

Tallinna Tehnikaülikool

Arvutiteaduse instituut

2014/15 õppeaasta

Programmeerimise põhikursus Javas

ITI0011

EKSAMITÖÖ

II osa

Eksami I osa on kirjalik ning aega on 1,5 tundi (90 minutit)
Eksami II osa toimub arvutiga, aega on 1,5 tundi (90 minutit)

**Ülesanded tuleb lahendada arvutiga,
st tuleb kirjutada töötav kood, mis vastab etteantud nõuetele.
Iga ülesande õige ja korrektne lahendus annab 10 punkti.**

Kood tuleb üles laadida git'i repositooriumisse:

<https://git.ttu.ee/eksamid/iti0011/uniid.git>

Iga ülesande lahendus peab olema eraldi kaustas.

Kausta nimi on ette antud ülesande juures.

Lahendus peab olema ühes failis.

Lahendusega samas failis peab olema üks **main**-meetod (võib ka tühi olla).

Ei ole lubatud kasutada kalkulaatoreid, mobiiltelefone ega muid elektroonilisi abivahendeid.

Samuti pole lubatud kasutada konspekte, raamatuid ega muid kaasa võetud märkmeid.

ÄRA AVA eksamitööd enne vastava juhise saamist!

4 Rekursioon: alamsõne loendamine

Kirjuta rekursiivne funktsioon, mis võtab argumendina sisse kaks sõne ning tagastab, mitu korda leidub teine sõne esimese sees. Otsing peab olema *case-insensitive* ehk "KOER" ja "koer" on samaväärsed. (10p)
Mitte-rekursiivne lahendus annab 5p.

Näiteks:

```
findCount("lehm ja lammas ja siga", "ja") => 2
findCount("Lehm ja lehm", "lehm") => 2
findCount("tere tulemast", "e") => 3
findCount("Lehm, lammas ja lehm", "lehm") => 0
findCount("", "") = 0
findCount("test test", "") = 0
findCount("", "test") = 0
findCount(null, null) = findCount(null, "") = findCount("", null) = 0
Kaust gitis: E2EX4
```

```
public static int findCount(String text, String needle) {}
```

5 Sulgude kontroll

Kirjuta funktsioon, mis kontrollib, kas etteantud sõne vastab järgnevalt kirjeldatud reeglitele. Sisend koosneb sulgudest '(', ')', '[,]', '{, }'. Sõne on korrektne järgmistel tingimustel:

- Tühisõne (pikkusega 0) on korrektne
- Kui sõne A on korrektne, on ka (A), {A} ja [A] korrektsed
- Kui A ja B on korrektsed, on ka AB korrektne

Funktsioon tagastab väärtuse tõene (true), kui sisendsõne on korrektne, ja väär kõikidel muudel juhtudel. (10p)

Näiteks:

```
isCorrectParentheses("") => true
isCorrectParentheses("([{}[]]){}()") => true
isCorrectParentheses(null) => false
isCorrectParentheses("((){}])") => false
Kaust gitis: E2EX5
```

```
public static boolean isCorrectParentheses(String input) {}
```

6 Sorditud massiivide ühendamine

Kirjuta funktsioon, mis saab sisendiks kaks sorditud täisarvude massiivi ja tagastab uue massiivi, milles sisalduvad kõik kahes alguses massiivis olevad elemendid sorteeritult. (10p)

Näiteks:

```
[1, 2, 3] + [1, 2] => [1, 1, 2, 2, 3]
[1, 3, 6, 9] + [1, 2] => [1, 1, 2, 3, 6, 9]
[] + [1, 2, 3] => [1, 2, 3]
[1, 2, 3, 4] + [5, 6, 7] => [1, 2, 3, 4, 5, 6, 7]
null + [] => null
Kaust gitis: E2EX6
```

```
public static int[] combineSortedArrays(int[] arr1, int[] arr2) {}
```