

## 1. Java API kasutamine

**Teemad.** Tutvumine Java rakendusteebiga (API). Informatsiooni otsimine klasside ja meetodite kohta. API kasutamine programmeerimisel.

Pärast selle praktikumi läbimist oskab üliõpilane

- leida Java API-st infot kindla klassi kohta;
- leida Java API-st infot kindla meetodi kohta;
- kasutada Java API-t programmide koostamisel.

### Praktikumijuhend

Java suurt levikut ja arendustööd silmas pidades on klasside dokumentatsioon kokku kogutud spetsiaalses vormingus nn Java rakendusteeiki (*Application Programming Interface, API*). Java API on vabavara ja on kättesaadav veebilehelt

<http://java.sun.com/j2se/1.5.0/docs/api>

Terminite tõlkimisel inglise keelest eesti keelde või vastupidi võib abiks võtta J. Kiho raamatu „Väike Java leksikon“, Tartu Ülikool 2002.

Alustuseks tuletame meelde, kuidas Java programmi algteksti salvestada, kompileerida ja käivitada. Sisestada faili Tere.java järgmine programm.

```
class Tere {  
  
    public static void main(String[] args) {  
        System.out.println("Tervist!");  
    }  
}
```

Kompileerimiseks samas kataloogis anda käsurealt käsk

```
> javac Tere.java
```

Käivitamiseks samas kataloogis anda käsurealt käsk

```
> java Tere
```

Tulemusena väljastatakse

```
Tervist!  
>
```

Järgnevalt lühike näide Java API kasutamisest. Olgu vaja luua klass, mis väljastab ekraanile programmi käivitaja kasutajanime masinas, kuhu ta on sisse loginud, kusjuures on teada, et seda saab teha klassi `java.lang.System` abil. Selleks oleme klasside nimelist valla klassi `System` ja meenutame, et

key, String def). Dokumentatsioonist on näha, et kasutajanime teadasaamiseks võib kasutada meetodit `getProperty(String key)`, andes sõnetüüpi argumenti key väärtuseks "user.name". Seega programm, mis väljastab ekraanile kasutajanime, on järgmine:

```
class KesMaOlen {  
  
    public static void main(String[] args) {  
        String võti = "user.name";  
        String nimi = System.getProperty(võti);  
        System.out.println("Tervist, " + nimi);  
    }  
}
```

### Ülesanne 1

Täiendada näiteprogrammi `KesMaOlen` nii, et kui programm käivitatakse võtmege `-os`, siis väljastatakse ka operatsioonisüsteemi nimi ja versiooninumber.  
*Näide.*

```
> java KesMaOlen -os  
Tervist, ahtip. Tegutsed operatsioonisüsteemis  
Linux 2.4.20-31.9smp
```

### Ülesanne 2

Käsurealt antakse ette trigonomeetrilise funktsiooni nimi (`sin`, `cos`, `tan`, `cot`, `arcsin`, `arccos`, `arctan`, `arccot`), argumenti ühik (`-r` radiaanid, `-d` kraadid) ja argumenti väärtus `x`. Kasutades klassi `java.lang.Math`, luua programm, mis tagastab antud funktsiooni väärtuse kohal `x`.

*Näide.*

```
> java Trigo tan -d 120  
tan(120) = -1.7320508075688783
```

### Ülesanne 3

Koostada programm, mis väljastab ekraanile käsileva hetke kellaaja kujul `HH:MM.SS`, kus `HH` on tunnid, `MM` minutid, `SS` sekundid ja `s` millisekundid. Vajalikud meetodid leida klassist `java.util.Calendar`.

*Näide.*

## Iseseisev töö

### Ülesanne 4

Koostada täringuviskamist jäljendav programm, mis kasutab juhuslike arvude generaatorina esiteks meetodit `java.lang.Math.random()` ja teiseks meetodit `java.util.Random.nextGaussian()`. Meetodi `nextGaussian()` puhul võtta sobivate väärtuste lõiguks  $[-3, 3]$  ja iga tulemus, mis jääb sellest väljapoole, lageda ebaõnnestunud katseks. Saadud juhuslik väärtus teisendada üheks arvuku hulgast  $\{1, 2, 3, 4, 5, 6\}$ . Mõlema meetodi puhul teha 10000 viset ja väljastada ekraanile, kui palju igat numbrit tuli.

### Ülesanne 5

Luu programm, mis väljastab ekraanile käsurealt etteantava naturaalarvu 2nd-, 8nd- ja 16ndesituse. Vajalikud meetodid leida klassist `java.lang.Integer`.

### Ülesanne 6

Luu programm bitijadadega sooritatavate tehtete and, or ja xor demonstreerimiseks. Vajalikud meetodid leida klassist `java.util.BitSet`.

### Ülesanne 7

Luu programm, mis käsurealt ette antud klassinime korral väljastab kõik selle klassi ülemklasside nimed ja ülemklassides sisalduvate meetodite loetelu. Vajalikud meetodid leida klassist `java.lang.Class`.

Ülesannete 8–10 lahendamiseks vajalikud meetodid leida klassidest `java.util.Locale` ja `java.util.Currency`.

### Ülesanne 8

Koostada programm, mis väljastab ekraanile kohalikus Java virtuaalmasinas realiseeritud standardid riikide jaoks kujul

```
riigi nimetus, riigikeel, riigis kehtiv rahaühik
```

### Ülesanne 9

### German korral on tulemuseks

```
Austria EUR
Germany EUR
Switzerland CHF
Luxembourg EUR
```

### Ülesanne 10

Koostada programm, mis käsurealt ette antud rahaühiku tähise korral leiab riigid, kus see rahaühik kehtib. Näiteks sisendi ALL korral on tulemuseks Albania.