

# Algoritmid ja andmestruktuurid

Arvutipraktikum 2017/2018 sügissemelstril

J.Liivi 2-202, E12-14

J.Liivi 2-202, T14-16

[kodu.ut.ee/~kiho/ads/fall17](http://kodu.ut.ee/~kiho/ads/fall17)

Jüri Kiho

[jyri.kiho@ut.ee](mailto:kyri.kiho@ut.ee)

# Eesmärgid

Paremini omandada, kinnistada aine *LTAT.03.005 Algoritmid ja andmestruktuurid* õpiväljundeid – vastavate mõistete kasutamisel praktiliste programmeerimisülesannete lahendamise käigus. Täiendada programmeerimisoskust.

## Üldnõuded

**Osavõtt** praktikumidest tarvilik.

**Praktikumitööd** tuleb reeglina teha kohapeal. Vajadusel võib viimistleda praktikumiväliselt.

Praktikumi alguses on väike (~5 min) **test**. Peamiselt eelneva praktikumi/loengu ainetel.

**Kodutöö (iseseisva töö)** ülesanne antakse praktikumis ja lahendus kontrollitakse (lõplikult) samuti praktikumis. Kodutööks on aega 12 päeva. Soovitav on kodutöö esitada ennetähtaegselt – siis on võimalik saada tagasisidet ning sellele vastavalt kodutööd veel viimistleda. Kui eelneva(te)st teemakohas(t)est praktikumi(de)st on puudunud, siis tuleb koos kodutööga esitada ka mõningad vastava(te)s praktikumi(de)s esitletud harjutusülesannete lahendused (nõuetekohaselt vormistatud programmid). **Paarisprogrammeerimine** lubatud (selgelt peab olema näidatud kummagi autori panus; hinne - mõlemal sama).

Kodutööde lahendused tuleb laadida Moodle'isse, autonoomselt **käsurealt kompileeritavate ja käivitatavate programmidenä**. Sõltumata sellest, millises arenduskeskkonnas programm on loodud, teha esitatava programmi lõppviimistlus toimetiga *Notepad++*; kontrollida, et kodeering oleks ANSI ja kompileerida/käivitada saaks käsurealt.

Alates 14. õppenädalast saab taotleda **kuni kahte lisaülesannet** teemadest 1-6, kuid ainult nendel teemadel, mille kodutöö on esitamata või mitteamestatud. **Lisaülesande esitamise tähtaeg** antakse koos lisaülesandega; hiljemalt on see kaks päeva enne aine esimest korralist eksamit.

**Hilinenult esitatud kodutööd vastu ei võeta.**

# Kodutööde vormistusnõudeid

Praktikumides ja koduülesannetes on nõutud kasutada programmeerimiskeelt *Java*. Praktikumimaterjalid on orienteeritud versioonile *Java8*.

Lubatud on **ainult *Java API* standardpaketid**. Erandina võib lisaks kasutada vaid paketti `ee.ut.kiho.aa.*`.

Esitatavad programmid peavad olema mõistlikult **kommenteeritud** ning varustatud ka **päise** ehk **preambuliga**, milles on toodud autor(id), ülesanne, liik (praktikumitöö, iseseisev töö) jmt.

Preambuli näide:

```
/* Algoritmid ja andmestruktuurid
 * 2017/2018 sügissemester
 * Praktikumitöö (KKP_5)
 * Koodipuu Huffmani algoritmiga
 * @author Ülli Õpilane
 */
import ee.ut.kiho.aa.graaf.*;
    ...
```

Peaprogrammi alguses antakse kommentaarina juhend programmi kompileerimiseks ja käitamiseks käsurealt.

**Eesti keele (emakeele) kasutamine on prioriteet** (kommentaarid ning muutujate, meetodite, parameetrite nimed jmt).

Iga meetod tuleb varustada **välise spetsifikatsiooniga**: mis on **antud** (sisendina) ja milline on oodatav **tulemus** (väljund).

Programmi osad, mis on mujalt saadud (või mille eeskuju on mujalt võetud) peavad olema varustatud **viitega allikale**.

Programmi töö **tulemused** väljastatakse koos **selgitava tekstiga**

# Kodutööde hindamine

Kodutöid, mis on esitatud õigeaegselt, hinnatakse skaalas 1-100:

Hinne 100 (suurepärase) – ilma selgete puudusteta suurepärase töö, sh üle keskmise keerukuse, lahenduse selguse, loova lähenemise vm osas.

Hinne 80 – 99 (sõltuvalt pisivigadest) – korrektne lahendus, vormistus. Pisivead on aktsepteeritavad.

Hinne 60 – 79 (sõltuvalt puudustest) – mõnede selgete puudustega töö.

Hinne 40 – 59 (sõltuvalt märkimisväärsetest puudustest) – suurte, märkimisväärsete või suure hulga puudustega töö.

---

Hinne 20 – 39 (sõltuvalt suurtest puudustest ja vigadest) – väga suurte puuduste ja vigadega töö.

Hinne 1 - 15 – ei vasta nõuetele, või plagiaat

Kodutöö hindega üle 39 on **arvestatud**. Selle hinne läheb arvesseksamipunktides.

Kodutöö hindega alla 40 on **mittearvestatud**. Mittearvestatud või **esitamata** kodutöö annab eksamihindesse 0 punkti.

Arvutipraktikumi juhendaja võib mõnele üliõpilasele anda lisaks 1 kuni 5 **boonuspunkti**, eriti silmapaistvate töötulemuste eest praktikumis.

# Eksamihinde kujunemine

Eksamihinne määratakseksamipunktide (0 .. 100) järgi. Eksamipunktidesse panustavad (liidetavatena):

- kahe kontrolltöö tulemused (kumbki kuni 30 punkti)
- arvutipraktikumi eest saadud punktid – arvestatud kodutööde punktisumma ja boonuspunktid.

**Kodutööde korral on nõutud, et töödest 5. ja 6. peab vähemalt üks olema arvestatud ning töödest 7. ja 8. peab vähemalt üks olema arvestatud. Vastasel korral kodutööde punktisummat vähendatakse 2 korda.** Täpsemalt kirjeldabksamihinde kujunemist alljärgnev *Java*-meetodina esitatud algoritm.

```
static char eksamihinne(int kntr1, int kntr2, int[] k){
// Antud: kntr1, kntr2 -- kahe kontrolltöö punktid, 0 <= kntr1, kntr2 <= 30;
//      k[1] .. k[8] -- kaheksa kodutöö punktid,
//      k[i] = 0, kui i-s kodutöö on arvestamata või esitamata,
//      vastasel korral k[i] = kodutöö hinne, 40 <= k[i] <= 100;
//      k[0] - boonuspunktid, 0 <= k[0] <= 5
// Tulem: tagastatakse eksami hinne, üks tähtedest A, B, C, D, E, F
//      kodutööde punktisumma (0 <= sumK <= 800):
int sumK = 0; for(int i = 1; i <= 8; i++) sumK += k[i];
// trahv (1 või 0):
int trahv = (k[5] == 0 && k[6] == 0 || k[7] == 0 && k[8] == 0)? 1 : 0;
// punktid arvutipraktikumi kodutööde eest:
double apD = sumK/20.0/(1 + trahv); // 0.0 <= apD <= 40.0
int ap = (int)Math.round(apD); // ümardatult 0 <= ap <= 40
// eksamipunktid, boonuspunktide (k[0]) tõttu võib olla > 100:
int ep = kntr1 + kntr2 + ap + k[0];
// eksamihinne:
return (ep > 100)? 'A' : "FFFFFFEDCBA".charAt((ep-1)/10); // 91-100:A, 81-90:B jne
}
```

# Arvutipraktikumi kava

Praktikumis lahendatakse teemakohaseid harjutusülesandeid, ettevalmistusena sellel teemal antavale kodutööle. Antakse teemale vastava kodutöö ülesanne (või lisaülesanne 14.-15. nädalal), vajadusel täpsustades esitamise tähtaja, ning näpunäiteid selle sooritamiseks.

Alates teisest teemast kontrollitakse (pistelisel) eelmise, esitatud kodutöö autori asjatundlikkust.

Demonstreeritakse eelmise teema kodutöö näidislahendust, kui selle kodutöö esitamise tähtaeg on juba möödas.

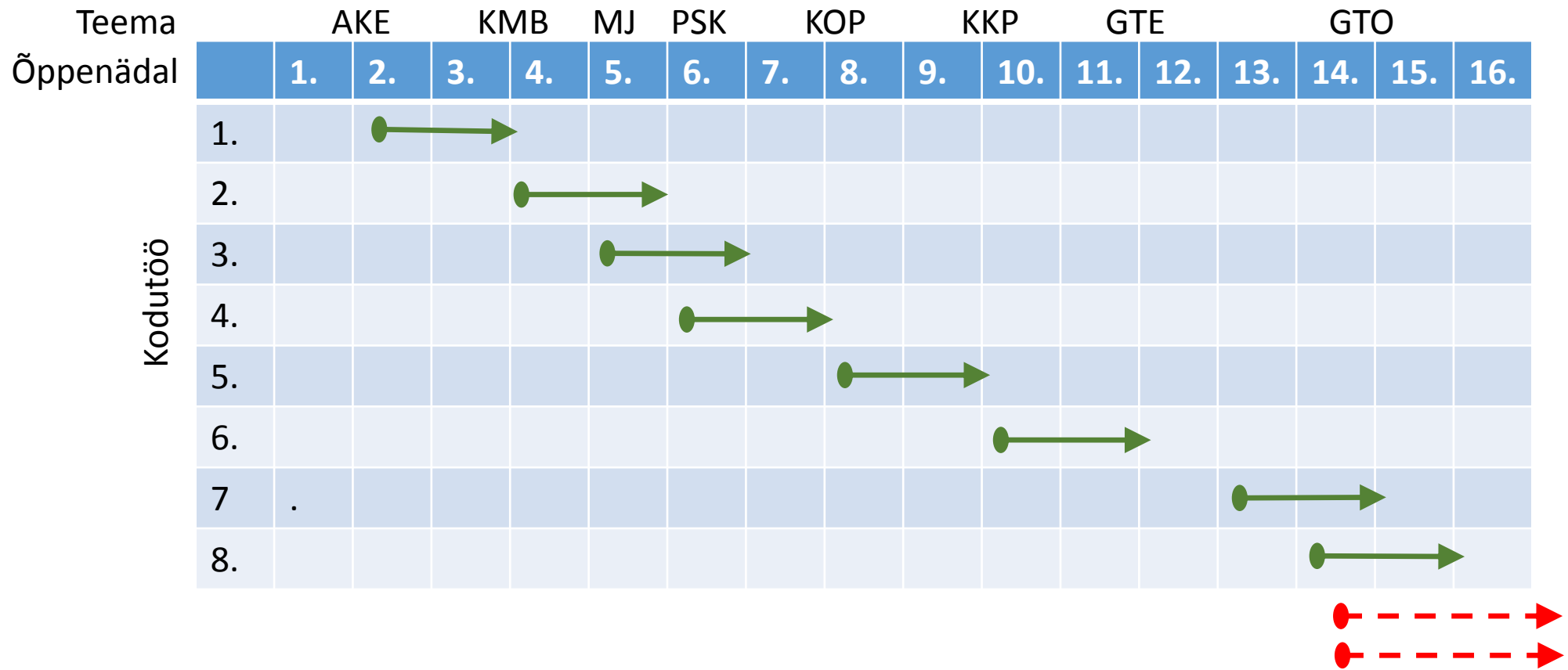
Arvutipraktikumi kogumaht on 32 akadeemilist tundi. Ühele teemale (kaheksast) pühendatakse keskmiselt 4 tundi, st kaks praktikumi.

Arvutipraktikumi täpse ajakava oma rühmale koostab praktikumi juhendaja.

# Praktikumide ja kodutööde teemad ning ajakava

## TEEMAD

1. AKE. Ajalise keerukuse empiiriline hindamine.
2. KMB. Kombinatorika. Rekursioon.
3. MJ. Magasin ja järjekord.
4. PSK. Loendamine. Paisksalvestus.
5. KOP. Kahend(otsimis)puud.
6. KKP. Kahendkuhi. Pakkimine.
7. GTE. Teed graafis.
8. GTO. Graafi toes.





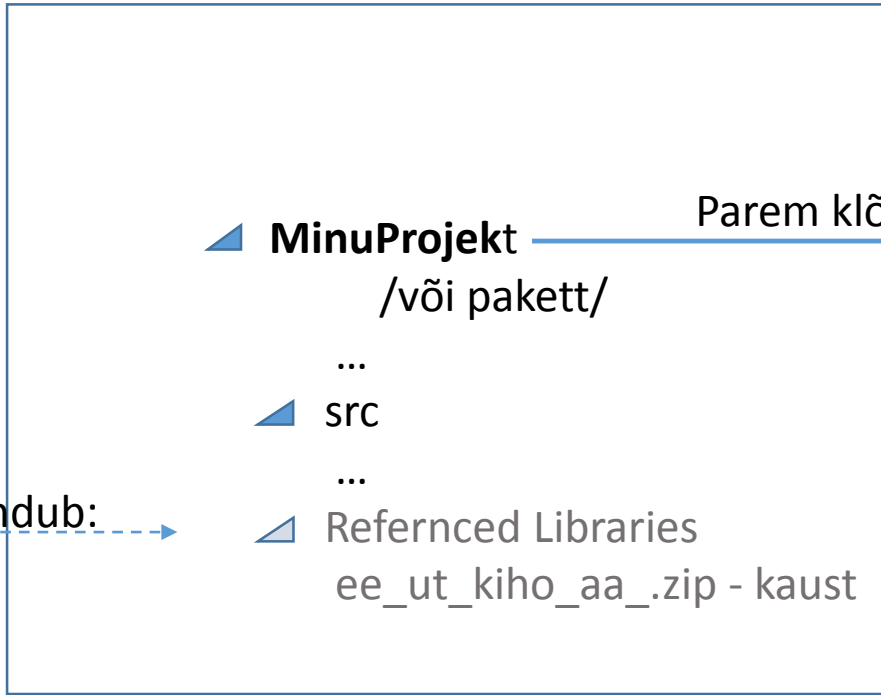


# ECLIPSE

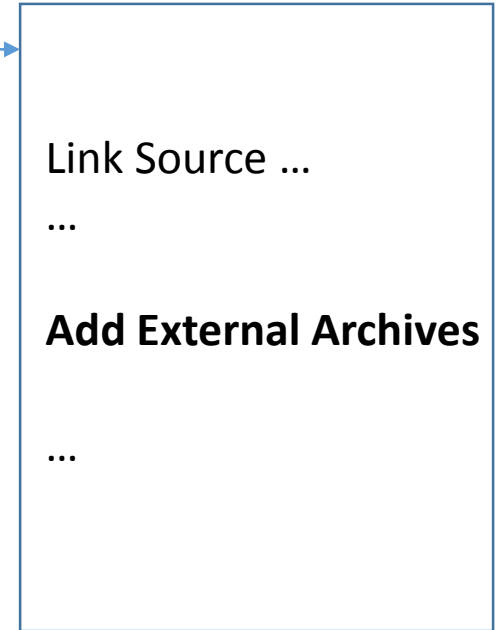
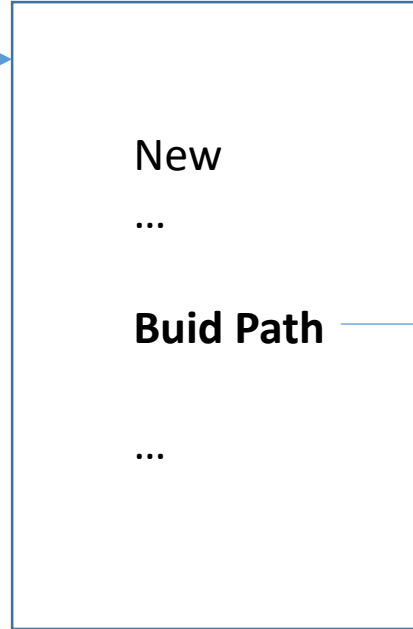
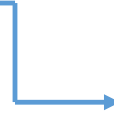
## Paketi *ee\_ut\_kiho\_aa.zip* lisamine klassiteele

```
tagamaks import ee.ut.kiho.aa.util.*;  
import ee.ut.kiho.aa.graaf.*;
```

### Package Explorer

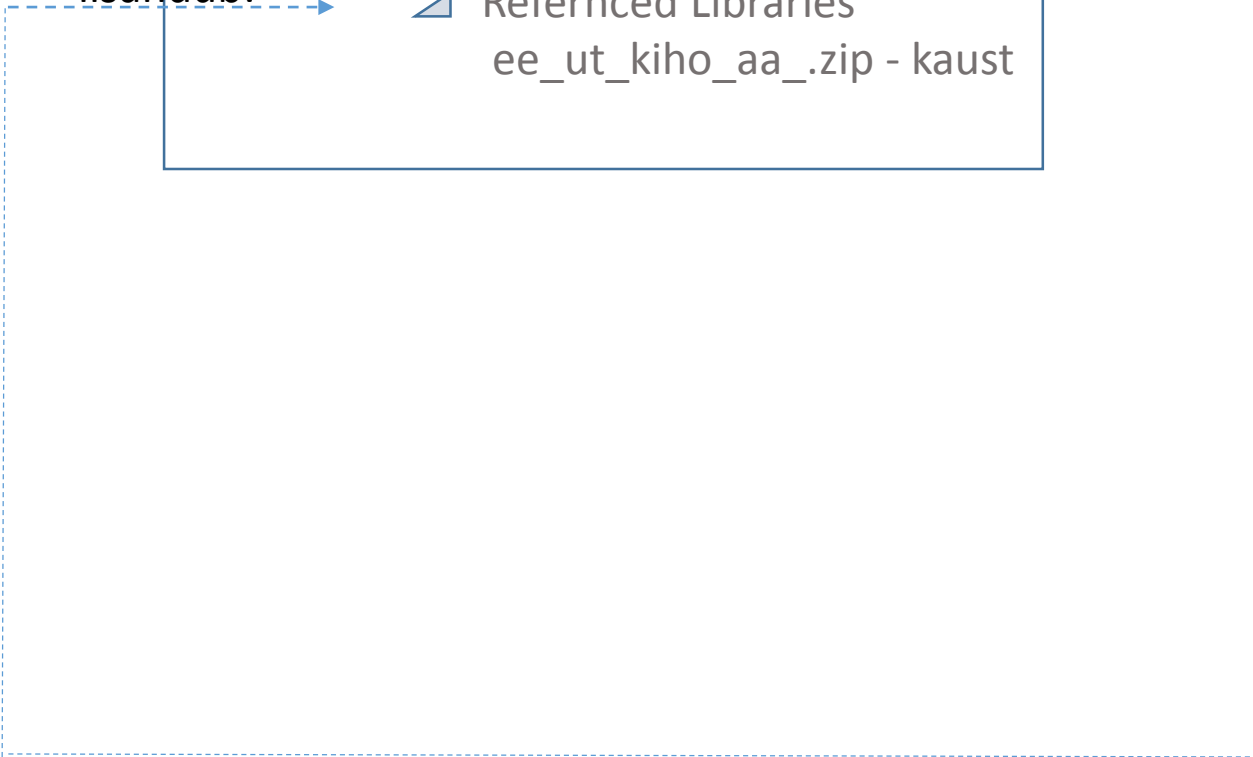


Parem klõps



Otsida fail: kaust\ ee\_ut\_kiho\_aa .zip  
avada

lisandub:



# Juhend *ee.ut.kiho.aa* pakettide lisamiseks IntelliJ IDEA arenduskeskkonda

Karl-Sander Erss

Lae alla [http://kodu.ut.ee/~kiho/ads/fall17/Paketid/ee\\_ut\\_kiho\\_aa.zip](http://kodu.ut.ee/~kiho/ads/fall17/Paketid/ee_ut_kiho_aa.zip)

Ava programm

Create New Project

Vali "Project SDK" (harilikult kaust C:\Program Files\Java\jdk1.8.0\_60)

Create project from template (Hello world)

Ava (File | Project Structure) aken, vali Modules-leht ja Dependencies vahekaart

Vajuta plussmärk-ikoonile, vali (Library | Java) ning vali allalaetud zip-fail.

Impordi pakist klass ja jookruta progamm (Run | Run 'Main')

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Katse &gt; src &gt; com &gt; company &gt; Main

Main

Project

Main.java

Katse C:\Users\Jyri\IdeaProjects\Katse

External Libraries

- < 1.8 > C:\Program Files\Java\jdk1.8.0\_31\bin\java
- ee\_ut\_kiho\_aa\_

  - ee\_ut\_kiho\_aa\_zip library root
    - ee.ut.kiho.aa
      - graaf
      - util

```
1 package com.company;
2 // kasutatakse klassi StopperNano paketist:
3 import ee.ut.kiho.aa.util.*;
4
5 public class Main {
6     public static void main(String[] args) {
7         StopperNano stopper = new StopperNano();
8         stopper.käima();
9         for(long l = 0; l < 3000000; l++){}; // ajaviide
10        stopper.kinni();
11        System.out.println("Aeg nano: " + stopper.näit());
12        int t = (int)(stopper.näit()/1000000); // millisek
13        System.out.println("Aeg millisek: " + t);
14    } //main
15 } //class Main
```

Run Main

```
"C:\Program Files\Java\jdk1.8.0_31\bin\java" ...
Aeg nano: 11042744
>> >> Aeg millisek: 11
```

Typo: In word 'paketist'

2:43 CRLF UTF-8

### AlgJava

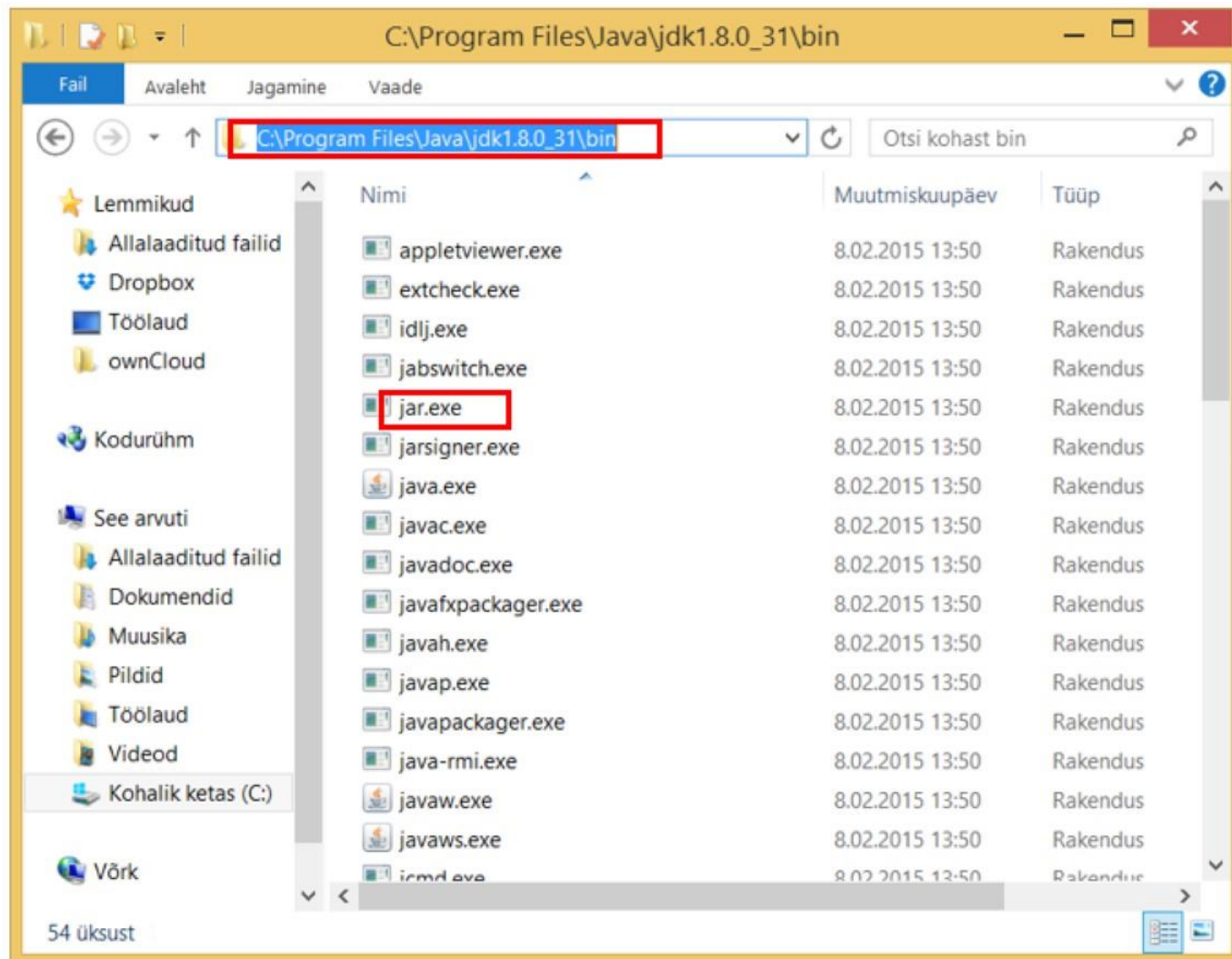
Praktikumijuhendaja esitab *Java*-näiteprogramme graafilisemal kujul, kasutades süsteemi *Amadeus\_AlJava*. Näiteprogrammide (*.algjava*) vaatamiseks on üliõpilasel võimalik see süsteem samuti installeerida

(<http://kodu.ut.ee/~kiho/AlgJavaDownload.shtml> ).

Programm *jar.exe* peab olema käivitatav käsurealt.

Keskkonnamuutujas *Path* kaustatee seadmine programmini *jar.exe* (kui pole veel seatud).

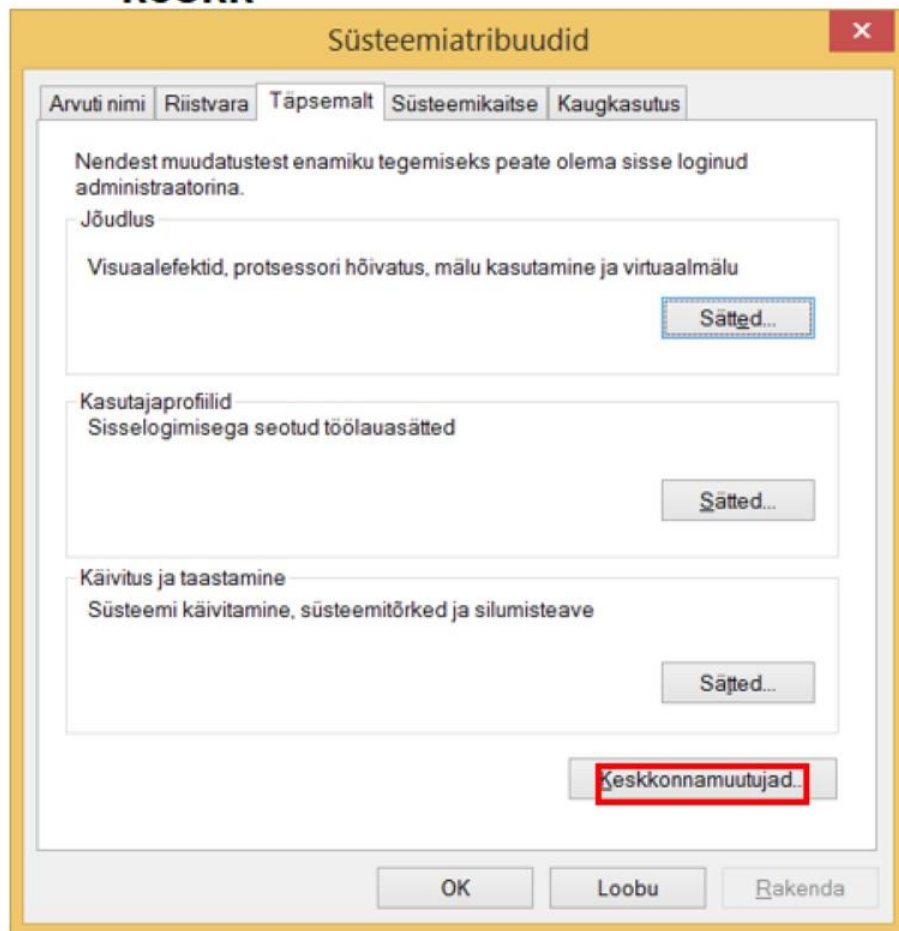
1. Leida kaustatee programmini *jar.exe*



Kaustatee programmini *jar.exe* on (antud juhul)  
`C:\Program Files\Java\jdk1.8.0_31\bin`

## 2. Redigeerida keskkonnamuutuja *Path*:

Otsi  
keskk



Lisada keskkonnamuutujasse *Path*  
(eraldajaga ';')

C:\Program Files\Java\jdk1.8.0\_31\bin

