

Harjutustund 6

Ülesannete kontroll. Klaviatuurilt andmete lugemine. Tekstiline sisend/väljund.

Klaviatuurilt andmete lugemine

Klaviatuurilt lugemiseks on vajalik importida pakett `java.io` (Input/Output).

```
import java.io.*;
```

Peameetodi juures on kirjas `throws IOException` - `BufferedReader`eri loomine on vaeohtlik käsklus ning tuleb seega kirjeldada. Peameetodi main esimese reaga loome vahendi, mille abil klaviatuurilt lugeda. Edaspidi võime `readLine` käsuga iga kord ühe rea inimese käest sisse lugeda.

```
public static void main(String[] argumendid) throws IOException{
    BufferedReader input =
        new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Your name:");
    String name=input.readLine();
    System.out.println("Hi, "+ name);
}
```

Saadud elemendid on sõned. Kui tahame neid kasutada näiteks täisarvudena või reaalarvudena, peame neid enne teisendama. Selleks saab kasutada vastavates *mähisklassides* olevaid meetodeid. Näiteks sõne täisarvuks teisendamisel on kasutatav meetod `Integer.parseInt`, mille argumendiks tulebki sõne anda. Näiteks oletame, et kui failis on igal real reaalarv (sõnena), siis

```
BufferedReader input=new BufferedReader(
    new InputStreamReader(System.in));

    System.out.println(" :");
    String str = input.readLine();
    double number = Double.parseDouble(str);
```

muudab selle `double`-tüüpi arvuks, millega saame tehteid teha.

Tekstiline sisend/väljund

Failidega suhtlemise saab *Javas* korraldada mitmeti. Siinkohal toome vaid ühe võimaluse.

Kõigepealt püüame määrata, millise failiga suhtlemine käib. Selleks loome klassi `java.io.File` isendi.

```
File fail = new File("c://temp//nimed.txt");
```

Kasutada saab mitmesuguseid meetodeid (vt. [API](#)), millest hetkel vaatleme meetodit `exists`, mille abil saame teada, kas antud fail eksisteerib või mitte.

```

if (fail.exists()) {
    System.out.println("Fail on juba olemas.");
}else{
    System.out.println("Faili ei ole olemas.");
}

```

Kui fail ei eksiteeri, siis same faili luua.

```

if (fail.createNewFile()){
    System.out.println("Fail on loodud!");
}

```

Faili pikkuse saame teada :

```

long length = fail.length();

```

Vajadusel võib faili nime ka muuta :

```

fail.renameTo(new File("c://temp//uued_nimed.txt"));

```

või faili kustutada :

```

fail.delete();

```

Püüame nüüd sinna faili midagi kirjutada. Selleks loome klassi `java.io.PrintWriter` isendi. Selles klassis on mitmeid konstruktoreid (vt. [API](#)), meie kasutame seda, mille argumendiks on fail.

```

PrintWriter pw = new PrintWriter(fail);

```

Selle rea lisamisel tekib aga veateade, mis räägib käsitlemata erindist (*unhandled exception*). Erinditest tuleb hiljem eraldi praktikum, siinkohal lahendame olukorra lihtsalt peameetodi päisele kahe sõna lisamisega.

```

public static void main(String[] args) throws Exception

```

Klassis `PrintWriter` on mitmeid meetodeid, sealhulgas ka meile tuttavad `print` ja `println`, mida nüüd kasutamegi.

```

pw.print("Mari Maarikas ");
pw.println("25");
pw.print("Johan Juurikas ");
pw.println("18");
pw.print("Peeter Tamm ");

```

```
pw.println("33");
```

Kui me enam juurde kirjutada ei taha, siis sulgeme voo.

```
pw.close();
```

Failist lugemiseks kasutame klassi `BufferedReader`. Isendi konstrueerimisel saab ette anda faili.

```
BufferedReader input =  
    new BufferedReader(new FileReader("c://temp//nimed.txt"));  
  
String line=null;  
  
do{  
    line= input.readLine();  
    if(line!=null)  
        System.out.println(line);  
} while(line!=null);  
  
input.close();
```

Analoogilisi meetodeid on teisi. Vt. ka [API](#).

Ülesanded koju

1. Kirjutada programm, mis küsib uut parooli ning kontrollib, kas paroolile esitatud nõuded on täidetud. Paroolis peab olema vähemalt 1 väiketäht, 1 suurtäht ning 1 number. Näiteks, "asdFg1" on korrektne. Kui ei, annab vastava teate. Kui jah, küsib parooli kontrolli mõttes teist kord.
2. Koostada programm kasutaja sisestatud arvu liitmiseks ja lahutamiseks. Luba lisaks valida, millist tehet sooritada.