

Praktikum 5

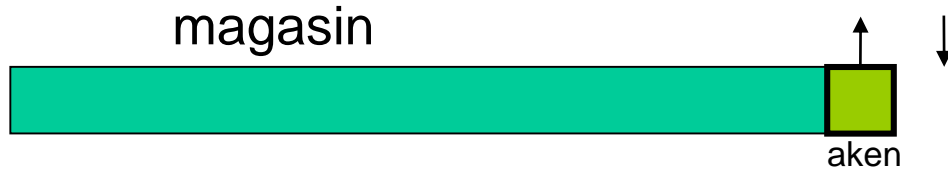
Magasin ja järjekord

Harjutusülesanded

- MJ_1. Suluavaldise arvutuseeskiri
- MJ_2. Sorteerimise magasinimeetod
- MJ_3. Sorteerimise järjekorrimeetod
- MJ_4. Avaldise infikskuju postfiks-kujuks
- MJ_5. Sorteerimise listimeetod

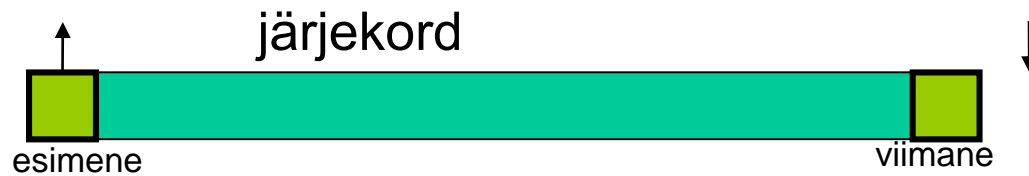
Iseseisev töö nr 3

Magasin



Operatsioon	Java
Teha uus, tühi	<pre>import java.util.Stack; // magasin Stack mag = new Stack(); Stack<tüüp> mag = new Stack<tüüp>();</pre>
Lisada element <i>x</i>	<pre>mag.push(x);</pre>
Aknal oleva väärtus	<pre>mag.peek();</pre>
Võtta (aknalt ära) Võtta aknalt ja omistada <i>y</i> -le	<pre>mag.pop(); y = mag.pop();</pre>
Kas on tühi?	<pre>if (mag.empty())</pre>

Järjekord



Operatsioon	Java
	<pre>import java.util.ArrayDeque; // (üld)järjekord</pre>
Teha uus, tühi	<pre>ArrayDeque<tüüp> jrk = new ArrayDeque<tüüp>();</pre>
Lisada element x järjekorra lõppu	<pre>jrk.addLast(x);</pre>
Võtta esimene Võtta esimene ja omistada y-le	<pre>jrk.pollFirst(); y = jrk.pollFirst();</pre>
Esimese elemendi väärtus	<pre>jrk.getFirst();</pre>
Viimase elemendi väärtus	<pre>jrk.getLast();</pre>
Kas on tühi?	<pre>if (jrk.isEmpty())</pre>

Harjutusülesanded

MJ_1. Suluavaldise arvutuseeskiri [Ülesannete kogu, ül nr 5.11]

Koostada ja testida programm järgmise ülesande lahendamiseks.

Antud: mingi suluavaldis (sõnena), näiteks

$$(((a + b) * (c - d * k) + 2.5) / (\pi * (1 - e)))$$

Kogu avaldis on ka sulgudes.

Tulemus: väljastatakse arvutuseeskiri, mis näitab millises järjekorras tuleb (võiks) arvutada selles esinevate suluavaldiste väärtused, näiteks:

```
Antud: (( (a+b) * (c-d*k) + 2.5) / (pi * (1-e) ) )
```

```
Arvutada:
```

```
S1 := a+b
```

```
S2 := c-d*k
```

```
S3 := S1*S2+2.5
```

```
S4 := 1-e
```

```
S5 := pi*S4
```

```
S6 := S3/S5
```

Programm peab kasutama magasini.

Edasijõudnud: tuvastada ka sulustatuse ebakorrektsus (“*invalid syntax*”).

```
Amadeus - workspace2016s\MJ_1_skeem.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2016oCDB

» MJ_1 põhitegevuse skeem
»»
String avaldis; antud suluavaldis
Stack mag = new Stack(); abimagasin mag
ArrayList<String> tulem = new ArrayList<String>();»
» listi tulem elemendiks on tulemuse üks rida
int nr = 1; muutja S number

+ ...»
» iga sümboli s korral antud avaldisest:
    if (s != ')')»
        ... mag <-- s (jrk sümbol avaldisest magasinini)
    - continue;»
» -- jrk sümbol avaldises oli sulgev sulg
»»
String muutuja = 'S'+Integer.toString(nr++); S1 või S2 või ...
»»
» Korjata osaavaldis magasinist välja, kuni tuleb avav sulg,
» ja siis muutuja (Snr) magasinini.
»»
tulem.add(muutuja + " := " + osaavaldis); jrk rida tulemusse
```

Stack<String> mag = new Stack<String>();

Sõnena: " "+s

MJ_2. Sorteerimise magasinimeetod [*Ülesannete kogu*, ül nr 7.12]

Koostada ja testida sorteerimismeetod, mis põhineb magasinidesse jaotamisel.

I osa: paigutada järjendi elemendid magasinidesse, igas mittekasvavalt, nt

[7, 4, 5, 3, 1, 2, 8, 6, 4] → [7, 4, 3, **1**], [5, **2**], [8, 6, **4**]

II osa: korjata magasinidest tulemusjärjend.

Vt ka järgmine slaid

MJ_3. Sorteerimise järjekorrameetod [*Ülesannete kogu*, ül nr 7.13]

Koostada ja testida sorteerimismeetod, mis põhineb järjekordadesse jaotamisel.

I osa: paigutada järjendi elemendid järjekordadesse,

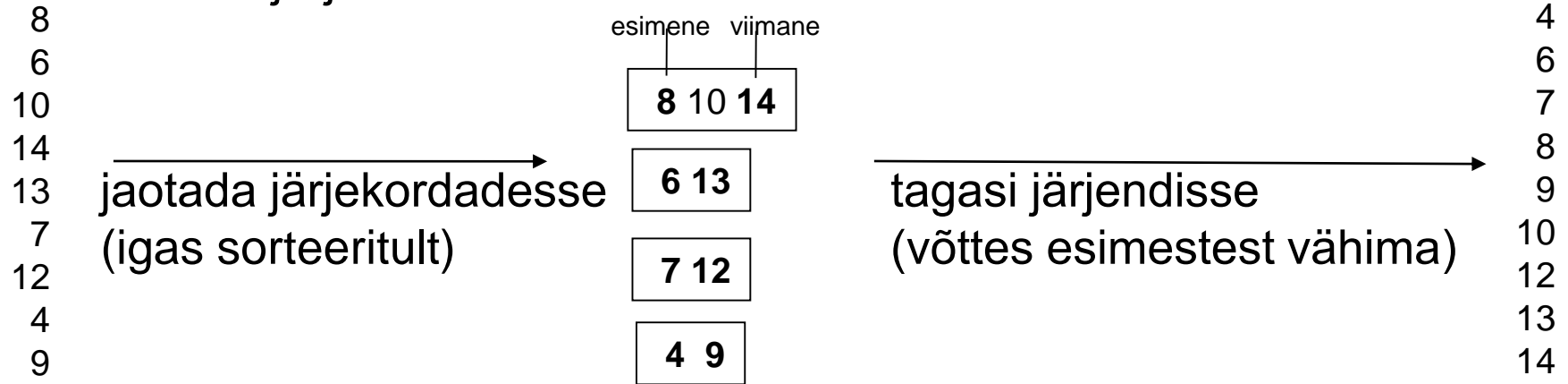
igas mittekahanevalt (suunas esimene->viimane), nt

[1, -7, 4, 5, 8, 3, 6, -2, 3, 4] → [1, 4, 5, **8**] [-7, 3, **6**] [-2, 3, **4**]

II osa: korjata järjekordadest tulemusjärjend.

Vt ka järgmine slaid

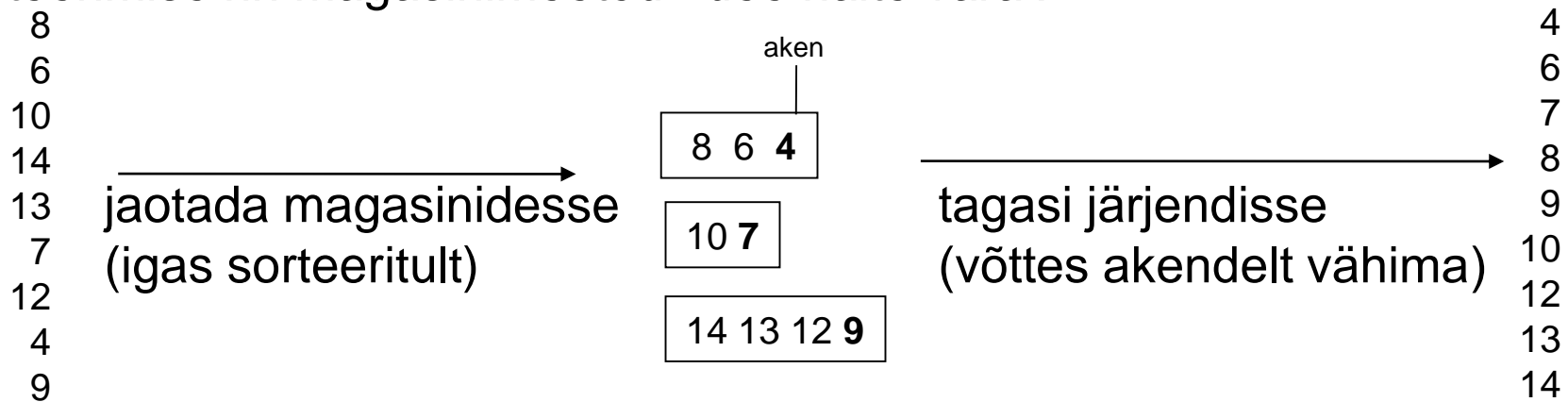
Sorteerimise nn järjekorrameetod. Idee näite varal:



Millisest klassist on selle algoritmi keerukus?

Millise järjendi korral töötab kõige kiiremini / kõige aeglasemalt?

Sorteerimise nn magasinimeetod. Idee näite varal:



Millisest klassist on selle algoritmi keerukus?

Millise järjendi korral töötab kõige kiiremini / kõige aeglasemalt??

MJ_4. Avaldise infikskuju postfikskujuks [Ülesannete kogu, ül nr 5.6]

Koostada ja testida meetod, mis (magasine kasutades) teisendab sõnena etteantud infikskujul aritmeetilise avaldise vastavaks postfikskujul avaldiseks.

Võib eeldada, et avaldises esinevad ainult binaarsed tehted ning infiks-kuju ja kõik tehete operandid peale arvude ja muutujate on sulgudesse võetud. Näiteks: “(a + (b²))”.

Testi tulemuse näide

Infiks-kujul avaldise	((5 + ((19 + 2)*4)) - 3)
postfiks-kujuks on	5 19 2 + 4 * + 3 -

Infiks-kujul avaldise	(a + (b ²))
postfiks-kujuks on	a b 2 ^ +

Infiks-kujul avaldise	((a1+b)*(x - (a2 ³)))
postfiks-kujuks on	a1 b + x a2 3 ^ - *

Infiks-kujul avaldise	((a - 1.53)*((b + (a + c))/(c + d)))
postfiks-kujuks on	a 1.53 - b a c + + c d + / *

MJ_5. Sorteerimise listimeetod [*Ülesannete kogu*, ül nr 7.14]

Koostada ja testida sorteerimismeetod, mis põhineb üldjärjekordadesse jaotamisel.

I osa: paigutada järjendi elemendid üldjärjekordadesse,
igas mittekahanevalt (suunas esimene->viimane), nt

[1, -7, 4, 5, 8, 3, 6, -2, 3, 4] → [-7, 1, 4, 5, 8] [-2, 3, 6] [3, 4]

II osa: korjata järjekordadest tulemusjärjend.

Meetodi skeem vt järgmine slaid

Amadeus - workspace2017s\ISE_3.algjava*

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2017oCDB

```
preambul
import java.util.ArrayDeque;»
import java.util.ArrayList;»

TEST

void listSort(int[] a)»
»
» Antud: täisarvumassiiv a
» Tulemus: a elementide väärtused ümber paigutatud mittekahanevalt
» Meetod: nn listimeetod
ArrayList<ArrayDeque<Integer>> abid = new ArrayList<ArrayDeque<Integer>>(); tühi
» 'abid' on koht abilistidele - üldjärjekordadele
    I. Paigutada järjendi a elemendid üldjärjekordadesse
    * i = 0 .. a.length-1»
    »
    panna(abid, a[i]);»
    I. Paigutada järjendi a elemendid üldjärjekordadesse
    II. Korjata arvud üldjärjekordadest massiivi a
    * i = 0 .. a.length-1»
    »
    a[i] = võtta(abid);»
    II. Korjata arvud üldjärjekordadest massiivi a

void panna(ArrayList<ArrayDeque<Integer>> abid, int x)

Integer võtta(ArrayList<ArrayDeque<Integer>> abid)
```

Iseseisev töö nr 3

Esitamise tähtaeg rühmal J. Liivi 2-202, E12 -14: **14. oktoober 2017, kell 12.15**

Esitamise tähtaeg rühmal J. Liivi 2-202, T14 -16: **15. oktoober 2017, kell 14.15**