux>

*5 užduotis*. NAUDOJANT TERMINALO LANGĄ ESANTĮ IDE PANAUDOKITE COMPOSER PHP MAILER BIBLIOTEKOS ĮDIEGIMUI PROJEKTE

$ composer require phpmailer/phpmailer

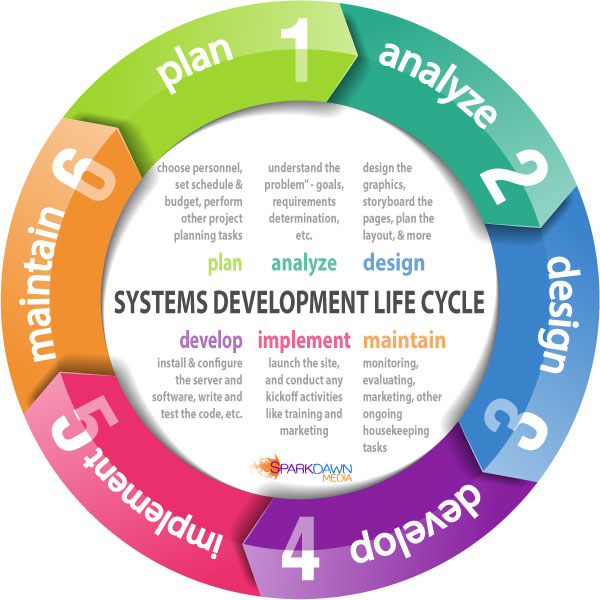
### 3.2. Mokymosi rezultatas. Sekti programavimo darbų vykdymą naudojant komandinio darbų planavimo sistemas.

*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 VARTOTOJUI.

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.>

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

*1 užduotis*. TEORINIŲ ŽINIŲ APIE VERSIJAVIMĄ TESTAS.

1. c)

2. b)

3. c)

*2 užduotis*. PARUOŠKITE GIT VERSIJAVIMO ĮRANKĮ.

Diegimo instrukcijos:

Getting Started - *Installing Git*.[žiūrėta 2019-06-28]. Prieiga per internetą

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

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

$ git config --global user.name "Vardenis Pavardenis"

$ git config --global user.email [vardenis@gmail.com](mailto:vardenis@gmail.com)

$ git config --global core.editor emacs

*4 užduotis*. SUKURKITE NAUJĄ PROJEKTO APLANKĄ.

$ mkdir projektas2

$ cd projektas2

$ git init

*5 užduotis*. PROJEKTO APLANKE PRIDĖKITE FAILĄ INDEX.PHP.

$ mkdir projektas2

$ cd projektas2

$ touch index.php

*6 užduotis.* PROJEKTO APLANKE PRIDĖKITE FAILĄ INDEX.PHP.

index.php

<?php

echo date("Y-m-d");

$ git add index.php

$ git commit –m

$ git commit -m "Create php file with date output function"

Projekto aplanke pridėkite failą index.php. Jame išveskite šios dienos datą. Pridėkite sukurtą failą į repositoriją. Užfiksuokite pakeitimus kodo versijavimo sistemoje su prasmingą informacija apie pridėtą kodą.

*7 užduotis*. PATIKRINKITE GIT ŽURNALE AR *6 užduotyje* DARYTI PAKEITIMAI BUVO UŽFIKSUOTI.

$ git log

*8 užduotis*. INDEX.PHP FAILE PARAŠYKITE KODĄ, KURĮ ĮVYKDŽIUS BUS IŠVESTAS DIENŲ SKAIČIUS IKI PROGRAMUOTOJŲ DIENOS.

index.php

<?php

$devDay = date('Y-09-13');

$today = date('Y-m-d');

if ($today > $devDay) {

$devDay = date('Y-09-13', strtotime($devDay . ' +1 year'));

$date1 = $devDay;

$date2 = $today;

} else {

$date1 = $today;

$date2 = $devDay;

}

echo (strtotime($devDay) - strtotime($today)) / 86400;

$ git add index.php

$ git commit -m "Calculate days until international developers day"

*9 užduotis*. INDEX.PHP FAILE PARAŠYKITE KODĄ, KURĮ ĮVYKDŽIUS BUS IŠVESTAS DIENŲ SKAIČIUS IKI KALĖDŲ.

index.php

<?php

$christmas = date('Y-12-25');

$today = date('Y-m-d');

if ($today > $christmas) {

$christmas = date('Y-12-25', strtotime($christmas . ' +1 year'));

$date1 = $christmas;

$date2 = $today;

} else {

$date1 = $today;

$date2 = $christmas;

}

echo (strtotime($christmas) - strtotime($today)) / 86400;

$ git add index.php

$ git commit -m "Calculate days until christmas day"

*10 užduotis*. NAUDOJAT KODO VERSIJAVIMO SISTEMOS PRIEMONES SUGRĄŽINKITE *8 užduoties* KODO VERSIJĄ.

$ git log

commit da15871eb59abb1e543899a7a18cb8698498c13c (HEAD -> master)

Author: Martynas - netface <m.kaselionis@martinetas.lt>

Date: Fri Jul 5 12:36:28 2019 +0300

Calculate days until dev day

commit 5df192df4ddf9d2549b07a6aa533efd3c37274ac

Author: Martynas - netface <m.kaselionis@martinetas.lt>

Date: Fri Jul 5 12:31:06 2019 +0300

Calculate days until christmas day

$ git checkout 5df192df4ddf9d2549b07a6aa533efd3c37274ac

*11 užduotis*. PATIKRINKITE, KURIOJE ŠAKOJE (BRANCH) ŠIUO METU ESATE.

$ git branch

*12 užduotis*. SUKURKITE NAUJĄ ŠAKĄ PAVADINIMU BIRHDAY.

$ git branch birhday

$ git checkout birhday

<?php

if(isset($\_POST['submit']))

{

$input=$\_POST['birhday'];

$birhday = date("Y-$input");

$today = date('Y-m-d');

if ($today > $birhday) {

$birhday = date("Y-$input" , strtotime($birhday . ' +1 year'));

$date1 = $birhday;

$date2 = $today;

} else {

$date1 = $today;

$date2 = $birhday;

}

echo (strtotime($birhday) - strtotime($today)) ;

}

?>

<!doctype html>

<html lang="lt">

<head>

<meta charset="UTF-8">

<title>Gimtadienio skaičiuoklė</title>

</head>

<body>

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

<input type="text" name="birhday" placeholder="Įveskite gimimo mėnėsį ir dieną. Formatas: mm:dd">

<button type="submit" name="submit">Skaičiuoti</button>

</form>

</body>

</html>

*13 užduotis*. ŠAKĄ (BRANCH) BIRHDAY SUJUNKITE SU ŠAKA (BRANCH) MASTER.

$ git checkout master

$ git merge birhday

*14 užduotis*. PAŠALINKITE BIRHDAY ŠAKĄ (BRANCH)

$ git branch –d birhday

*15 užduotis*. GITHUB.COM SUKURKITE NUOTOLINĘ REPOSITORIJĄ.

Github repositorijos kūrimas

Instrukcijos:

Create a repo. [žiūrėta 2019-06-28]. Prieiga per internetą

<https://help.github.com/en/articles/create-a-repo>

*16 užduotis*. PRIE SAVO PROJEKTO PRIDĖKITE NUOTOLINĘ REPOSITORIJĄ IR IŠSIŲSKITE VISUS KODO PAKEITIMUS Į JĄ.

$ git remote add origin https://github.com/martynas-kaselionis/PHP-kitm.git

$ git push origin master

*17 užduotis*. GITHUB.COM SURASKITE JUMS PATINKANTĮ PHP PROJEKTĄ IR JĮ NUKLONUOKITE.

$ git clone https://github.com/JREAM/design-patterns.git .

**Modulis „Įvadas į darbo rinką“**

## TESTO ATSAKYMAI

1. c)

2. d)

3. d)

4. d)

5. b)

6. b)

7. b)

8. d)

9. c)

10. a)

11. b)

12. a)

13. c)

14. b)

15. a)

16. b)

17. a)

18. b)

19. b)

20. a)

21.

<?php

use function abs;

use function array\_fill;

use function array\_merge;

use function array\_shift;

use function array\_slice;

use function count;

use function floor;

use function intdiv;

use function log10;

use function max;

use function range;

final class SortComplete

{

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function bubble(array $input): array

{

$noSwap = true;

$length = count($input) - 1;

while ($noSwap) {

$noSwap = false;

for ($i = 0; $i < $length; ++$i) {

if ($input[$i] > $input[$i + 1]) {

self::swap($input[$i], $input[$i + 1]);

$noSwap = true;

}

}

}

return $input;

}

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function selection(array $input): array

{

for ($i = 0, $min = $i, $length = count($input); $i < $length; $min = ++$i) {

for ($j = $i + 1; $j < $length; ++$j) {

if ($input[$j] < $input[$min]) {

$min = $j;

}

}

if ($i !== $min) {

self::swap($input[$i], $input[$min]);

}

}

return $input;

}

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function insertion(array $input): array

{

foreach ($input as $i => $value) {

for ($j = $i; $j > 0 && $input[$j - 1] > $value; --$j) {

$input[$j] = $input[$j - 1];

}

$input[$j] = $value;

}

return $input;

}

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function merge(array $input): array

{

$length = count($input);

if ($length <= 1) {

return $input;

}

$middle = intdiv($length, 2);

$left = array\_slice($input, 0, $middle);

$right = array\_slice($input, $middle);

$left = self::merge($left);

$right = self::merge($right);

return self::merger($left, $right);

}

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function quick1(

array &$input,

?int $left = 0,

?int $right = null

): array {

$right = $right ?? count($input) - 1;

if ($right - $left < 1) {

return [];

}

$pivotIndex = self::pivot($input, $left, $right);

self::quick1($input, $left, $pivotIndex - 1);

self::quick1($input, $pivotIndex + 1, $right);

return $input;

}

/\*\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function quick2(

array &$input,

?int $left = 0,

?int $right = null

): void {

$pivot = $left;

$right = $right ?? count($input) - 1;

if ($right - $left < 1) {

return;

}

// Avoid bad performance when sorting an already sorted array

self::swap($input[intdiv($left + $right, 2)], $input[$right]);

for ($i = $left; $i < $right; ++$i) {

if ($input[$i] < $input[$right]) {

self::swap($input[$i], $input[$pivot]);

++$pivot;

}

}

self::swap($input[$pivot], $input[$right]);

self::quick2($input, $left, $pivot - 1);

self::quick2($input, $pivot + 1, $right);

}

/\*\*

\* Only for unsigned integers from 0 to n.

\*

\* @param mixed[] $input

\*

\* @return mixed[]

\*/

public static function radix(array $input): array

{

$digitsMax = self::mostDigits($input);

foreach (range(0, $digitsMax) as $i) {

$bucket = array\_fill(0, 10, []);

foreach ($input as $item) {

$digit = self::getDigit($item, $i);

$bucket[$digit][] = $item;

}

$input = array\_merge(...$bucket);

}

return $input;

}

/\*\* @param int[] $input \*/

private static function mostDigits(array $input): int

{

$maxDigits = 0;

foreach ($input as $iValue) {

$maxDigits = max($maxDigits, self::digitCount($iValue));

}

return $maxDigits;

}

private static function digitCount(int $value): int

{

if ($value === 0) {

return 1;

}

return (int) floor(log10(abs($value))) + 1;

}

private static function getDigit(int $value, int $position): int

{

return floor(abs($value) / 10 \*\* $position) % 10;

}

/\*\*

\* @param mixed $x

\* @param mixed $y

\*/

private static function swap(&$x, &$y): void

{

$tmp = $x;

$x = $y;

$y = $tmp;

}

/\*\*

\* Helper method for Merge sort.

\*

\* @param mixed[] $left

\* @param mixed[] $right

\*

\* @return mixed[]

\*/

private static function merger(array $left, array $right): array

{

$results = [];

while (count($left) && count($right)) {

if ($left[0] < $right[0]) {

$results[] = array\_shift($left);

} else {

$results[] = array\_shift($right);

}

}

return array\_merge($results, $left, $right);

}

/\*\* @param mixed[] $input \*/

private static function pivot(

array &$input,

?int $start = 0,

?int $end = null

): int {

$end = $end ?? count($input) - 1;

$pivot = $input[$start];

$swapIndex = $start;

for ($i = $start + 1; $i <= $end; ++$i) {

if ($input[$i] < $pivot) {

self::swap($input[++$swapIndex], $input[$i]);

}

}

self::swap($input[$start], $input[$swapIndex]);

return $swapIndex;

}

}

Unit testas

<?php

use Exercises\Sort\Complete\SortComplete;

use PHPUnit\Framework\TestCase;

final class SortCompleteTest extends TestCase

{

use GetReflectionMethod;

/\*\* @var int[] \*/

private $input = [220, -32, 405, -114, 0, 23, 5, -1];

/\*\* @var int[] \*/

private $sorted = [-114, -32, -1, 0, 5, 23, 220, 405];

public function testHasMethods(): void

{

self::assertTrue(

method\_exists(SortComplete::class, 'bubble'),

'Class does not have method bubble'

);

self::assertTrue(

method\_exists(SortComplete::class, 'selection'),

'Class does not have method selection'

);

self::assertTrue(

method\_exists(SortComplete::class, 'insertion'),

'Class does not have method insertion'

);

self::assertTrue(

method\_exists(SortComplete::class, 'merge'),

'Class does not have method merge'

);

self::assertTrue(

method\_exists(SortComplete::class, 'merger'),

'Class does not have method merger'

);

self::assertTrue(

method\_exists(SortComplete::class, 'quick1'),

'Class does not have method quick1'

);

self::assertTrue(

method\_exists(SortComplete::class, 'quick2'),

'Class does not have method quick2'

);

self::assertTrue(

method\_exists(SortComplete::class, 'radix'),

'Class does not have method radix'

);

}

public function testBubbleSort(): void

{

self::assertSame($this->sorted, SortComplete::bubble($this->input));

}

public function testSelectionSort(): void

{

self::assertSame($this->sorted, SortComplete::selection($this->input));

}

public function testInsertionSort(): void

{

self::assertSame($this->sorted, SortComplete::insertion($this->input));

}

/\*\* @throws ReflectionException \*/

public function testMergerHelper(): void

{

$left = [1, 9];

$right = [3, 7, 10];

$merger = self::getMethod(SortComplete::class, 'merger');

self::assertSame([1, 3, 7, 9, 10], $merger->invoke(null, $left, $right));

}

public function testMergeSort(): void

{

self::assertSame($this->sorted, SortComplete::merge($this->input));

}

public function testQuick1Sort(): void

{

self::assertSame($this->sorted, SortComplete::quick1($this->input));

}

public function testQuick2Sort(): void

{

SortComplete::quick2($this->input);

self::assertSame($this->sorted, $this->input);

}

public function testRadixSort(): void

{

self::assertSame([ 12, 23, 345, 2345, 5467, 9852 ], SortComplete::radix([23, 345, 5467, 12, 2345, 9852]));

}

}

# Literatūros sąrašas

1. Apache 2.4 *binaries VS16.* [žiūrėta 2019-06-28]. Prieiga per internetą <https://www.apachelounge.com/download/>

2. Atlassian (2019). Create a project. [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/get-started-with-jira-core/create-a-project-917965385.html

3. Atlassian (2019*). Advanced searching*. [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jirasoftwareserver073/advanced-searching-861256227.html

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

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

6. Atlassian (2019). *Atlassian Documentation.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html

7. Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jiracoreserver073/creating-issues-and-sub-tasks-861257329.html

8. Atlassian (2019). *Working with boards.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jiracorecloud/working-with-boards-800712866.html

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

10. Atlassian (2019). *Working with issues in Jira* *Software*. [žiūrėta 2019-05-27]. Prieiga per internetąhttps://www.atlassian.com/agile/tutorials/issues

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

12. Atlassian (2019). *Creating issues and sub-tasks.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jirasoftwarecloud/creating-issues-and-sub-tasks-764478439.html

13. Atlassian (2019). *Working with issues.* [žiūrėta 2019-05-27]. Prieiga per internetą https://confluence.atlassian.com/jiracorecloud/working-with-issues-765593800.html

14. Atlassian (2019). *Watch, share, and comment on issues.* [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jirasoftwarecloud/watch-share-and-comment-on-issues-962349057.html

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

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

17. 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

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

19. Atlassian (2019). *Basic searching*. [žiūrėta 2019-05-27]. Prieiga per internetąhttps://confluence.atlassian.com/jirasoftwarecloud/basic-searching-764478306.html

20. Atlassian (2019). *Searching for issues.* [žiūrėta 2019-05-27]. Prieiga per internetą<https://confluence.atlassian.com/jirasoftwarecloud/searching-for-issues-764478280.html>

21. Binaries sources and releases. [žiūrėta 2019-06-28]. Prieiga per internetą <https://windows.php.net/download/>

22. Composer. [žiūrėta 2019-06-28]. Prieiga per internetą <https://getcomposer.org/download/>

23. Configuring Local PHP Interpreters.[žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jetbrains.com/help/phpstorm/configuring-local-interpreter.html>

24. Create a repo. [žiūrėta 2019-06-28]. Prieiga per internetą <https://help.github.com/en/articles/create-a-repo>

25. Creating New Project from Scratch .[žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jetbrains.com/help/phpstorm/creating-new-project-from-scratch.html>

26. Debugging a PHP HTTP Request .[žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jetbrains.com/help/phpstorm/debugging-a-php-http-request.html>

27. Design Patterns PHP [žiūrėta 2019-06-28]. Prieiga per internetą <https://designpatternsphp.readthedocs.io>

28. Download. [žiūrėta 2019-06-28]. Prieiga per internetą <https://download.sublimetext.com/Sublime%20Text%20Build%203207%20Setup.exe>

29. Getting Started - Installing Git.[žiūrėta 2019-06-28]. Prieiga per internetą <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

30. Getting Started with PHPUnit 8 .[žiūrėta 2019-06-28]. Prieiga per internetą <https://phpunit.de/getting-started/phpunit-8.html>

31. GbmbOrg (2019). *Data Units Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.gbmb.org.>

32. GIT documentation . [žiūrėta 2019-06-28]. Prieiga per internetą <https://git-scm.com/doc>

33. 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

34. Install Php storm .[žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jetbrains.com/help/phpstorm/installation-guide.html?section=Windows>

35. How Laravel implements MVC and how to use it effectively (MAY 24, 2018). *Pusher Tutorials and Articles*. [žiūrėta 2019-05-27]. Prieiga per internetą https://blog.pusher.com/laravel-mvc-use/

36. 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

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

38. How to make Login System in Laravel. Published on Mar 5, 2018. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.youtube.com/watch?v=OUFmwAnFclo>

39. JetBrains s.r.o. (2019). *Learn and Support*. [žiūrėta 2019-06-28]. Prieiga per internetą [https://www.jetbrains.com/phpstorm/documentation](https://www.jetbrains.com/idea/documentation/)

40. JGraph Ltd (2019). *draw.io*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.draw.io.>

41. Lietuvos statistikos departamentas (2019). *Ekonominės veiklos rūšių klasifikatorius*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://osp.stat.gov.lt/static/evrk2.htm>

42. 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>

43. MDN web docs . [žiūrėta 2019-06-28]. Prieiga per internetą <https://developer.mozilla.org/en-US/>

44. MySQL server. [žiūrėta 2019-06-28]. Prieiga per internetą <https://dev.mysql.com/downloads/mysql/5.7.html#downloads>

45. Oracle (2019). *Lambda Expressions*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html.>

46. PHP unit testing [žiūrėta 2019-06-28]. Prieiga per internetą <https://phpunit.de/>

47. PHP 7 - Installation on Windows with Apache. [žiūrėta 2019-06-28]. Prieiga per internetą <https://www.tutorialspoint.com/php7/php7_installation_windows_apache.htm>

48. Set up a GIT.[žiūrėta 2019-06-28]. Prieiga per internetą <https://www.jetbrains.com/help/phpstorm/set-up-a-git-repository.html?section=Windows%20or%20Linux>

49. 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/)

50. Tutorials Point (2019). *Number System Conversion*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.tutorialspoint.com/computer_logical_organization/number_system_conversion.htm>

51. Vilniaus universitetas (2019). *Apie skaičiavimo sistemas.* [žiūrėta 2019-05-27]. Prieiga per internetą <https://www.mif.vu.lt/ljmm/rasmenys/skaiciavimo_sistemos.htm>

52. Wikipedia (2019). *List of numeral systems*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/List_of_numeral_systems>

53. Wikipedia (2019). *Units of information*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Units_of_information>

54. Wikipedia (2019). *Data rate units*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Data-rate_units>

55. Wikipedia (2019). *Boolean Algebra*. [žiūrėta 2019-05-27]. Prieiga per internetą <https://en.wikipedia.org/wiki/Boolean_algebra>

56. Wikipedia (2019). *Projektavimo pavyzdys.* [žiūrėta 2019-05-27]. Prieiga per internetą https://lt.wikipedia.org/wiki/Projektavimo\_pavyzdys

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

58. Wikipedia (2019). *Model View Controller*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Model-view-controller

59. Wikipedia (2019). *SOLID*. [žiūrėta 2019-05-27]. Prieiga per internetą [*https://en.wikipedia.org/wiki/SOLID*.](https://en.wikipedia.org/wiki/SOLID.)

60. Wikipedia (2019). *Don’t repeat yourself*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Don’t\_repeat\_yourself

61. Wikipedia (2019). *Separation of Concerns*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Separation\_of\_concerns

62. Wikipedia (2019). *Code reuse*. [žiūrėta 2019-05-27]. Prieiga per internetą https://en.wikipedia.org/wiki/Code\_reuse