

**TALLINNA TEHNIKAÜLIKOOL**  
**INFORMAATIKAINSTITUUT**

## **LAB 2 “TESTIPÕHINE ARENDUS”**

Projekt õppeaines “Tarkvara protsessid ja kvaliteet” (ITB8826)

**Autorid, õpperühm:**

**Triinu Tammer IABM**

**Berit Põldoja IABM**

**Sylvia Krupp IABM**

**Esitatud: 04.11.2018**

**Juhendaja: Jaak Tepandi**

Tallinn 2018

# Sisukord

<b>1</b>	<b>Testipõhine arendus .....</b>	<b>4</b>
1.1	Nõuded .....	4
1.1.1	Kasutajaks registreerumine .....	4
1.1.2	Sisse logimine .....	5
1.1.3	Sündmuse otsing .....	6
1.2	Testipõhise arenduse tsükkel .....	6
1.2.1	Kasutajaks registreerumine .....	7
1.2.2	Sisse logimine .....	10
1.3	Sündmuse otsing .....	11
1.4	Testijuhtumite lähtekoodid ja testi tulemuste raportid .....	12
1.4.1	TC 1 "Kasutajanime kontroll (Pass)" .....	13
1.4.2	TC 2 "Kasutajanime kontroll (Fail)" .....	13
1.4.3	TC 3 "Parooli kontroll (Pass)" .....	14
1.4.4	TC 4 "Parooli kontroll (Fail)" .....	14
1.4.5	TC 5 "Kordus parooli kontroll (Pass)" .....	15
1.4.6	TC 6 "Kordu parooli kontroll (Fail)" .....	15
1.4.7	TC 7 "E-maili validatsioon (Pass)" .....	16
1.4.8	TC 8 "E-maili validatsioon (Fail)" .....	16
1.4.9	TC 9 "Telefoni numbri validatsioon (Pass)" .....	17
1.4.10	TC 10 "Telefoni numbri validatsioon (Fail)" .....	17
1.4.11	TC 11 "Kas leidub sisestatud Kasutajanimi ja Parool (Pass)" .....	18
1.4.12	TC 12 "Kas leidub sisestatud Kasutajanimi ja Parool (Fail)" .....	19
1.4.13	TC 13 "Kas kasutaja ei ole robot (Pass)" .....	20
1.4.14	TC 14 "Kas kasutaja ei ole robot (Fail)" .....	20
1.4.15	TC 15 "Sündmuse otsing (Pass)" .....	21
1.4.16	TC 16 "Sündmuse otsing (Fail)" .....	21
<b>2</b>	<b>Koodi katvus .....</b>	<b>23</b>
2.1	Programm „Matemaatilised arvutused“ nõuded lühidalt .....	23
2.2	Programmi kood .....	23
2.3	Koodi katvuse testid .....	23

2.3.1	T1 „Liitmine“ .....	24
2.3.2	T2 „Lahutamine“ .....	24
2.3.3	T3 „Korrutamine“ .....	24
2.3.4	T4 „Jagamine“ .....	25
2.3.5	T5 „Räsimine“ .....	25
2.3.6	T6 „Fibonacci arvujada“ .....	26
2.3.7	T7 „The Lazy Caterer's jada“ .....	26
2.3.8	T8 „Astendamine“ .....	27
2.3.9	T9 „Astendamise väärtused listis“ .....	27
2.3.10	T10 „Faktoriaal“ .....	27
2.3.11	Testimata programmiosad .....	28
<b>2.4</b>	<b>Testid ja testi tulemite raportid.....</b>	<b>29</b>
2.4.1	T1 „Liitmine“ .....	29
2.4.2	T2 „Lahutamine“ .....	29
2.4.3	T3 „Korrutamine“ .....	29
2.4.4	T4 „Jagamine“ .....	30
2.4.5	T5 „Räsimine“ .....	30
2.4.6	T6 „Fibonacci arvujada“ .....	31
2.4.7	T7 „The Lazy Caterer's jada“ .....	32
2.4.8	T8 „Astendamine“ .....	32
2.4.9	T9 „Astendamise väärtused listis“ .....	32
2.4.10	T10 „Faktoriaal“ .....	33
<b>3</b>	<b>Eeldused tarkvara testimiseks .....</b>	<b>34</b>
<b>3.1</b>	<b>Tarkvara esimese osa testimiseks.....</b>	<b>34</b>
3.1.1	Eeldused esimese osa testimiseks.....	34
<b>3.2</b>	<b>Tarkvara teise osa testimiseks.....</b>	<b>36</b>
3.2.1	Eeldused teise osa testimiseks .....	37
<b>3.3</b>	<b>Piletilevi arhitektuur .....</b>	<b>38</b>
	<b>Projekti liikmete osalus Lab2 .....</b>	<b>40</b>

# 1 Testipõhine arendus

## 1.1 Nõuded

Projekti esimesest tööst “Piletilevi süsteem” valisime välja järgmised kaks nõuet – FR1 ja FR3, mida sai antud töö jaoks kohandatud. Lisaks lisasime uue nõude FR12. Järgnevalt on välja toodud kolme nõude kirjeldus.

### 1.1.1 Kasutajaks registreerumine

Use Case ID	FR1
Use Case Name	Kasutajaks registreerumine
Primary Actor	Piletilevi lehe külastaja
Preconditions	Lehekülastaja soovib registreeruda Piletilevi kasutajaks
Postconditions	Klient on registreeritud Piletilevi kasutajaks
Main Success Scenario	<ol style="list-style-type: none"><li>1. Piletilevi lehe külastaja vajutab ‘Logi sisse’.</li><li>2. Süsteem kuvab kasutajale vormid ‘Registreeru siin’:<ol style="list-style-type: none"><li>a. Kasutajanimi</li><li>b. Parool</li><li>c. Korda parooli</li><li>d. E-mail</li><li>e. Mobiiltelefoni number</li></ol></li><li>3. Piletilevi lehe külastaja sisestab nõutud andmed ning klient vastab küsimusele ‘Ma ei ole robot’ = ‘Jah’.</li><li>4. Süsteem valideerib andmeid:<ol style="list-style-type: none"><li>a. Kasutajanimi ei ole tühi</li><li>b. Parool on vähemalt 6 märki pikk, millest vähemalt 1 märk on number</li><li>c. ‘Korda parool’ vastab ‘Parool’ väärtusele</li></ol></li></ol>

	<p>d. E-mail ei tohi olla tühi ja peab vastama e-maili reeglitele:</p> <ul style="list-style-type: none"> <li>i. sisaldama @ märki</li> <li>ii. sisaldama . märki</li> <li>iii. @ ja . ei tohi olla kõrvuti</li> </ul> <p>e. Mobiiltelefoni number ei ole tühi, vähemalt 7 numbrit pikk</p> <p>5. Eduka valideerimise korral süsteem kuvab Piletilevi lehe külastajale teate kinnituse saatmisest e-posti aadressile.</p>
--	---

### 1.1.2 Sisse logimine

<b>Use Case ID</b>	<b>FR12</b>
<b>Use Case Name</b>	Piletilevi süsteemi sisse logimine
<b>Primary Actor</b>	Piletilevi kasutaja
<b>Preconditions</b>	Piletilevi kasutaja on süsteemis registreeritud
<b>Postconditions</b>	Kasutaja on Piletilevi süsteemi sisse logitud
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Piletilevi kuvab külastajale 'Logi sisse' siin vormi: <ul style="list-style-type: none"> <li>a. Kasutajanimi</li> <li>b. Parool</li> <li>c. Küsimuse 'Ma ei ole robot'</li> </ul> </li> <li>2. Piletilevi lehe külastaja sisestab nõutud andmed ja määrab 'Ma ei ole robot' = 'Jah'.</li> <li>3. Piletilevi külastaja märgistab märkeruudu 'Ma ei ole robot' = 'Jah'.</li> <li>4. Süsteem kontrollib: <ul style="list-style-type: none"> <li>a. Kas sisestatud Kasutajanimi ja Parool leidub süsteemis</li> <li>b. Kas 'Ma ei ole robot' = 'Jah'</li> </ul> </li> </ol>

	5. Eduka süsteemi kontrolli korral kuvatakse kasutajale teade eduka sisse logimise kohta.
--	---

### 1.1.3 Sündmuse otsing

<b>Use Case ID</b>	<b>FR3</b>
Use Case Name	Sündmuse, esineja või toimumiskoha otsing
Primary Actor	Sisse loginud Piletilevi kasutaja, Piletilevi lehe külastaja
Preconditions	1. Kasutaja on Piletilevi keskkonda sisse loginud. või 2. Autentimata Piletilevi lehe külastaja on veebilehel.
Postconditions	1. Kasutajale on kuvatud vähemalt üks otsingu tulem või 2. Kasutajale kuvatakse teade 'Sinu otsing ei andnud tulemust'
Main Success Scenario	1. Süsteem kuvab kasutajale/lehe külastajale lehe otsingu välja. 2. Kasutaja/Lehe külastaja sisestab otsingu välja otsitavad märksõnad. 3. Süsteem otsib ürituste nimekirjast kas leidub üritus, mis vastab kasutaja sisestatud. 4. Süsteem kuvab kasutajale leitud sündmused, mis sisaldavad kasutaja sisestatud.

## 1.2 Testipõhise arenduse tsükkel

Antud kodutöös valisid autorid programmeerimise keeleks Java ning ühiktestide tegemiseks JUnit. Projekti esimene töö (Lab1) on Piletilevi süsteem, mille lähtekood ei ole avalik ning seetõttu töö autorid koostasid ise süsteemi koodi:

- Tarkvara nimi: Piletilevi süsteem (Lab2)

- Versioon: 1.0
- Autor: Triinu Tammer (projektimeeskonna liige)
- Lähtekood on alla laetav aadressilt: [https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps\\_tdyL8Tu1XxfMPhNy\\_LAioLuPpT3Kaa6E0WgDh\\_s\\_mBd2Y](https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps_tdyL8Tu1XxfMPhNy_LAioLuPpT3Kaa6E0WgDh_s_mBd2Y)

Täpsem informatsioon (lähtekood, testid) ja juhend testide käivitamiseks ja projekti ülespanekuks on kirjeldatud peatükis 3.1. *Tarkvara esimese osa testimiseks.*

Testimise põhises arendustsüklis lähtusime järgnevatest sammudest:

1. Testi kirjutamine
2. Testi käivitamine
3. Programmikoodi kirjutamine
4. Testi uuesti käivitamine
5. Programmikoodi refaktoreerimine
6. Testi uuesti käivitamine

Juhul, kui Testi käivitamine ei anna soovitud tulemit, siis korratakse arendustsüklit alates sammust 5.

Järgnevates alampeatükkides on välja toodud testijuhtumid, mis peavad tagastama vastavalt sisendile kas positiivse tulemi või negatiivse tulemi.

## 1.2.1 Kasutajaks registreerumine

### 1.2.1.1 TC 1 “Kasutajanime kontroll (Pass)”

Test Case ID	TC 1 (testUsername)
Nõude kirjeldus	Kasutajanimi ei tohi olla tühi
Sisend	Test
Oodatav väljund	True

#### 1.2.1.2 TC 2 “Kasutajanime kontroll (Fail)”

Test Case ID	TC 2 (testUsername)
Nõude kirjeldus	Kasutajanimi ei tohi olla tühi
Sisend	(tühi)
Oodatav väljund	False

#### 1.2.1.3 TC 3 “Parooli kontroll (Pass)”

Test Case ID	TC 3 (testPassword)
Nõude kirjeldus	Parool on vähemalt 6 tähemärki pikk, millest vähemalt 1 tähemärk on number
Sisend	Malle1
Oodatav väljund	True

#### 1.2.1.4 TC 4 “Parooli kontroll (Fail)”

Test Case ID	TC 4 (testPassword)
Nõude kirjeldus	Parool on vähemalt 6 märki pikk, millest vähemalt 1 märk on number
Sisend	Malle
Oodatav väljund	False

#### 1.2.1.5 TC 5 “Kordus parooli kontroll (Pass)”

Test Case ID	TC 5 (testPasswordMatching)
Nõude kirjeldus	‘Korda parool’ vastab ‘Parool’ väärtusele
Sisend	Parool = Malle1



	Korda parooli = Malle1
Oodatav väljund	True

#### 1.2.1.6 TC 6 “Kordu parooli kontroll (Fail)”

Test Case ID	TC 6 (testPasswordMatching)
Nõude kirjeldus	‘Korda parool’ vastab ‘Parool’ väärtusele
Sisend	Parool = Malle1 Korda parooli = Malle
Oodatav väljund	False

#### 1.2.1.7 TC 7 “E-maili validatsioon (Pass)”

Test Case ID	TC 7 (testEmail)
Nõude kirjeldus	E-mail ei tohi olla tühi ja peab vastama e-maili reeglitele: <ul style="list-style-type: none"> <li>• sisaldama @ märki</li> <li>• sisaldama . märki</li> <li>• @ ja . ei tohi olla kõrvuti</li> </ul>
Sisend	test@hot.ee
Oodatav väljund	True

#### 1.2.1.8 TC 8 “E-maili validatsioon (Fail)”

Test Case ID	TC 8 (testEmail)
Nõude kirjeldus	E-mail ei tohi olla tühi ja peab vastama e-maili reeglitele: <ul style="list-style-type: none"> <li>• sisaldama @ märki</li> <li>• sisaldama . märki</li> <li>• @ ja . ei tohi olla kõrvuti</li> </ul>

Sisend	test@.ee
Oodatav väljund	False

#### 1.2.1.9 TC 9 “Telefoni numbri validatsioon (Pass)”

Test Case ID	TC 9 (testPhone)
Nõude kirjeldus	Mobiiltelefoni number ei ole tühi, vähemalt 7 numbrit pikk
Sisend	5500228
Oodatav väljund	True

#### 1.2.1.10 TC 10 “Telefoni numbri validatsioon (Fail)”

Test Case ID	TC 10 (testPhone)
Nõude kirjeldus	Mobiiltelefoni number ei ole tühi, vähemalt 7 numbrit pikk
Sisend	+5500228
Oodatav väljund	False

### 1.2.2 Sisse logimine

#### 1.2.2.1 TC 11 “Kas leidub sisestatud Kasutajanimi ja Parool (Pass)”

Test Case ID	TC 11 (testUsernamePassword)
Nõude kirjeldus	Kas sisestatud Kasutajanimi ja Parool leidub süsteemis
Sisend	Kasutajanimi = Detsember Parool = "Detsember3";
Oodatav väljund	True

#### 1.2.2.2 TC 12 "Kas leidub sisestatud Kasutajanimi ja Parool (Fail)"

Test Case ID	TC 12 (testUsernamePassword)
Nõude kirjeldus	Kas sisestatud Kasutajanimi ja Parool leidub süsteemis
Sisend	Kasutajanimi = Detsember Parool = "123";
Oodatav väljund	False

#### 1.2.2.3 TC 13 "Kas kasutaja ei ole robot (Pass)"

Test Case ID	TC 13 (testNotRobot)
Nõude kirjeldus	Kas 'Ma ei ole robot' = 'Jah'
Sisend	Jah
Oodatav väljund	True

#### 1.2.2.4 TC 14 "Kas kasutaja ei ole robot (Fail)"

Test Case ID	TC 14 (testNotRobot)
Nõude kirjeldus	Kas 'Ma ei ole robot' = 'Jah'
Sisend	Ei
Oodatav väljund	False

### 1.3 Sündmuse otsing

#### 1.3.1.1 TC 15 "Sündmuse otsing (Pass)"

Test Case ID	TC 15 (testingSearch)
--------------	-----------------------

Nõude kirjeldus	Süsteem otsib ürituste nimekirjast kas leidub üritus, mis vastab kasutaja sisestatule ja kuvab kasutajale leitud sündmused, mis sisaldavad kasutaja sisestatut
Sisend	ED
Oodatav väljund	True

#### 1.3.1.2 TC 16 “Sündmuse otsing (Fail)”

Test Case ID	TC 16 (testingSearch)
Nõude kirjeldus	Süsteem otsib ürituste nimekirjast kas leidub üritus, mis vastab kasutaja sisestatule ja kuvab kasutajale leitud sündmused, mis sisaldavad kasutaja sisestatut
Sisend	sssss
Oodatav väljund	Fail

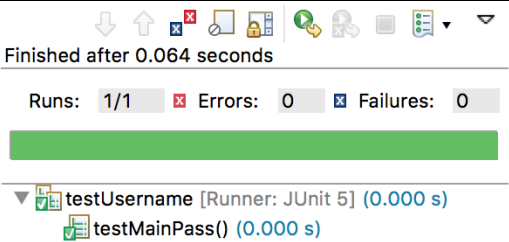
## 1.4 Testijuhtumite lähtekoodid ja testi tulemuste raportid

Programmikoodid on leitavad *Lisa 1* (Kasutajaks registreerumine), *Lisa 2* (Sisse logimine) ja *Lisa 3* (Sündmuse otsing) alt. Lisaks selle on toodud peatükis 3.1. *Tarkvara esimese osa testimiseks* välja, mis, on programmikoodi GitHub aadress.

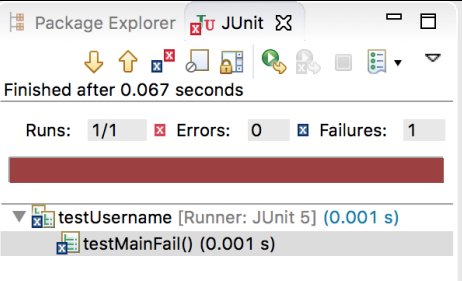
JUnit testide koodis on kirjas sama testitava ärinõude positiivne tsenaarium kui ka negatiivne tsenaarium. Mõlemat tsenaariumi ei saa koos käivitada ja seetõttu tuleb testide käivitamisel tuleb see kood välja kommenteerida (*/\* kood \**), mida ei testita ja see kood välja kommenteerimise *tag*-idest (*/\* kood \*/*) eemaldada, mida soovitakse käivitada.

Järgnevalt on esitletud iga JUnit testi kood ja testi raporti tulem.

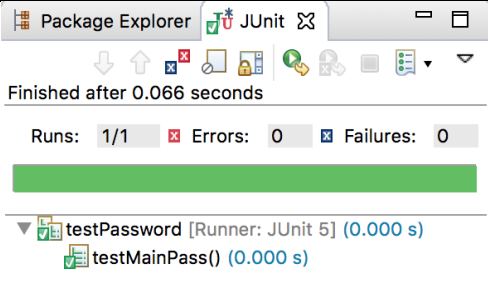
### 1.4.1 TC 1 “Kasutajanime kontroll (Pass)”

Lähtekood	<pre> @Test void testMainPass() {     Boolean username;     Boolean ans = true;     String insertedValue = "Test";     username = Registration.validationUsername(insertedValue);     assertEquals(ans, username); } </pre>
Testi raport	 <p>Finished after 0.064 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 0</p> <p>testUsername [Runner: JUnit 5] (0.000 s)</p> <p>testMainPass() (0.000 s)</p>

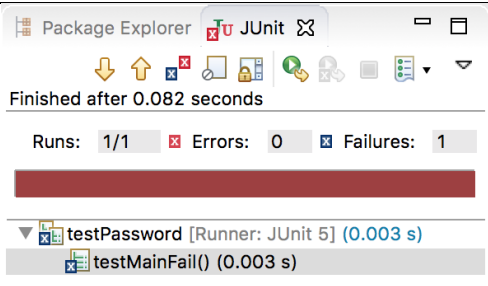
### 1.4.2 TC 2 “Kasutajanime kontroll (Fail)”

Lähtekood	<pre> @Test void testMainFail() {     Boolean username;     Boolean ans = true;     String insertedValue = "";     username = Registration.validationUsername(insertedValue);     assertEquals(ans, username); } </pre>
Testi raport	 <p>Package Explorer JUnit</p> <p>Finished after 0.067 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 1</p> <p>testUsername [Runner: JUnit 5] (0.001 s)</p> <p>testMainFail() (0.001 s)</p>

### 1.4.3 TC 3 “Parooli kontroll (Pass)”

<b>Lähtekood</b>	<pre>@Test void testMainPass() {     Boolean password;     Boolean ans = true;     String passwordValue = "Malle1";     password = Registration.validationPassword(passwordValue);     assertEquals(ans, password); }</pre>
<b>Testi raport</b>	 <p>Package Explorer JUnit</p> <p>Finished after 0.066 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 0</p> <p>testPassword [Runner: JUnit 5] (0.000 s)</p> <p>testMainPass() (0.000 s)</p>

### 1.4.4 TC 4 “Parooli kontroll (Fail)”

<b>Lähtekood</b>	<pre>@Test void testMainFail() {     Boolean password;     Boolean ans = true;     String passwordValue = "Malle";     password = Registration.validationPassword(passwordValue);     assertEquals(ans, password); }</pre>
<b>Testi raport</b>	 <p>Package Explorer JUnit</p> <p>Finished after 0.082 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 1</p> <p>testPassword [Runner: JUnit 5] (0.003 s)</p> <p>testMainFail() (0.003 s)</p>

#### 1.4.5 TC 5 “Kordus parooli kontroll (Pass)”

Lähtekood	<pre>@Test void test() {     Boolean passwordControll;     Boolean ans = true;     String password = "Malle1";     String password2 = "Malle1";     passwordControll = Registration.validationTwoPasswords(password2, password);     assertEquals(ans, passwordControll); }</pre>
Testi raport	 <p>The screenshot shows the JUnit test runner interface. It indicates that the test finished after 0.064 seconds. The summary shows 1/1 runs, 0 errors, and 0 failures. A green progress bar is visible. The test list shows 'testPasswordMatching' and 'testMainPass()' both passing in 0.001 seconds.</p>

#### 1.4.6 TC 6 “Kordu parooli kontroll (Fail)”

Lähtekood	<pre>@Test void testMainFail() {     Boolean passwordControll;     Boolean ans = true;     String password = "Malle1";     String password2 = "Malle";     passwordControll = Registration.validationTwoPasswords(password2, password);     assertEquals(ans, passwordControll); }</pre>
-----------	--

Testi raport	<p>Package Explorer JUnit</p> <p>Finished after 0.079 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 1</p> <p>testPasswordMatching [Runner: JUnit 5] (0.002 s)</p> <p>testMainFail() (0.002 s)</p>
--------------	--

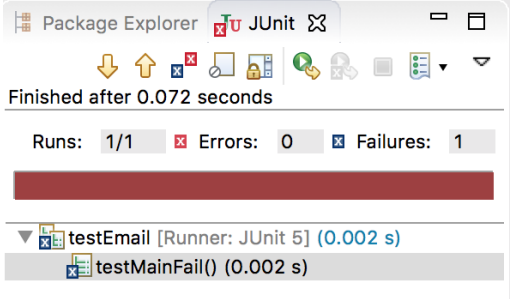
#### 1.4.7 TC 7 “E-maili validatsioon (Pass)”

Lähtekood	<pre>@Test void testMainPass() {     Boolean email;     Boolean ans = true;     String aadress = "test@hot.ee";     email = Registration.validationEmail(aadress);     assertEquals(ans, email); }</pre>
Testi raport	<p>Package Explorer JUnit</p> <p>Finished after 0.067 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 0</p> <p>testEmail [Runner: JUnit 5] (0.000 s)</p> <p>testMainPass() (0.000 s)</p>

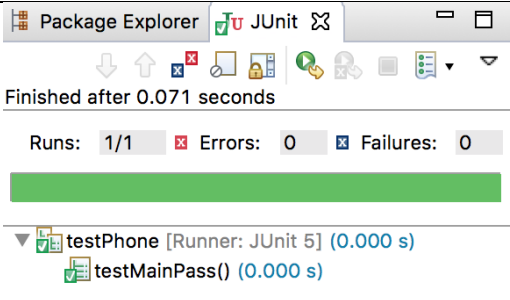
#### 1.4.8 TC 8 “E-maili validatsioon (Fail)”

Lähtekood	<pre>@Test void testMainFail() {     Boolean email;     Boolean ans = true;     String aadress = "test@.ee";     email = Registration.validationEmail(aadress);     assertEquals(ans, email); }</pre>
-----------	---



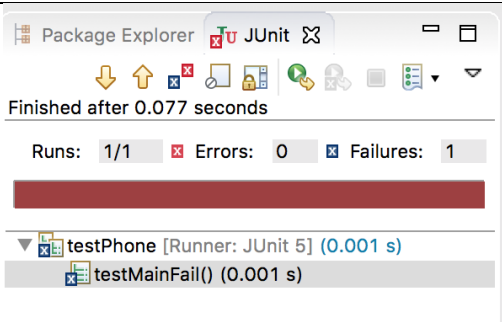
	}
Testi raport	

#### 1.4.9 TC 9 “Telefoni numbri validatsioon (Pass)”

Lähtekood	<pre> @Test void testMainPass() {     Boolean phone;     Boolean ans = true;     String phoneValue = "5500228";     phone = Registration.validationPhone(phoneValue);     assertEquals(ans, phone); } </pre>
Testi raport	

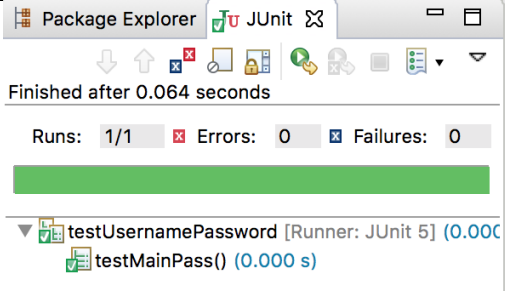
#### 1.4.10 TC 10 “Telefoni numbri validatsioon (Fail)”

Lähtekood	<pre> @Test void testMainFail() {     Boolean phone;     Boolean ans = true;     String phoneValue = "+5500228";     phone = Registration.validationPhone(phoneValue); } </pre>
-----------	---

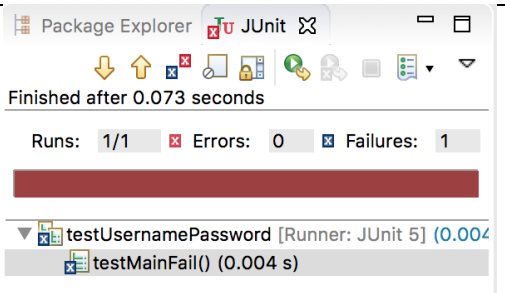
	<pre>assertEquals(ans, phone); }</pre>
Testi raport	

#### 1.4.11 TC 11 "Kas leidub sisestatud Kasutajanimi ja Parool (Pass)"

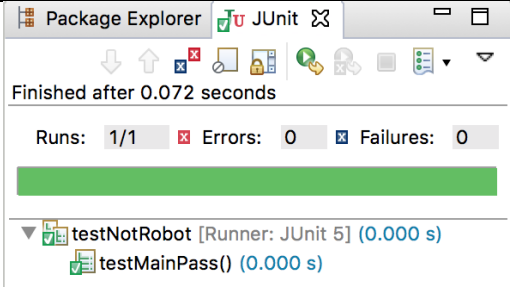
Lähtekood	<pre>@Test void testMainPass() {     Boolean <u>usernamePasswordCorrect</u>;     Boolean ans = true;     String[][] accounts = { {"TestTest", "TestTest1"}, {"TereTere", "TereTere2"}, {"Detsember", "Detsember3"} };     String username = "Detsember";     String password = "Detsember3";      Users check = new Users(username, password);     check.accounts = accounts;     boolean a = check.auth();     assertEquals(ans, a);     check.auth(); }</pre>
-----------	---

Testi raport	 <p>Package Explorer JUnit</p> <p>Finished after 0.064 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 0</p> <p>testUsernamePassword [Runner: JUnit 5] (0.000 s)</p> <p>testMainPass() (0.000 s)</p>
--------------	--

#### 1.4.12 TC 12 "Kas leidub sisestatud Kasutajanimi ja Parool (Fail)"

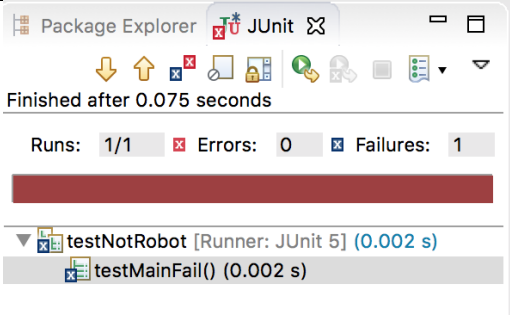
Lähtekood	<pre> @Test void testMainFail() {     Boolean usernamePasswordCorrect;     Boolean ans = true;     String[][] accounts = { {"TestTest", "TestTest1"}, {"TereTere", "TereTere2"}, {"Detsember", "Detsember3"} };     String username = "Detsember";     String password = "Detsember";      Users check = new Users(username, password);     check.accounts = accounts;     boolean a = check.auth();     assertEquals(ans, a);     check.auth(); } </pre>
Testi raport	 <p>Package Explorer JUnit</p> <p>Finished after 0.073 seconds</p> <p>Runs: 1/1 Errors: 0 Failures: 1</p> <p>testUsernamePassword [Runner: JUnit 5] (0.004 s)</p> <p>testMainFail() (0.004 s)</p>

#### 1.4.13 TC 13 "Kas kasutaja ei ole robot (Pass)"

Lähtekood	<pre>@Test void testMainPass() {     Boolean notRobot;     Boolean ans = true;     String notRobotValue = "Jah";     notRobot = Login.controlIfNotRobot(notRobotValue);     assertEquals(ans, notRobot); }</pre>
Testi raport	 <p>The screenshot shows the JUnit test runner interface. At the top, it says 'Finished after 0.072 seconds'. Below that, a summary bar shows 'Runs: 1/1', 'Errors: 0', and 'Failures: 0'. A green progress bar is visible. The test list shows 'testNotRobot [Runner: JUnit 5] (0.000 s)' and 'testMainPass() (0.000 s)' both with green checkmarks indicating they passed.</p>

#### 1.4.14 TC 14 "Kas kasutaja ei ole robot (Fail)"

Lähtekood	<pre>@Test void testMainFail() {     Boolean notRobot;     Boolean ans = true;     String notRobotValue = "Ei";     notRobot = Login.controlIfNotRobot(notRobotValue);     assertEquals(ans, notRobot); }</pre>
-----------	---

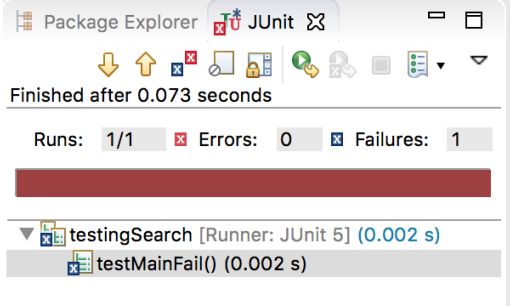
Testi raport	
--------------	---

#### 1.4.15 TC 15 “Sündmuse otsing (Pass)”

Lähtekood	<pre> @Test void testMainPass() {     Boolean userInput;     Boolean ans = true;     String insertedValue = "ED".toUpperCase();     String[] events = { "SIMPLE SESSION 19, SIMPLE SESSION 2019, SAKU SUURHALL", "VALGUSE OOTEL, HEDVIG HANSON JA UKU SUVISTE, PÄRNU KULTUURIMAJA", "STAND-UP COMEDY, COMEDY ESTONIA, PÖÖNING" };      userInput = Search.checkIfEventExists(insertedValue, events);     assertEquals(ans, userInput); } </pre>
Testi raport	

#### 1.4.16 TC 16 “Sündmuse otsing (Fail)”

Lähtekood	<pre> @Test void testMainFail() { </pre>
-----------	--

	<pre> Boolean userInput;  Boolean ans = true;  String insertedValue = " ssss".toUpperCase();  String[] events = { "SIMPLE SESSION 19, SIMPLE SESSION 2019, SAKU SUURHALL", "VALGUSE OOTEL, HEDVIG HANSON JA UKU SUVISTE, PÄRNU KULTUURIMAJA", "STAND-UP COMEDY, COMEDY ESTONIA, PÖÖNING" };  userInput = Search.checkIfEventExists(insertedValue, events); assertEquals(ans, userInput); } </pre>
Testi raport	

## 2 Koodi katvus

Koodi katvuse analüüsiks on projekteerimeeskond valinud programmi „Matemaatilised arvutused“ ehk kalkulaatori, mille lähtekoodi on kirjutanud projekteerimeeskonna liige Berit Põldoja ise. Projekti meeskond valis antud programmi seetõttu, et Piletilevi lähtekood pole saadaval avalikkusele. Kood on kirjutatud *Python*’is, kasutades tekstiredaktorit Sublime Text 3 ja käivitavat *Command Prompt*’i abil.

### 2.1 Programm „Matemaatilised arvutused“ nõuded lühidalt

Programm peab suutma arvutada põhilisi matemaatilisi tehteid. Kõige põhilisemad tehted, mida ta läbima peab on liitmine, lahutamine, korrutamine, jagamine, *Fibonacci* arvujada leidmine, astendamine ja faktoriaali leidmine. Lisaks peaks programm oskama leida, kuidas astendamine toimub ehk kuvama massiivis astendamise ajutised väärtused (näide:  $4^4 = [1, 4, 16, 64, 256]$ ). Samuti on programmis vajalik arvutada *The Lazy Caterer’s* jada, mis näitab maksimaalse arvu tükke (piiratud / piiranguteta), mida võidakse ringi kujulisest söögist (näide: pitsa, pannkook) lõigata, mida on võimalik teha ette antud numbri sirglõigete põhjal.

### 2.2 Programmi kood

Lähtekood on alla laetav versioonihaldusest: <https://github.com/bpoldoja/ITproject>.

Täpsem informatsioon (lähtekood, testid) ja juhend testide käivitamiseks ja projekti ülespanekuks on kirjeldatud peatükis 3.2. *Tarkvara teise osa testimiseks*.

### 2.3 Koodi katvuse testid

Koodi katvus on testitud *Python Coverage tool*’i abil. Kood, mida testiti asub failis *mathematics.py* ja läbis katvuse 69% ulatuses. Testid asuvad failis *test\_math.py* ning läbis koodi katvuse 100% ulatuses.

Kuna tegemist on Beriti poolt loodud programmiga, mis pole Piletilevi süsteemiga seotud, siis pole võimalik hinnata programmi kvaliteeti ja testide adekvaatsust.

## Coverage report: 79%

Module ↓	statements	missing	excluded	coverage
mathematics.py	119	37	0	69%
test_math.py	60	0	0	100%
<b>Total</b>	<b>179</b>	<b>37</b>	<b>0</b>	<b>79%</b>

coverage.py v4.5.1, created at 2018-11-01 18:20

Joonis 1: Koodi katvuse raport

Programmikoodi osa, mida ei ole võimalik testidega katta on välja toodud peatükis 2.3.12 Testimata programmiosad.

### 2.3.1 T1 „Liitmine“

Test kontrollib matemaatilist tehet – liitmine. Programm suudab mistahes negatiivsete ja positiivsete arvude puhul sooritada tehte positiivselt. Katab koodi osa:

```
3 def add(x, y):  
4     return x + y  
5
```

### 2.3.2 T2 „Lahutamine“

Test kontrollib matemaatilist tehet – lahutamine. Programm suudab mistahes negatiivsete ja positiivsete arvude puhul sooritada lahutamistehte edukalt. Katab koodi osa:

```
7 def subtract(x, y):  
8     return x - y  
9
```

### 2.3.3 T3 „Korrutamine“

Test kontrollib matemaatilist tehet – korrutamine. Programm suudab mistahes negatiivsete ja positiivsete arvude puhul sooritada korrutamistehte edukalt. Katab koodi osa:



```

11 ▼ def multiply(x, y):
12     a = 0
13     result = 0
14     pos = True
15     if x == 1:
16         return y
17     elif y == 1:
18         return x
19 ▼     elif x > 0 and y < 0:
20         pos = False
21         y = y * (-1)
22
23 ▼     elif x < 0 and y > 0:
24         pos = False
25         y = y * (-1)
26
27 ▼     elif x < 0 and y < 0:
28         y = y * (-1)
29         x = x * (-1)
30
31     elif x == 0 or y == 0:
32         return 0
33
34
35 ▼     while a < y:
36         result = result + x
37         a += 1
38
39     if not pos:
40         result = result * (-1)
41
42     return result

```

### 2.3.4 T4 „Jagamine“

Test kontrollib matemaatilist tehet – jagamine. Programm suudab mistahes negatiivsete ja positiivsete arvude puhul sooritada jagamistehte edukalt. Katab koodi osa:

```

45 def divide(x, y):
46     try: return x / y
47     except ZeroDivisionError:
48         return None
49

```

### 2.3.5 T5 „Räsimine“

Test kontrollib ühesuunalise tekstistringide kodeerimist. Programm suudab mistahes tekstistringide puhul muuta nad räsiks. Katab koodi osa:

```

50 ▼ def hashing(x):
51     """
52     Input hashing dependant on the length of the input
53
54     """
55     x = x.encode(encoding='UTF-8')
56 ▼     if len(x) < 5:
57         hash_result = hashlib.sha1(x)
58         #print('sha1')
59 ▼     elif len(x) == 5:
60         hash_result = hashlib.md5(x)
61         #print('md5')
62 ▼     elif len(x) <= 25:
63         hash_result = hashlib.sha256(x)
64         #print('sha256')
65 ▼     elif len(x) == 38:
66         hash_result = hashlib.sha384(x)
67         #print('sha384')
68 ▼     else:
69         hash_result = hashlib.sha512(x)
70         #print('sha512')
71
72     return hash_result.hexdigest()
73

```

### 2.3.6 T6 „Fibonacci arvujada“

Test kontrollib matemaatilist tehet – Fibonacci arvu. Programm suudab mistahes positiivsete arvude leida Fibonacci arvujada edukalt. Katab koodi osa:

```

75 def fib(n):
76     """
77     Fibonacci sequence calculator
78
79     """
80     if n < 0:
81         return None
82     elif n == 0:
83         return 0
84     elif n == 1:
85         return 1
86     else:
87         return fib(n-1)+fib(n-2)
88

```

### 2.3.7 T7 „The Lazy Caterer’s jada“

Test kontrollib matemaatilist tehet – *The Lazy Caterer’s* jada. Programm suudab leida maksimaalse numbri tükke, mistahes positiivse arvu ette antes. Katab koodi osa:

```

89 ▼ def pie(n):
90     """
91     The Lazy Caterer's Sequence - describes the maximum number of pieces (or bounded/unbounded regions)
92     of a circle (a pancake or pizza is usually used to describe the situation) that can be made with a
93     given number of straight cuts.
94
95     """
96
97     if n < 0:
98         return None
99
100     piece_count = (n*n+n+2) / 2
101
102     return piece_count

```

### 2.3.8 T8 „Astendamine“

Test kontrollib matemaatilist tehet – astendamine. Programm suudab mistahes positiivsete arvude puhul sooritada astendamise edukalt. Katab koodi osa:

```

105 def powerof(base, maxi):
106     """
107     Calculates the power of numbers
108
109     """
110     counter = 1
111     res = base
112     if maxi == 0:
113         return 1
114
115     while counter < maxi:
116         counter = counter + 1
117         res = multiply(res, base)
118
119

```

### 2.3.9 T9 „Astendamise väärtused listis“

Test kontrollib matemaatilist tehet – astendamine massiivis. Programm suudab mistahes positiivsete arvude puhul kuvada massiivis astendamise ajutised väärtused edukalt. Katab koodi osa:

```

122 def arrayof(base, maxi):
123     """
124     Shows all the temporary values of taking to the power of given value
125
126     """
127
128     counter = 0
129     array = []
130
131     while counter <= maxi:
132         array.append(powerof(base, counter))
133         counter = counter + 1
134
135     return array

```

### 2.3.10 T10 „Faktoriaal“

Test kontrollib matemaatilist tehet – faktoriaali arvutamine. Programm suudab mistahes positiivsete arvude puhul võtta numbraid faktoriaali edukalt. Katab koodi osa:

```

137 ▼ def factorial(n):
138     """
139     Calculates the factorial of a number
140
141     """
142     num = 1
143
144 ▼     while n >= 1:
145         num = num * n
146         n = n - 1
147
148     return num

```

### 2.3.11 Testimata programmiosad

Testimata programmi osad, mida koodi katvus coverage.py ei saanud katta failis mathematics.py olid kaetud manuaalselt. Koodi katvuse raportis (joonis.1) - 69%.

Nendeks osadeks olid programmiga suhtlemine ehk arvude sisestamine:

```

152 # Take input from the user
153 choice = input("Enter choice(1/2/3/4/5/6/7/8/9/10):")
154 ▼ if choice == '1':
155     num1 = int(input("Enter first number: "))
156     num2 = int(input("Enter second number: "))
157     print(num1,"+",num2,"=", add(num1,num2))
158 ▼ elif choice == '2':
159     num1 = int(input("Enter first number: "))
160     num2 = int(input("Enter second number: "))
161     print(num1,"-",num2,"=", subtract(num1,num2))
162 ▼ elif choice == '3':
163     num1 = int(input("Enter first number: "))
164     num2 = int(input("Enter second number: "))
165     print(num1,"*",num2,"=", multiply(num1,num2))
166 ▼ elif choice == '4':
167     num1 = int(input("Enter first number: "))
168     num2 = int(input("Enter second number: "))
169     print(num1,"/",num2,"=", divide(num1,num2))
170 ▼ elif choice == '5':
171     txt = input("Enter text: ")
172     print("Text:",txt, hashing(txt))
173 ▼ elif choice == '6':
174     num3 = int(input("Enter number: "))
175     print("Fibonacci:",num3,"=", fib(num3))
176 ▼ elif choice == '7':
177     num3 = int(input("Enter number: "))
178     print("Pie:",num3,"is", pie(num3))
179 ▼ elif choice == '8':
180     num1 = int(input("Enter first number: "))
181     num2 = int(input("Enter second number: "))
182     print("Power of",num1,"^", num2,"=",powerof(num1,num2))
183 ▼ elif choice == '9':
184     num1 = int(input("Enter first number: "))
185     num2 = int(input("Enter second number: "))
186     print("Array of",num1,"^", num2,"=", arrayof(num1,num2))
187 ▼ elif choice == '10':
188     num3 = int(input("Enter number: "))
189     print("Factorial",num3,"is", factorial(num3))
190 ▼ else:
191     print("Invalid input")

```

## 2.4 Testid ja testi tulemite raportid

### 2.4.1 T1 „Liitmine“

Lähtekood	<pre>def test_add(self):     self.assertEqual(mathematics.add(10,5), 15)     self.assertEqual(mathematics.add(-1,1), 0)     self.assertEqual(mathematics.add(3.31,5), 8.31)</pre>
Testi raport	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

### 2.4.2 T2 „Lahutamine“

Lähtekood	<pre>def test_subtract(self):     self.assertEqual(mathematics.subtract(10,5), 5)     self.assertEqual(mathematics.subtract(-1,-1), 0)     self.assertEqual(mathematics.subtract(3.4,5), -1.6)</pre>
Testi raport	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

### 2.4.3 T3 „Korrutamine“

Lähtekood	<pre>def test_multiply(self):     self.assertEqual(mathematics.multiply(10,5), 50)     self.assertEqual(mathematics.multiply(-1,1), -1)     self.assertEqual(mathematics.multiply(3.5,5), 17.5)     self.assertEqual(mathematics.multiply(0,0), 0)</pre>
-----------	--

	<pre>self.assertEqual(mathematics.multiply(0,1), 0) self.assertEqual(mathematics.multiply(1,0), 0)</pre>
<b>Testi raport</b>	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s OK</pre>

#### 2.4.4 T4 „Jagamine“

<b>Lähtekood</b>	<pre>def test_divide(self):     self.assertEqual(mathematics.divide(10,5), 2)     self.assertEqual(mathematics.divide(-3,1), -3)     self.assertEqual(mathematics.divide(45,0), None)     self.assertEqual(mathematics.divide(0,0), None)     self.assertEqual(mathematics.divide(0,1), 0)     self.assertEqual(mathematics.divide(4,3.6), 1.1111111111111112)</pre>
<b>Testi raport</b>	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s OK</pre>

#### 2.4.5 T5 „Räsimine“

<b>Lähtekood</b>	<pre>def test_hashing(self):     self.assertEqual(mathematics.hashing('Lore'), '730949e23ca46f310466fbf205ffb165aef1fd7b')     self.assertEqual(mathematics.hashing('Lorem'), 'db6ff2ffe2df7b8cfc0d9542bdce27dc')</pre>
------------------	---

	<pre> self.assertEqual(mathematics.hashing('Lorem ipsum dolor sit ame'), '92c40e81b7724f813e95dfab7ee61108474eed86ab785bd62bb6f873ffa11d98' )  self.assertEqual(mathematics.hashing('Lorem ipsum dolor sit amet, ex ius ass'), 'd668f938bf678b6a5ce0f355ac1b8e11af70e9b44b8cefe01a383a38880f108e1 0f78965c3a2ec526987b19771c62db4')  self.assertEqual(mathematics.hashing('Lorem ipsum dolor sit amet, ex ius assum nonumes. Iudico conceptam consequuntur pri ei, mei no vero velit, et prompta fabulas rationibus vel. Ex vim magna'), 'b00f28ffa5ea37497072474c7d66cd891253070b05cf1c030cb56168614e03c8 cfa80f728b002b2a6fb05099a47d84e9005566b184f3d43f999d501216666d79' ) </pre>
Testi rapo rt	<pre> C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK </pre>

#### 2.4.6 T6 „Fibonacci arvujada“

Lähtekood	<pre> def test_fib(self):      self.assertEqual(mathematics.fib(0), 0)      self.assertEqual(mathematics.fib(1), 1)      self.assertEqual(mathematics.fib(10), 55)      self.assertEqual(mathematics.fib(15), 610)      self.assertEqual(mathematics.fib(-10), None) </pre>
Testi raport	<pre> C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK </pre>

#### 2.4.7 T7 „The Lazy Caterer’s jada“

Lähtekood	<pre>def test_pie(self):     self.assertEqual(mathematics.pie(0), 1)     self.assertEqual(mathematics.pie(1), 2)     self.assertEqual(mathematics.pie(2), 4)     self.assertEqual(mathematics.pie(10), 56)     self.assertEqual(mathematics.pie(-112313210), None)</pre>
Testi raport	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

#### 2.4.8 T8 „Astendamine“

Lähtekood	<pre>def test_powerof(self):     self.assertEqual(mathematics.powerof(4, 2), 16)     self.assertEqual(mathematics.powerof(7, 1), 7)     self.assertEqual(mathematics.powerof(8, 0), 1)     self.assertEqual(mathematics.powerof(0, 2), 0)</pre>
Testi raport	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

#### 2.4.9 T9 „Astendamise väärtused listis“

Lähtekood	<pre>def test_arrayof(self):     self.assertEqual(mathematics.arrayof(2, 3), [1, 2, 4, 8])     self.assertEqual(mathematics.arrayof(4, 4), [1, 4, 16, 64, 256])</pre>
-----------	---



	<pre>self.assertEqual(mathematics.arrayof(8, 0), [1]) self.assertEqual(mathematics.arrayof(0, 2), [1, 0, 0])</pre>
<b>Testi raport</b>	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

#### 2.4.10 T10 „Faktoriaal“

<b>Lähtekood</b>	<pre>def test_factorial(self):     self.assertEqual(mathematics.factorial(4), 24)     self.assertEqual(mathematics.factorial(7), 5040)     self.assertEqual(mathematics.factorial(8), 40320)     self.assertEqual(mathematics.factorial(0), 1)</pre>
<b>Testi raport</b>	<pre>C:\Users\Berit\Downloads&gt;python test_math.py . ----- Ran 1 test in 0.000s  OK</pre>

### 3 Eeldused tarkvara testimiseks

26.10.2018 toimunud praktikumi tunnis tõi juhendaja Rain Neemlaid välja, et antud peatükis tuleb välja tuua teave Lab 2 programmikoodide ja testide kohta, kui Lab 1 kirjeldatu ei ole avalikkusele kättesaadav (lähtekood) ja Lab 2 töö ei ole eelneva töö jätk. Samuti tõi juhendaja välja, et peatükis 3.3 *Piletilevi arhitektuur* tuleb esitada Lab 1 arhitektuur, sest enamikel tudengitel puudus see eelmises töös. Lisaks sellele on antud peatükis kirjas ka juhendid, kuidas retsenseerija või õppejõud saab antud teste käivitada.

Projektimeeskond lähtus antud peatüki kirjeldamisel praktikumis õppejõu Rain Neemlaid väljatoodule.

#### 3.1 Tarkvara esimese osa testimiseks

- Tarkvara nimi: Piletilevi
- Versioon: 1.0
- Autor: Triinu Tammer
- Veebileht: [https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps\\_tdyL8Tu1XxfMPhNy\\_LAioLuPpT3Kaa6E0WgDh\\_s\\_mBd2Y](https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps_tdyL8Tu1XxfMPhNy_LAioLuPpT3Kaa6E0WgDh_s_mBd2Y)
- Litsents: puudub

##### 3.1.1 Eeldused esimese osa testimiseks

Eeldus, et saaks esimese osa koodi testida, peab olema olemas Java ning selle IDE. Oma töös kasutame Java SE JDK ja Eclipse'i. Kuna Piletilevil pole kood kättesaadav, testisime me esimeses osas sarnaseid funktsionaalsusi, mis on Piletilevil olemas. Järgnevalt on välja toodud sammude haaval vajaminevad tegevused:

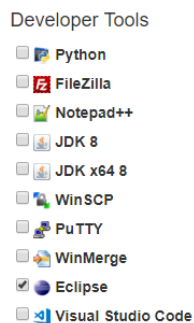
1. Mine lehele

<https://www.oracle.com/technetwork/java/javase/downloads/index.html> -> lae

alla oma OS-ile sobilik variant alljärgnevat:

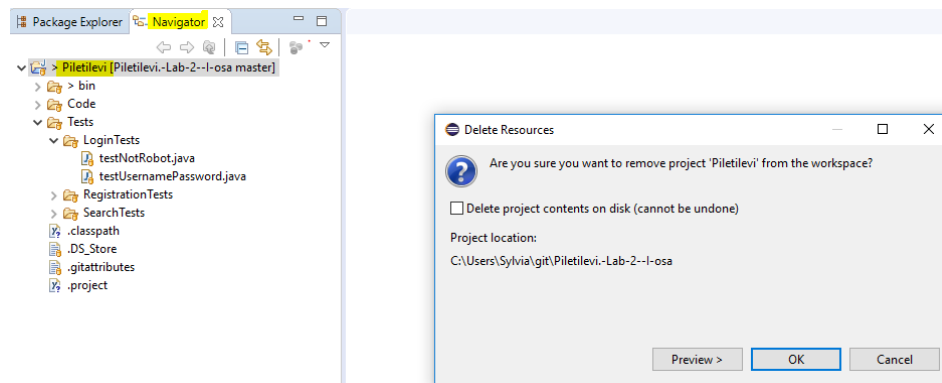
Java SE Development Kit 11.0.1		
You must accept the <a href="#">Oracle Technology Network License Agreement for Oracle Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux	147.4 MB	<a href="#">jdk-11.0.1_linux-x64_bin.deb</a>
Linux	154.09 MB	<a href="#">jdk-11.0.1_linux-x64_bin.rpm</a>
Linux	171.43 MB	<a href="#">jdk-11.0.1_linux-x64_bin.tar.gz</a>
macOS	166.2 MB	<a href="#">jdk-11.0.1_osx-x64_bin.dmg</a>
macOS	166.55 MB	<a href="#">jdk-11.0.1_osx-x64_bin.tar.gz</a>
Solaris SPARC	186.8 MB	<a href="#">jdk-11.0.1_solaris-sparcv9_bin.tar.gz</a>
Windows	150.98 MB	<a href="#">jdk-11.0.1_windows-x64_bin.exe</a>
Windows	170.99 MB	<a href="#">jdk-11.0.1_windows-x64_bin.zip</a>

2. Mine lehele <https://ninite.com/> -> tee linnuke Eclipse ette -> Get Your Ninite:

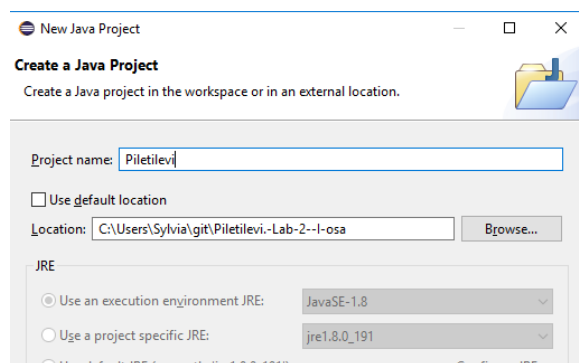


3. Ava Eclipse -> vali asukoht, kuhu failid salvestatake -> Launch -> Help -> Install New Software -> Work with: <http://download.eclipse.org/egit/updates> -> vajuta Enter klahvi -> lisa kõigile linnukesed -> Next -> Restart Now
4. (Eclipse) Window -> Show View -> Other... -> Git -> Git Repositories
5. (Eclipse) File -> Import -> Git -> Projects from Git -> Next -> Clone URI
6. Kopeeri leheküljelt [https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps\\_tdyL8Tu1XxfMPhNy\\_LAioLuPpT3Kaa6E0WgDh\\_s\\_mBd2Y](https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa?fbclid=IwAR1O6zoZKgWps_tdyL8Tu1XxfMPhNy_LAioLuPpT3Kaa6E0WgDh_s_mBd2Y) vajutades **Clone or download** -> <https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa.git>
7. (Eclipse) URI: <https://github.com/Triinukke/Piletilevi.-Lab-2--I-osa.git> -> Next -> **Import as general Project** -> Next

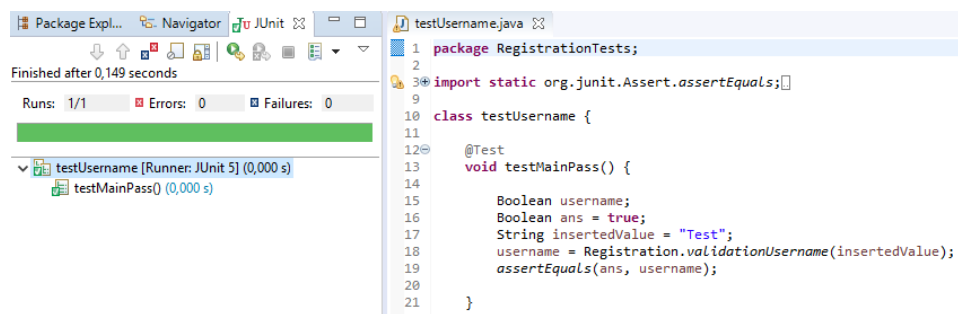
8. (Eclipse) Kustuta projekt ära, sest hetkel ei ole võimalik käivitada, et testima hakata -> kopeeri projekti aadress:



9. (Eclipse) **File -> New Java Project** -> sisesta projekti aadress ning nimi:



10. Testi JUnitTest-iga:



### 3.2 Tarkvara teise osa testimiseks

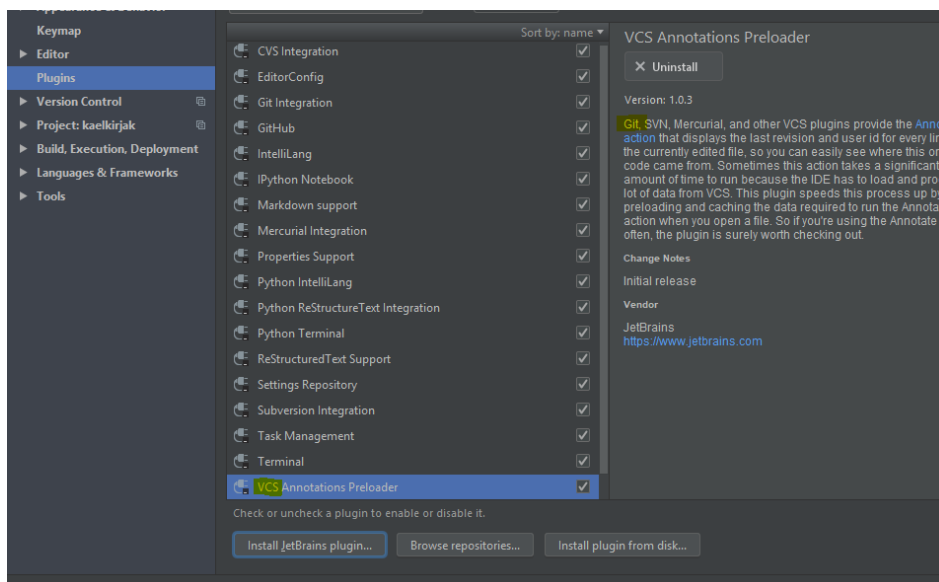
- Tarkvara nimi: Matemaatilised arvutused
- Version: 1.0
- Autor: Berit Põldoja

- Veebileht: <https://github.com/bpoldoja/ITproject?fbclid=IwAR1-W-EHlgWiy4VHakglG251zRv7NNXsahFbUiLDqxBtNZ4jMA57yIsIJp4>
- Litsents: puudub

### 3.2.1 Eeldused teise osa testimiseks

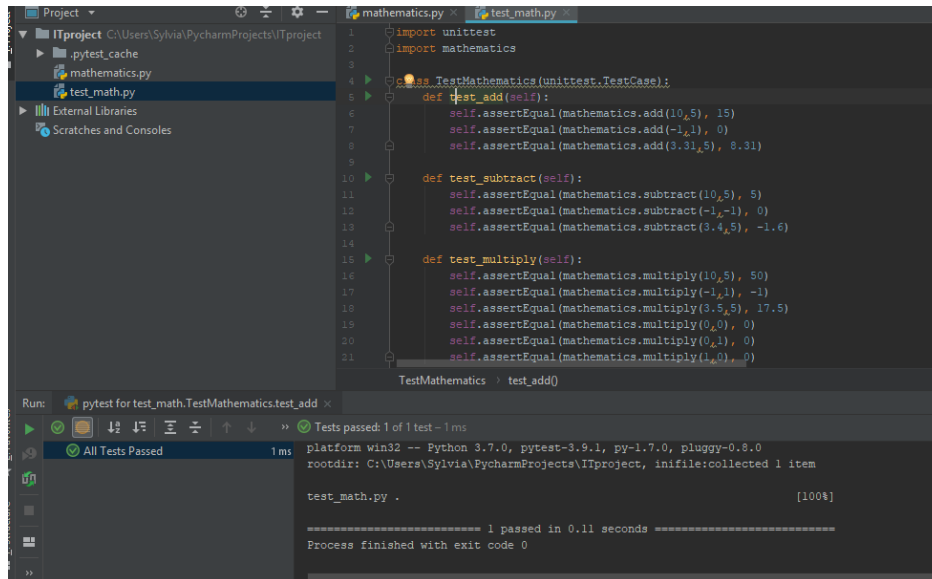
Eeldus, et saaks teise osa koodi testida, peab olema olemas Python ning selle IDE. Oma töös kasutame Python 3 ja PyCharm'i. Kuna Piletilevil pole kood kättesaadav, testisime me teises osas matemaatilisi arvutusi. Järgnevalt on välja toodud sammude haaval vajaminevad tegevused:

1. Mine lehele <https://www.python.org/downloads/> -> kliki **Download Python 3.7.1** -> ava .exe fail -> avaneb Python 3.7.1 Setup -> **Install Now** -> Setup was successful
2. Mine lehele <https://www.jetbrains.com/pycharm/> -> kliki **Download now** -> Community **Download** -> ava fail
3. Ava Windows otsing ning kirjuta „**PyCharm**“ -> kliki aplikatsiooni peale
4. (PyCharm) **File** -> **Settings** -> **Plugins** -> **Install JetBrains plugin...** -> **git** (VCS Annotations Preloader) -> **OK**



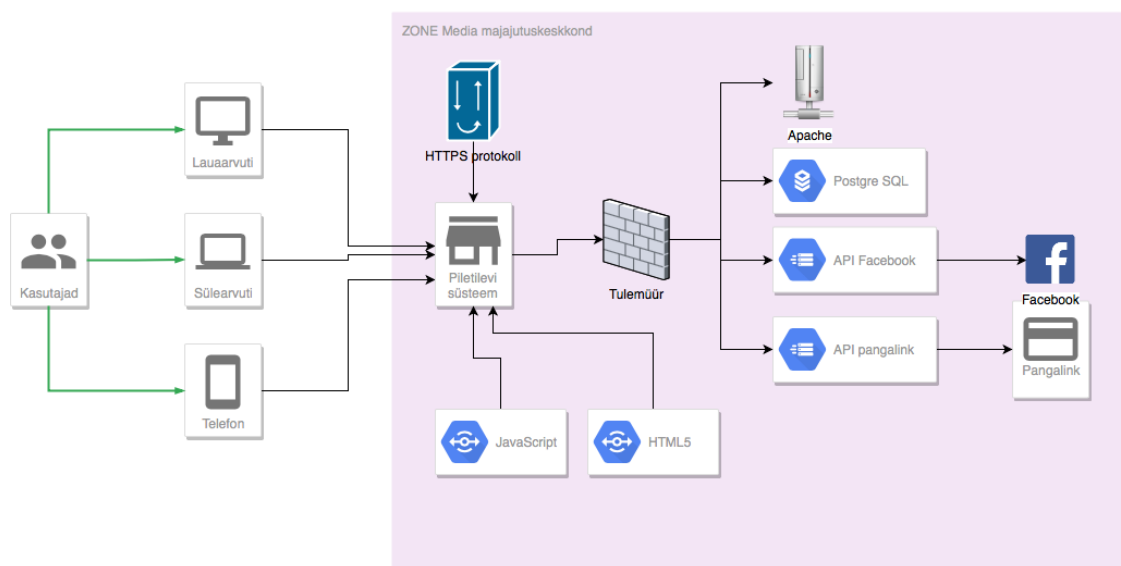
5. Lae alla git (Windowsi puhul <https://git-scm.com/download/win>) -> käivita .exe fail
6. (PyCharm) **Settings** -> **Version Control** -> **Git** (kontrolli, kas on leidnud git.exe üles)

7. (PyCharm) **Settings** -> **Version Control** -> **GitHub** -> logi GitHubi sisse
8. Kopeeri leheküljelt <https://github.com/bpoldoja/ITproject?fbclid=IwAR1-W-EHlgWiy4VHakglG251zRv7NNXsahFbUiLDqxBtNZ4jMA57yIsIjp4> vajutades **Clone or download** -> <https://github.com/bpoldoja/ITproject.git>
9. (PyCharm) **VCS** -> **Checkout from Version Control** -> **Git** (sisesta <https://github.com/bpoldoja/ITproject.git>) -> **Clone**
10. Testi PyCharm Unittest-iga:



### 3.3 Piletilevi arhitektuur

Lähtudes 26.10.2018 praktikumi tunnis juhendaja Rain Neemlaid väljatoodule, esitleb projektimeeskond antud töös sama arhitektuuri pilti, mida kajastati Lab 1 töös.



Joonis 2: Piletilevi.ee süsteemi üldise arhitektuuri mudel

## **Projekti liikmete osalus Lab2**

Lab 2 grupitöös osalesid ja panustasid töö autorid (Triinu Tammer, Sylvia Krupp, Berit Põldoja). Kuna töös on 3 punkti, jaotati need vastavalt töö autorite vahel. Esimese osa (ühiktestimine) kirjeldas Triinu Tammer, teise osa (koodi katvus) Berit Põldoja ning kolmanda osa (tarkvara testimise juhend) Sylvia Krupp. Lisaks oma osa kirjutamisele vaatasid meeskonnaliikmed teiste meeskonnaliikmete osa üle.