Praktikum 1

Ajalise keerukuse empiiriline hindamine (I)

JAVA ELEMENTE

- **Järjendi esitusi Javas**
- Funktsioonitüüpi parameeter
- Ridade lugemine tekstifailist

Harjutusülesanded

- **AKE_1. Järjendiloome**
- **AKE_2. Sooritusaja mõõtmine**

Järjendi esitusi Javas

Fikseeritud pikkusega järjend: **massiiv**.

```
int[] a;
int[] b = new int[]{2, -3, 1, 0, 12};//ehk: int[] b = {2, -3, 1, 0, 12};
int[] c = new int[5];                //ehk: int[] c = {0, 0, 0, 0, 0};
```

Massiivi *d* pikkus (elementide arv): `d.length`

Massiivi *e* *i*-s element (*i* = 0 .. *e.length-1*): `e[i]`

```
import java.util.Arrays;
```

Massiivi *a* sorteerimine mittekahanevalt*:*

```
  Arrays.sort(a);
```

Massiivi *a* koopia => *b:*

```
int[] b = a.clone();
```

või

```
int[] b = Arrays.copyOf(a, a.length);
```

Muutuva pikkusega järjend: **list**.
Klassi *ArrayList* (või ka klassi *Vector*) isend. Listi elemendi väärtus on viidatüüpi.
ArrayList realiseerib liidese *List*.

```
import java.util.List;
import java.util.ArrayList;

ArrayList<Integer> a;
List<Integer> b = new ArrayList<Integer>(); //  listis b on 0 elementi
List<String>  c = new ArrayList<String>();  //  listis c on 0 elementi
```

Lisada *elem* listi *d* (viimaseks): `d.add(elem)`
Listi *e* pikkus (elementide arv): `e.size()`
Listi *f i*-s element (*i* = 0 .. *f.size()-1*): `f.get(i)`

```
import java.util.Collections;
```

Listi *lst* sorteerimine :

```
Collections.sort(lst); --  mittekahanevalt (lst elementide on nt tüüpi Integer või String)
Collections.sort(lst, elementide võrdlemise lambda-avaldis);
```

Listi *lst* sorteerimine mittekasvavalt:

```
Collections.sort(lst, Collections.reverseOrder());
```

Listi *lst* transponeerimine:

```
Collections.reverse(lst);
```

Listi *b* segamine*:*

```
Collections.shuffle(b);
```

Juhuvalim listist *b*, *n* elementi:

```
List<Integer> tmp = new ArrayList<Integer>(b);
Collections.shuffle(tmp);
List<Integer> valim = tmp.subList(0, n);
```

Funktsioonitüüpi parameeter

Java8, ühemuutuja funktsiooni edastamine

Formaalsete parameetrite loetelus, meetodi või konstruktori päises:

`(..., ..., Function<`*`argumenditüüp, tulemusetüüp`*`>) par_nimi, ...)`

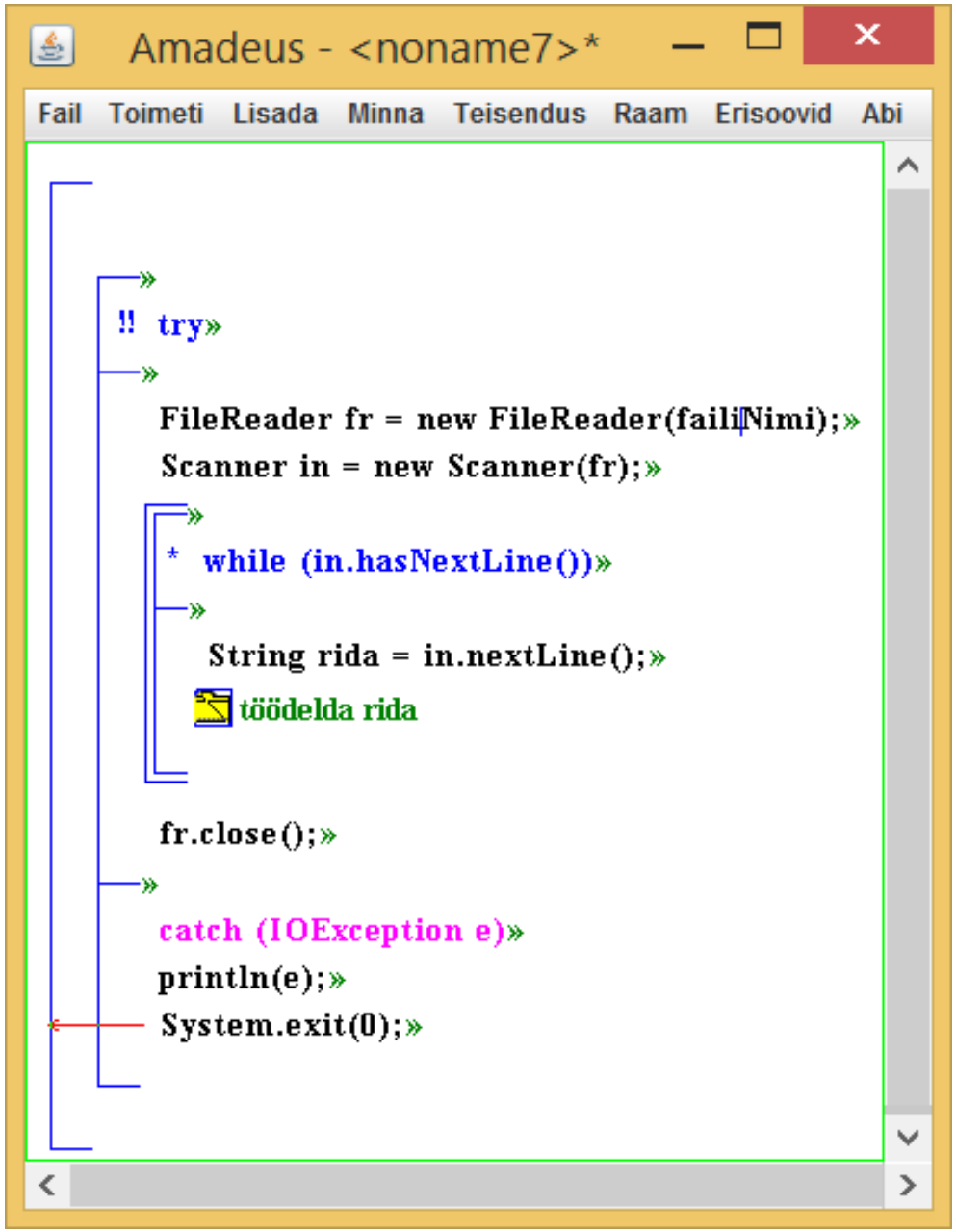
Rakendamine, selle meetodi või konstruktori kehas, avaldisena:

`par_nimi.apply(x)` --- *x* on argumenditüüpi, avadise väärtus on tulemusetüüpi

Tegelike parameetrite loetelus, meetodi või konstruktori väljakutses:

`(..., ..., lambda-avaldis, ...)`

Ridade lugemine teksifailist



Harjutusülesanded

AKE_1. Järjendiloome

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: loodava järjendi pikkus *n* ja tüübi tunnus *tüüp.*

Tulemus: nõutud tüüpi järjend (täisarvude massiivina) pikkusega *n.*

Tüübi tunnuseks olgu täisarv, näiteks:

Tüübi tunnus	*Loodava järjendi iseloom*
0	**Juhuarvud lõigult [-n, n] juhujärjestuses**
1	**Mittekahanev**
2	**Kasvav**
3	**Mittekasvav**
4	Kahanev
5	Konstantne
6	Umbes pooled elemendid korduvad
7	Üldiselt mittekahanev, kuid üks element keskel suur
8	Üldiselt mittekahanev, kuid üks element keskel väike
9	Paarisarvulise indeksiga elemendid kasvavalt, paarituarvulise indeksiga elemendid kahanevalt
10	**…**

Juhutäisarv lõigult $[a; b]$, kus $a < b$:

```
a+(int)(Math.random()*(b-a+1))
```

AKE_2. Sooritusaja mõõtmine

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: katsejärjendite pikkuste loetelu (massiivina)

Tulemus: katse(juhu)järjendite sorteerimisele kulunud ajad (massiivina).

Katsejärjendid on soovitavalt massiivid, sorteerimismeetod – `Arrays.sort`.

Testi tulemuse näide:

```
Nr      Järjendi        Sorteerimise
        pikkus          aeg (millisek)
---     -------         -----------
 0      100000          6.134579
 1      250000          15.709259
 2      500000          78.822029
 3      750000          87.518945
 4      1000000         109.976268
 5      1250000         127.592194
 6      5000000         423.839063
```

Aeg millisekundites:

```
long t0 = System.nanoTime(); // alustamise ajahetk
  ...<järjendi sorteerimine> ...
long t1 = System.nanoTime(); // lõpetamise ajahetk
  //järjendi sorteerimise kestus:
long nt = t1 - t0;                 // nanosekundites
double t = (t1 - t0)/1000000.0; // millisekundites
```

Meetodi
võimalik skeem: