

Ülesande lahenduse vormistamine

Nõudeid

Ülesandeks on koostada Java-programm (lahendusprogramm), mis sisaldab ühte või mitut, alamülesannet lahendavat meetodit. Koos ülesandega võidakse esitada ka lisanõudeid, samuti mõningaid soovitusi nn omatesti korraldamiseks; tuuakse sellise testi tulemuste väljundi näide.

Lahendusprogramm (näiteks *Nimi.java*) kirjutatakse ja silutakse ükskõik millises arenduskeskkonnas (*Eclipse*, *IntelliJ*, *JCreator*, *Notepad++* vm) ja operatsioonisüsteemi/versiooni/s (*Windows/..*, *Linux/..*, *macOS/..* vm). **Lubatud on kasutada vahendeid ainult Java standardsest rakendusteedist (API).** Lahendusprogramm ei suhtle kasutajaga, tulemuse (teksti kujul, mitte üle 2 lk) väljastab konsoolile. Tulemuse viimases reas peab leiduma väljastamise kuupäev ja kellaaeg (ajatempel).

Juhul kui lahendusprogramm kasutab mõnda andmefaili, siis peab sellekohane teave olema kirjas programmi päises. Lahendusprogramm võib olla ka mitmeklassiline. Vajalikud lisaklassid kirjeldatakse kas lahendusprogrammi siseklassidena või siis välisklassidena (failides **.java*).

Pärast silumise lõpetamist toimub **lahenduse vormistamine** ja **üleslaadimine**. Selleks:

- 1) lahendusprogramm *Nimi.java* koos selle poolt kasutatavate failidega kopeeritakse eelnevalt tühjendatud kausta;
- 2) käsurealt selles kaustas kompileeritakse (vajadusel eelnevalt sõltumatult kompileeritavaks redigeeritud) *Nimi.java*

`javac Nimi.java` (kui programmi teksti kodeering on *ANSI* (e *Windows* 1257))

või `javac -encoding utf8 Nimi.java` (kui programmi teksti kodeering on *utf-8*)

- 3) käivitatakse käsurealt selles kaustas (tavaliselt pärast kontrollkäivitust `java Nimi`)

`java Nimi > tulemus.txt` (st väljund suunatakse faili *tulemus.txt*)

- 4) kõik selles kaustas nüüd olevad failid

Nimi.java (+ välisklassid **.java*)

Nimi.class jt kompilaatori loodud klassifailid (**.class*)

tulemus.txt

[andmefailid (kui on) , nt **.txt*, **.csv* vmt]

moodustavadki **lahenduse** ja **laaditakse** (ühekaupa) Moodle'isse.

Tarvilik tingimus (selleks, et kodu- või kontrolltöö lahendus võetaks hindamisele):

Moodle'isse on üles laaditud nii testitulemuse fail „tulemus.txt“ kui ka kõik failid, mis on vajalikud

- 1) testitulemuse kuvamiseks lahendusprogrammi käsurealt käivitamisel;
- 2) lahendusprogrammi veatuks kompileerimiseks käsurealt.

Näiteks kõige lihtsamal juhul (nt ülesande *Kodune* korral) on lahenduse failideks

Kodune.java

Kodune.class

tulemus.txt

Käivitamine

```
java Kodune
```

väljastab konsoolile sama teksti (v.a ajatempel), mis on failis *tulemus.txt*.

Kompileerimine

```
javac Kodune.java
```

kulgeb veatult (uuendab faili *Kodune.class*).

Vorminõudeid

Java-programmi alguses peab (kommentaariblokina) olema esitatud **programmi päis**, nt

```

/*****
 * Programmeerimisharjutused. LTAT.03.007
 * 2020/2021 kevadsemester
 *
 * Kodutöö. Ülesanne nr /x/
 * Teema:
 *           Järjendi summa, nii iteratiivselt kui ka rekursiivselt
 *
 * Autor: Ülli Õpilane
 *
 * Kasutab andmefaile ..... Vajadusel
 *
 * Mõningane eeskuju: vt https:// ..... jmt
 *
 *****/
```

Ka iga sise- ja välisklass peab olema varustatud selgitava päisega (klassi otstarve, kasutamine, autor jmt).

Eesti keel olgu prioriteet, sh (muutujate, meetodite jmt) nimed ning kommentaarid.

Võimalust mööda hoiduda anglitsismidest!

Iga meetod (v.a peameetod *main*) peab olema varustatud oma **välise spetsifikatsiooniga** (otstarve, parameetrite ja tagastuse kirjeldus). Väline spetsifikatsioon võib olla *Javadoc* stiilis (eelneb vahetult meetodile) või esitatud reakommentaaridest koosneva lihttekstina kohe meetodi alguses:

```
/**                                                                 Javadoc
 * Järjendi summa leidmine rekursiivselt.
 * <p>
 * Rakendamine järjendi a summeerimiseks: summaRek(a, a.length).
 * @param a Antud järjend.
 * @param n Antud järjendi vaadeldava algusosa pikkus.
 * @return Järjendi a algusosa elementide summa a[0]+a[1]+...+a[n-1].
 */
static int summaRek(int[] a, int n){
```

```
static int summaRek(int[] a, int n){                                                                 Lihttekst
// Antud: järjend a ja selle vaadeldava algusosa pikkus n
// Tulemus: tagastatakse järjendi a algusosa elementide summa a[0]+a[1]+...+a[n-1]
// Rakendamine järjendi a summeerimiseks: summaRek(a, a.length)
```

Meetodi sees esitatakse olulisemad, eeskätt algoritmi ideed mõista aitavad kommentaarid.

Programmi(mõnes osa)s kasutatud või eeskujuks olnud välised (st mitte ainekursusest pärinevad) allikad peavad olema korrektselt kommentaaridena viidatud.

Programmi väljund peab olema kasutajale vahetult mõistetav, st varustatud vajalike selgitavate tekstilõikudega; esitab nii programmi töö tulemused kui ka lähteandmed ja/või nende iseloomustuse.