

# Objektorienteeritud programmeerimine

11. veebruar 2019

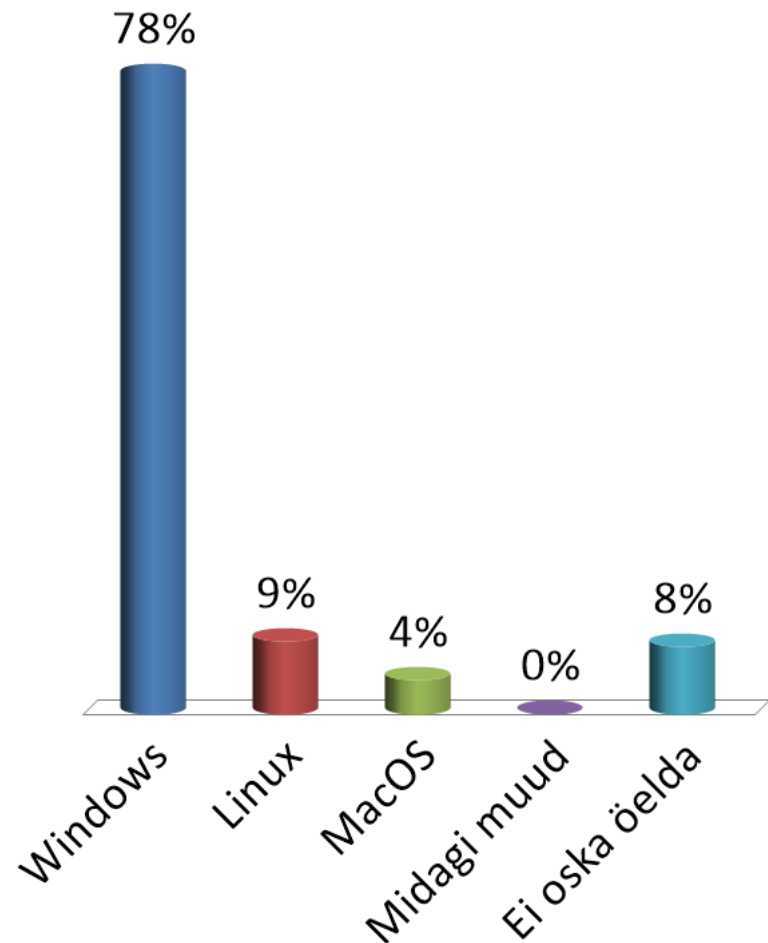
Marina Lepp

ATI informaatika dotsent

[marina.lepp@ut.ee](mailto:marina.lepp@ut.ee)

# Milline operatsioonisüsteem on paigaldatud teie arvutisse?

1. Windows
2. Linux
3. MacOS
4. Midagi muud
5. Ei oska öelda



# Tänane plaan

- Sissejuhatus
  - organisatoorselt
    - Milleks?
    - Kuidas?
    - Millal?
    - Kas?
  - teemasse
    - Java
    - Esimene programm
    - ...
    - ...

# Milleks OOP?

- Silmaring, maailmavaade
  - objektid, subjektid
- Õppimine
  - jätk esimese semestri programmeerimisele
    - Programmeerimine
    - Programmeerimise alused, Programmeerimise alused II

# Eeldus mitmetele ainetele

- LTAT.03.005 Algoritmid ja andmestruktuurid
- LTAT.05.003 Tarkvaratehnika
- LTAT.05.004 Veebirakenduste loomine
- MTAT.03.032 Kasutajaliideste kavandamine
- MTAT.03.158 Programmeerimine keeles C++
- MTAT.03.315 Programmeerimismustrid arvutimängudes
  
- Lävimõiste arvutiteaduses

# Keel

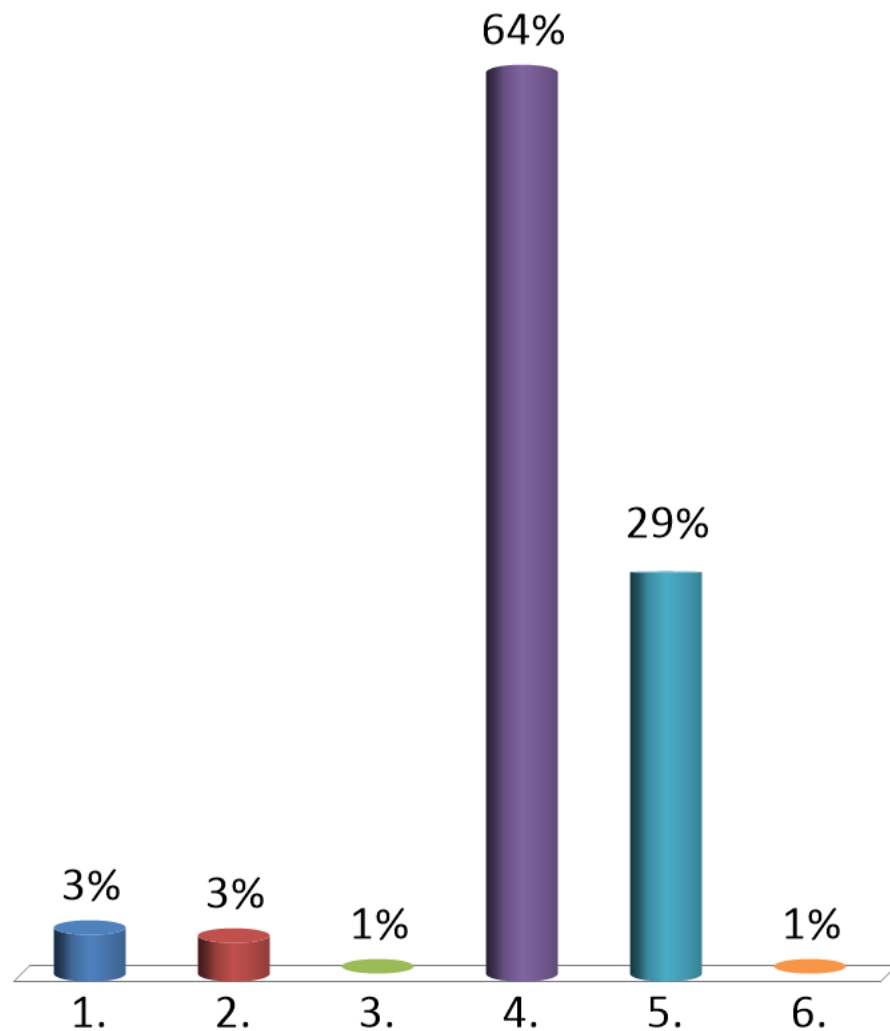
- Java
  - populaarsus
    - <https://www.tiobe.com/tiobe-index/>
  - töökuulutused
    - <http://www.cvkeskus.ee>
    - <http://www.cv.ee/too/infotehnoloogia/q-java>

# Inimesed on erinevad

- Siin võib-olla neid, kes
  - õpivad hoopis muud eriala
  - programmeerivad igapäevaselt suure innuga
  - said *Programmeerimises* hinde E
  - on edukalt osalenud (rahvusvahelisel) informaatikaolümpiaadil
  - suhtlevad mingis programmeerimiskeeles palju vabamalt kui mistahes inimkeeles
  - on selle aine püsikliendid
  - ...

# Mis valdkonnast te olete?

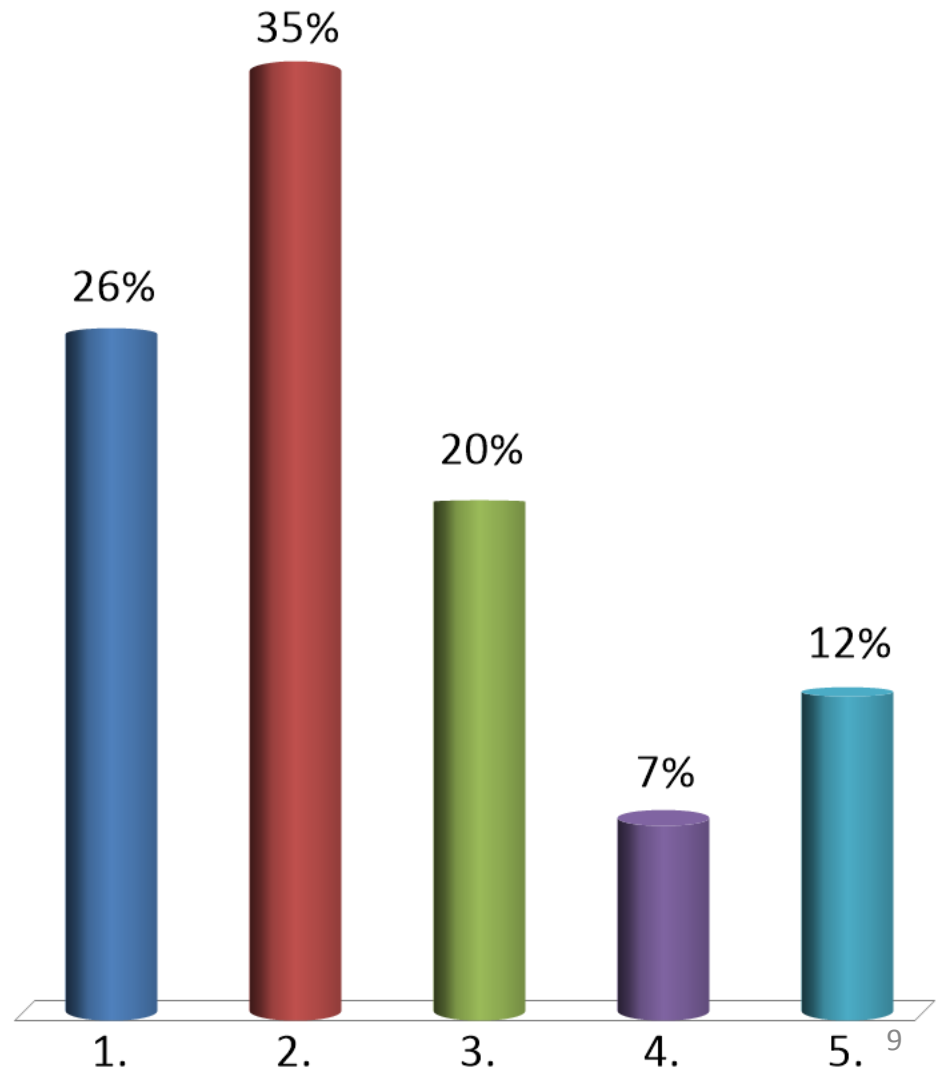
1. humanitaarteaduste ja kunstide valdkond
2. sotsiaalteaduste valdkond
3. meditsiiniteaduste valdkond
4. loodus- ja täppisteaduste valdkond, **informaatika**
5. loodus- ja täppisteaduste valdkond, **muu**
6. muu





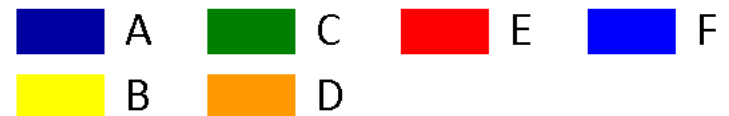
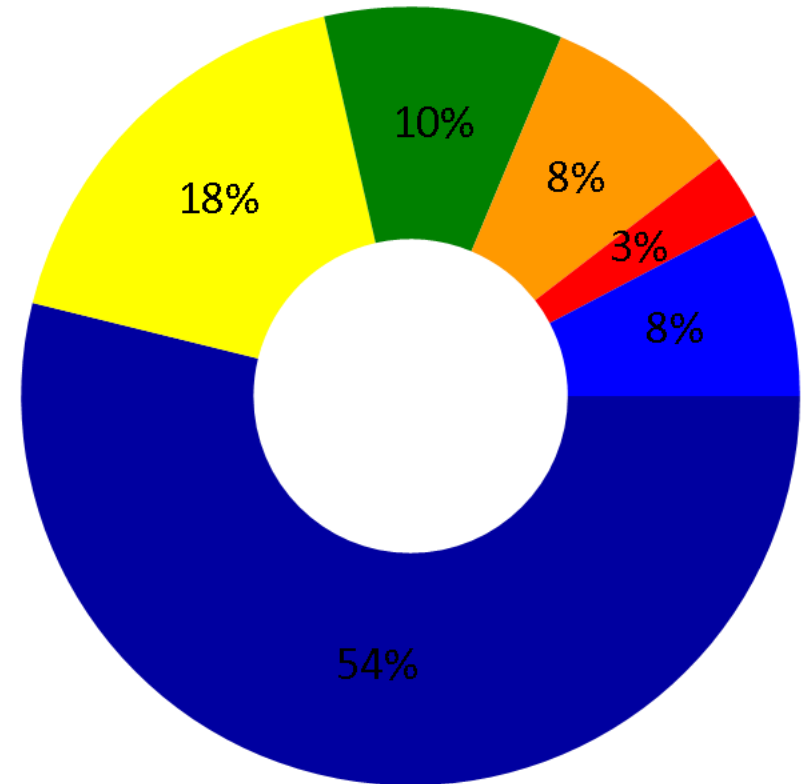
# Programmeerimise (Progr. alused II) kursus oli

1. väga lihtne
2. lihtne
3. paras
4. keeruline
5. väga keeruline



# Millise hinde saite Programmeerimise (Progr. alused II) kursusel?

- A. A
- B. B
- C. C
- D. D
- E. E
- F. F



# Kursus

- 6 EAP → 6\*26 tundi tööd
- Loengud
  - esmaspäeviti 12-14
    - põhirühm
    - edasijõudnute rühm
  - mittestatsionaarne rühm
- Praktikumide tunnitööd, kodutööd
  - suhteliselt kindlas tempos, kindlad teemad
  - arvutiklassis, tavalises auditooriumis oma sülearvutiga
  - alates 2. nädalast toimuvad K18 lisatunnid
- Kontrolltööd
  - ajapiiriga
- Rühmatööd
  - ISE!
- Eksam

# Millele (veel) tähelepanu?

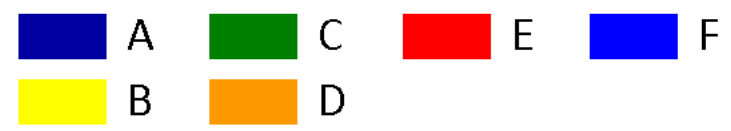
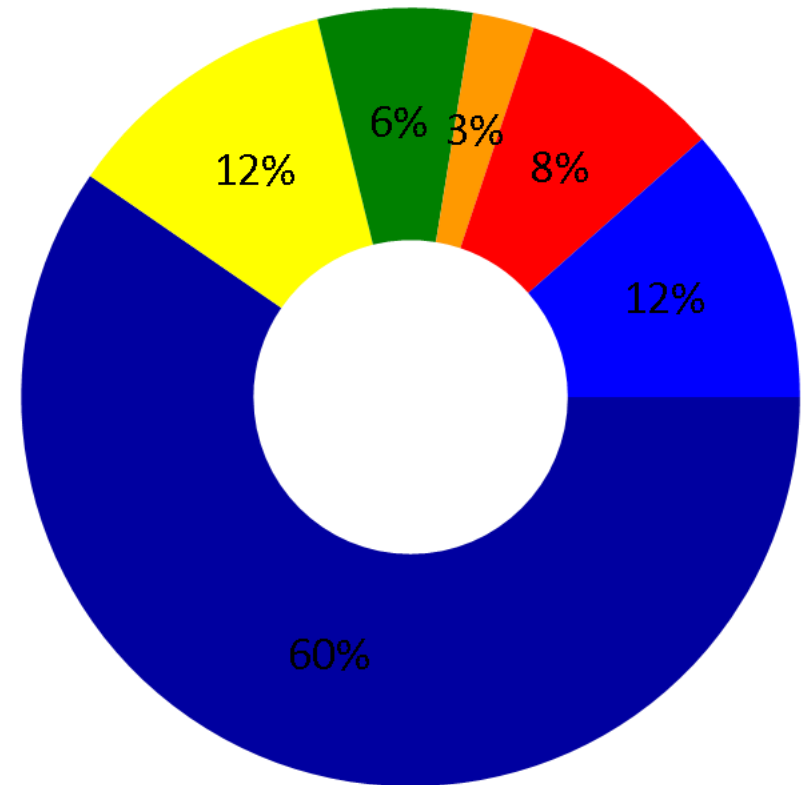
- Looming
  - programmeerimine
- Keskendumine
  - Loengusse-praktikumi selleks, et õppida (luua tingimused õppimiseks)
  - piisavalt iseseisvat õppimist tunnivälisel ajal
- Koostöö
  - üliõpilased omavahel
    - paaristööd praktikumis
    - rühmatöö
    - foorum
  - üliõpilane ja õppejõud
    - küsitakse-vastatakse
    - ÕISis registreeritud aadressid, slack
- Eestikeelne terminoloogia
  - kui meie ei kasuta ja ei arenda, siis ei tee seda (varsti) keegi

# Hindamine

- Loengud
  - 12 punkti
- Praktikumid
  - 12 punkti
  - 1. kontrolltöö, 16 punkti
  - 2. kontrolltöö, 16 punkti
  - 1. rühmatöö, 5 punkti
  - 2. rühmatöö, 5 punkti
  - rühmatöö esitlus, 3 punkti
- Eksamitöö, 33 punkti
  - Eksamile pääsemiseks peab olema praktikumidest (sh. rühmatööd) kogutud **vähemalt 28** punkti (sh 1. KT **12** punkti), loengutest **vähemalt 6** punkti. Eksamil peab koguma **vähemalt 15** punkti.
- Hindeskaala
  - "E" 51-60, "D" 61-70, "C" 71-80, "B" 81-90, "A" 91-

# Millise hindega oleksite selles aines rahul?

- A. A
- B. B
- C. C
- D. D
- E. E
- F. F



# Väljavõtteid tagasisidest

- Kes enne pole javaga tuttav, võiks arvestada, et igal nädalal kulub ainele 6-8 tundi või rohkemgi. Tasub semestri algusest saati pidevalt ainega tegeleda.
- Ära jää maha, tegele igapäev vähemalt tunnike programmeerimisega.
- Tehke kogu aeg kaasa ja kõik kodutööd ka, muidu jääte maha ja tunne pole väga hea.
- Tee kõik kodused ülesanded ära ja aine saab väga selgeks. Need võivad tihtipeale rohkem aega võtta kui oled alguses planeerinud, et ei soovita neid viimasele päevale jätta.

# Väljavõtteid tagasisidest

- Kui on olemas selle ainega seotud minged raskused, tule kindlasti lisapraktikumile.
- Valige projektiks teema, mis teid huvitab - siis läheb töö libedamalt.
- Nõuab palju iseseisvat tööd. Algul võib olla raske, aga kui hakkad mõistma, siis on täitsa sõltuvusttekitav.
- Selle aine põhimõtetest on vaja igal programmeerimisega tegeleval isikul aru saada. Soovitan materjalid selgeks teha lisaks hea hinde pärast ka enda jaoks, seda läheb tulevikus vaja.



# Väljavõtteid tagasisidest

- Eriti meeldis loengus suhteliselt karmilt distsipliini hoidmine. Leian, et see on väga vajalik.
- Loengutes kohe oli hea käia, sest vahete-vahel pidi kaasamõtleva ja nuppu vajutama, uni ei tulnud peale.
- Meeldis, et loengupunkte oli võimalik ka tagantjärgi videode vaatamisega teenida. Kuigi seda ise kordagi ei kasutanud, siis alati oli teadmine, et see võimalus on olemas.

# Eesmärk

- Kursuse eesmärgiks on anda alusteadmised objekt-orienteeritud programmeerimise eripärast, oskused programmide koostamiseks ning esmased rühmatööoskused
- Kursuse läbinud üliõpilane:
  - oskab selgitada objekt-orienteeritud paradigma põhimõisteid (kapseldus, abstraktsioon, pärimine, polümorfism, üledefineerimine, ülekate) ning analüüsida vastavaid programme
  - oskab kirjeldada erinevaid andmestruktuure (massiiv, ahel, magasin, järjekord, paisktabel) ja nende kasutusviise
  - oskab selgitada rakendusteekide väärtust ja olemust ning leida nendest vajalikku informatsiooni
  - oskab selgitada sündmuspõhise programmeerimise eripära ja erindite käitlemist ning tuua näiteid nende kasutamisest
  - oskab ühes objekt-orienteeritud programmeerimiskeeles kasutades integreeritud programmeerimiskeskonda koostada, testida ja siluda programme, rakendades selleks eelmistes punktides loetletut
  - oskab kirjeldada isikliku kogemuse põhjal rühmaprojekti keskseid elemente

# Tahaks juba programmeerimisest ka kuulda

- Objektorienteeritud programmeerimine
  - programmeerimiskeel Java
- Slaidide allikad
  - Helle Heina, Eno Tõnissoni slaidid
  - Y. Daniel Liang *Introduction to Java programming*
  - <http://docs.oracle.com/javase/tutorial/>

# Täna

- Java ajaloost
- Java omadused
- Esimene programm
- Muutujad ja tüübid
- Põhikonstruktsioonid

# Java

- James Gosling, *Sun Microsystems*
- *Oak, Green*
- 1995
- Nimest
  - Jaava saar, jaava keel
    - 100 miljonit, piirkondlik keel, pole päevalehti
  - Kohv
    - Coffee.
    - 'I'm dying for a cup of java'
    - <https://en.oxforddictionaries.com/definition/java>
  - <http://www.javaworld.com/javaworld/jw-10-1996/jw-10-javaname.html>
- Ajaloost
  - <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>

# Java

- Lihtsus
- Objektorienteeritus
- Staatiliselt rangelt tüübitud
- Turvalisus
- Arhitektuurist sõltumatus
  - write once, run anywhere
  - baitkood
- Multilõimelisus
- Dünaamilisus
- Vt.
  - <https://docplayer.net/15119984-Characteristics-of-java-optional-y-daniel-liang-supplement-for-introduction-to-java-programming.html>
  - <http://www.oracle.com/technetwork/java/langenv-140151.html>

# API, JDK, IDE

- *API – Application Programming Interface* – rakendusteek, eelnevalt programmeeritud objektid (Java) programmi koostamiseks
  - meil Java SE 11
- *JDK – Java Development Toolkit* – Java programmide koostamise tarkvara komplekt
  - meil JDK 11, võiks installida enne 1. praktikumi
- *IDE – Integrated Development Environment* – integreeritud programmeerimiskeskond
  - IntelliJ IDEA Community või Ultimate (ut.ee aadressiga)
  - Eclipse IDE

# Tehnoloogiad

- Java SE
  - Standard Edition
- Java EE
  - Enterprise Edition
- Java Cloud
- Java Embedded
  - Java SE Embedded
  - Java ME
    - Micro Edition
  - Java Card
  - Java TV

<https://www.oracle.com/java/technologies/>

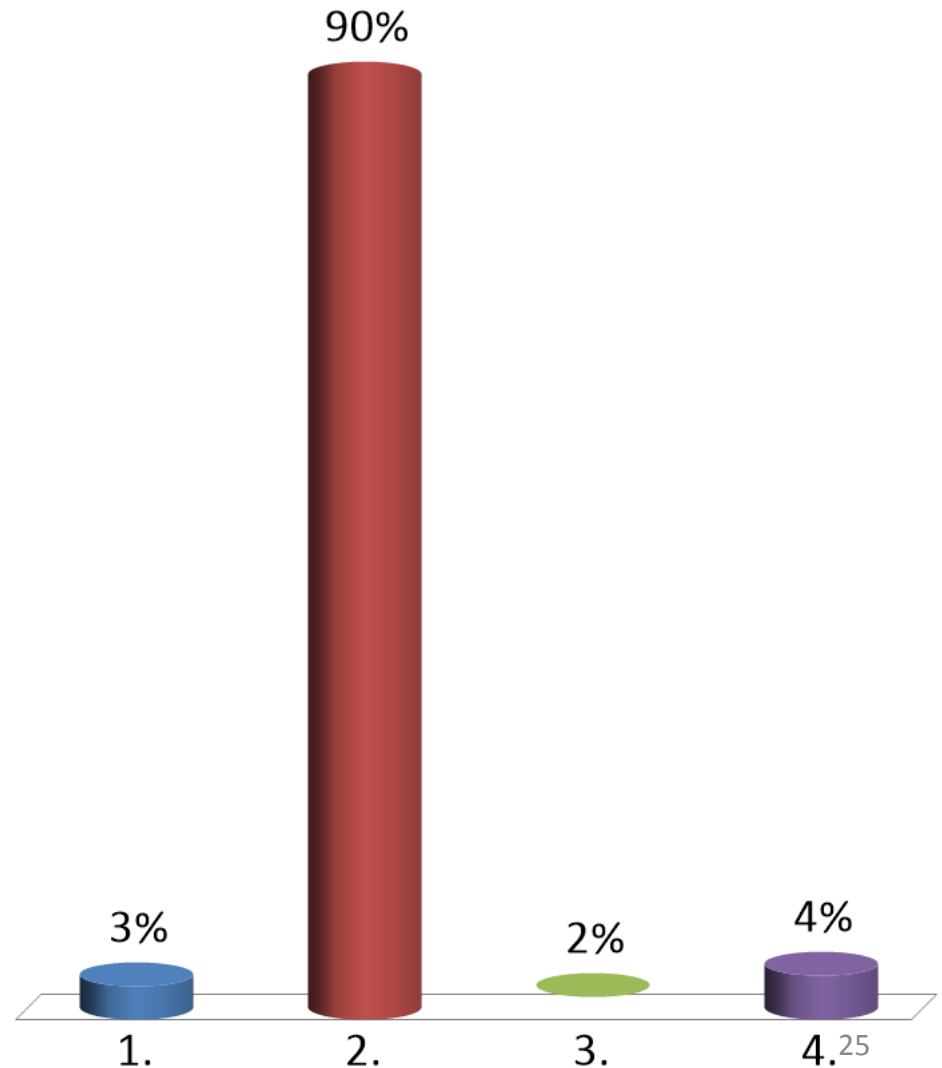
- JavaFX
  - Client Application Platform

<https://openjfx.io/>



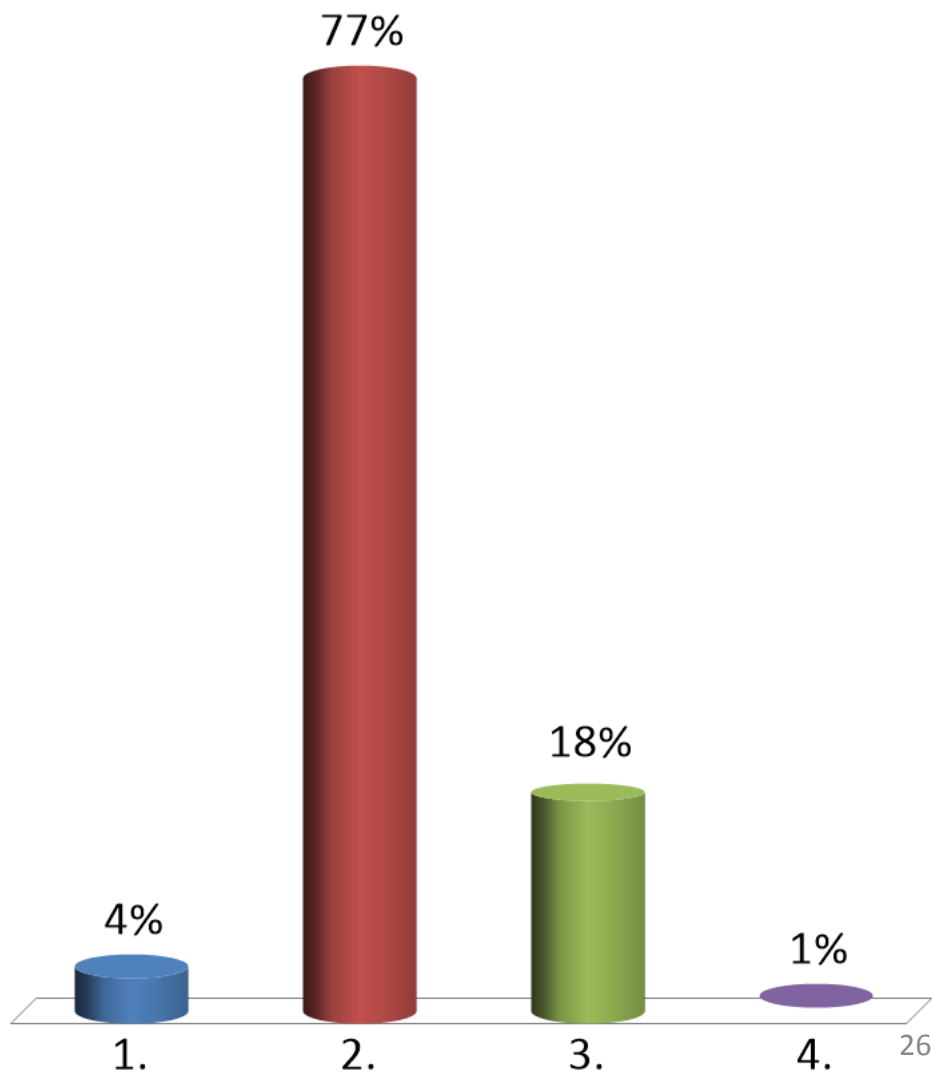
# Mis firmas Java loodi?

1. Dragon Ltd
- ✓ 2. Sun Microsystems
3. Smilers Computers
4. Grape International



# Java programmide koostamise tarkvara komplekt on

1. JFK
- ✓ 2. JDK
3. API
4. ACTA



# Esimene programm

```
/*  
Mitmerealine kommentaar  
*/
```

```
//Üherealine kommentaar
```

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

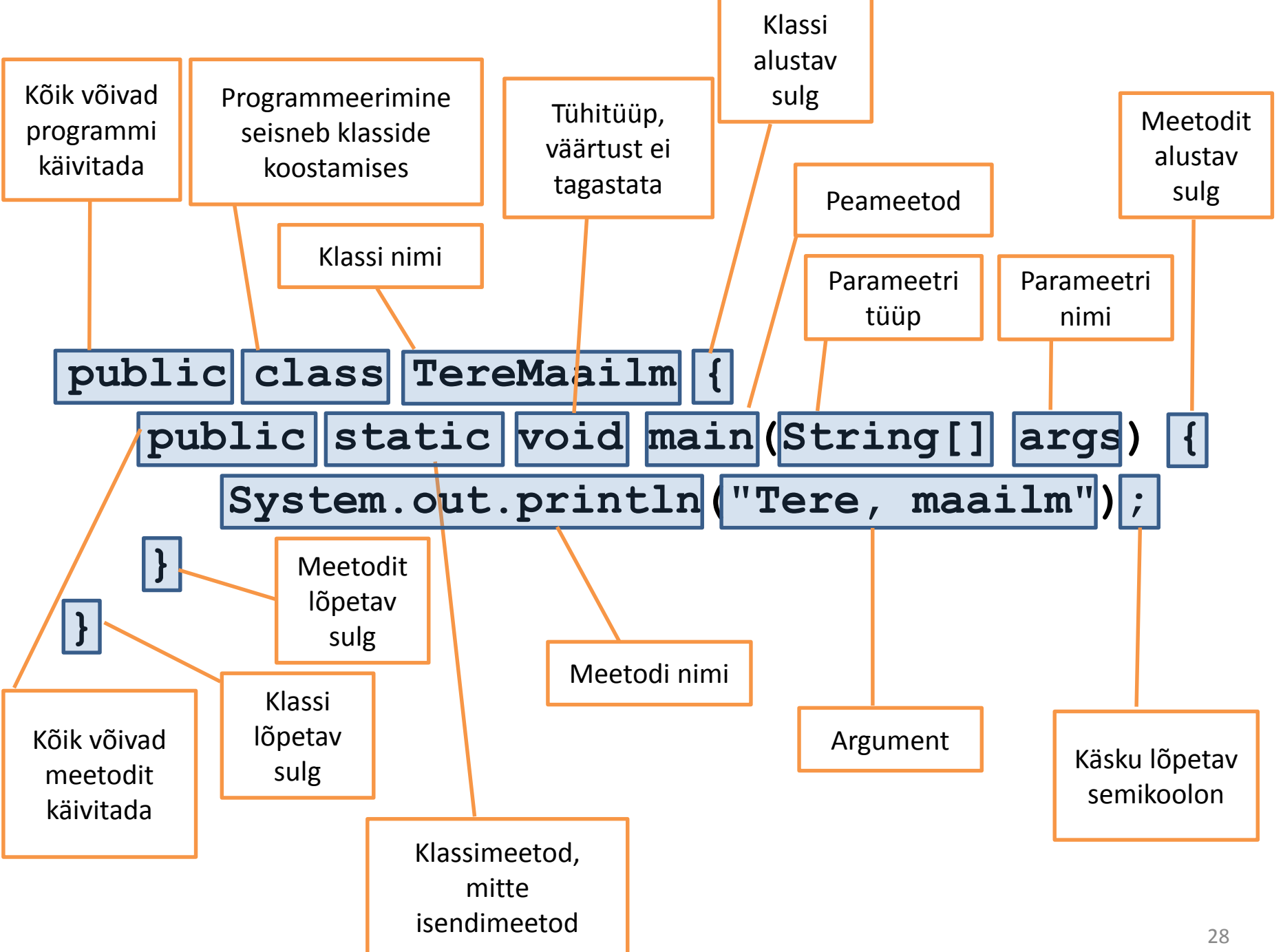
Python:

```
print("Tere, maailm")
```

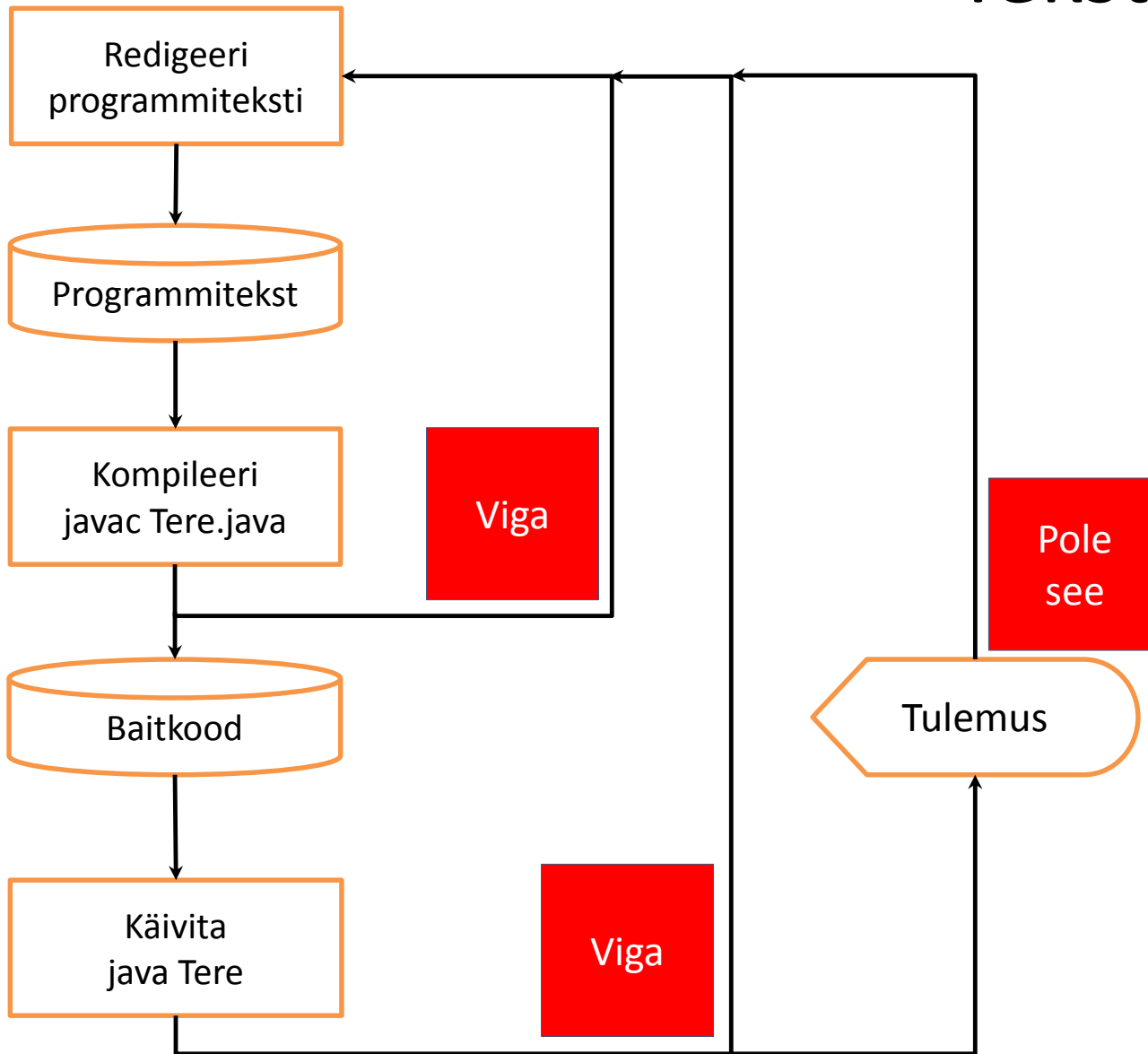
Fail nimega: **TereMaailm.java**

Kompileerimine: **>javac TereMaailm.java**

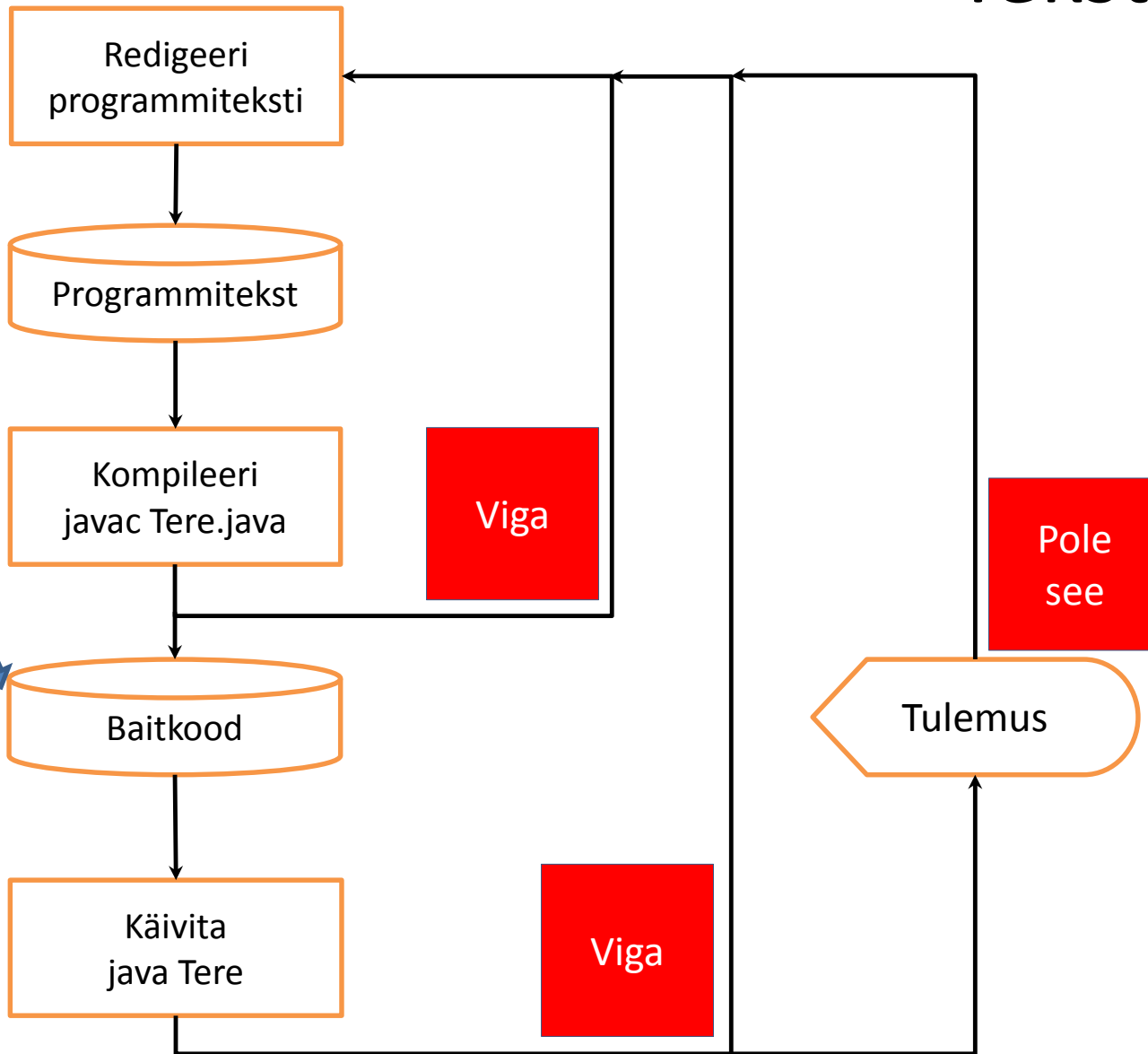
Käivitamine: **>java TereMaailm**



# Tekstist tööle



# Tekstist tööle



# Baitkood

```
H:\oop19> javap -c TereMaailm
```

```
Compiled from "TereMaailm.java"
```

```
public class TereMaailm {
```

```
    public TereMaailm();
```

```
    Code:
```

```
    0: aload_0
```

```
    1: invokespecial #1 //Method java/lang/Object."<init>":()V
```

```
    4: return
```

```
public static void main(java.lang.String[]);
```

```
    Code:
```

```
    0: getstatic    #2 //Field java/lang/System.out:Ljava/io/PrintStream;
```

```
    3: ldc        #3 //String Tere, maailm
```

```
    5: invokevirtual #4 //Method java/io/PrintStream.println:(Ljava/lang/String;)V
```

```
    8: return
```

```
}
```

# Baitkood

- keel JVM (*Java Virtual Machine*) jaoks

<https://docs.oracle.com/javase/specs/jvms/se11/html/index.html>

- Võib jõuda erineval moel
  - javac
  - Jruby (Ruby)
  - Jython (Python)
  - Clojure (spetsiaalselt JVM jaoks)
  - ....



# Millest programm koosneb?

- Kommentaarid
- Võtmesõnad
  - **public, class, static, void**
- Piiritlejad
  - **public, static**
- Laused
  - lõppevad üldiselt semikooloniga
- Plokid
  - { }
- Klassid
- Meetodid
  - Peameetod **main**

# Võtmesõnad

abstract assert boolean break byte case  
catch char class const\* continue default  
double do else enum exports extends  
final finally float for goto\* if  
implements import instanceof int  
interface long module native new package  
private protected public requires return  
short static strictfp super switch  
synchronized this throw throws transient  
try void volatile while

- Tunduvad võtmesõnadena, aga on literaalid  
`false` `null` `true`
- `var`

# Muuttujat ja tyybid

Igal Java muuttujal on

- tyypp
  - algtyypp või viittyypp
- nimi
- väärtus

```
int x = 17;
```

# Algtüüp

- Täisarvud
  - `byte`
  - `short`
  - `int`
  - `long`
- Ujukomaarvud
  - `float`
  - `double`
- Tõeväärtused
  - `boolean`
- Sümbolid
  - `char`

# Täisarvud

- Teine täiendväärtus (two's complement)
  - **byte**
    - 8-bitiline
    - -128 kuni 127
  - **short**
    - 16-bitiline
    - -32 768 kuni 32 767
  - **int**
    - 32-bitiline
    - -2 147 483 648 kuni 2 147 483 647
  - **long**
    - 64-bitiline
    - -9 223 372 036 854 775 808 kuni 9 223 372 036 854 775 807
      - üheksa kvintiljonit kakssada kakskümmend kolm kvadriljonit kolmsada seitsekümmend kaks triljonit kolmkümmend kuus miljardit kaheksasada viiskümmend neli miljonit seitsesada seitsekümmend viis tuhat kaheksasada seitse
- **int x;**

# Mille toob järgmine programmi lõik ekraanile?

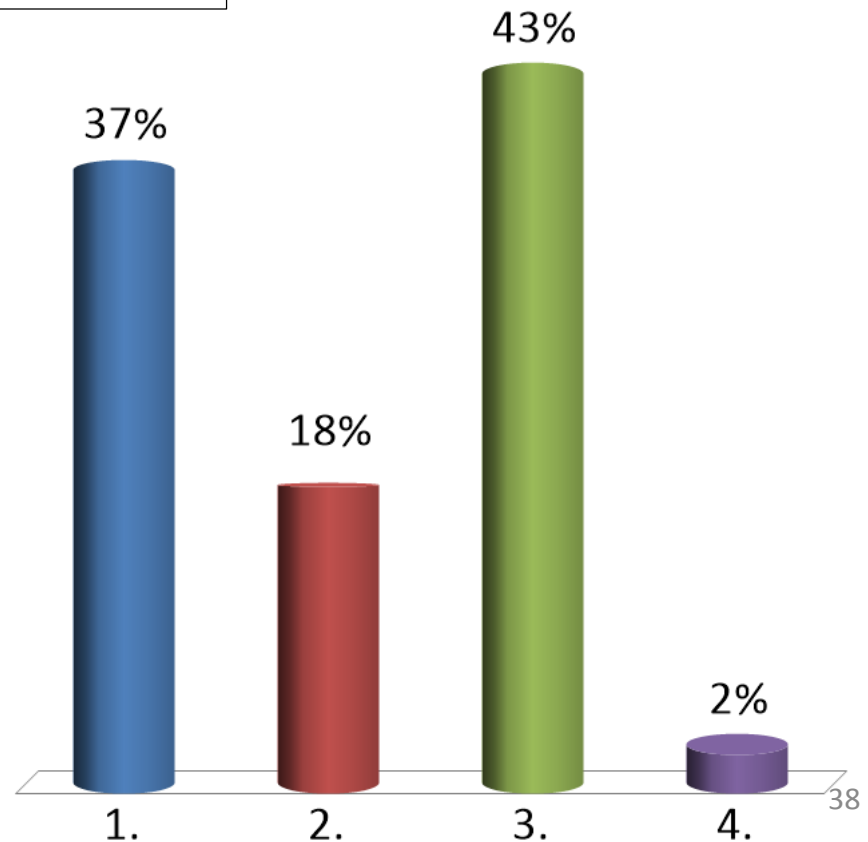
```
int n = 1000000;  
System.out.println(n*n);
```

1. 1000000000000000

✓ 2. -727379968

3. veateate

4. midagi muud



# Ujukomaarvud

- IEEE 754
  - **float**
    - 32-bitiline
  - **double**
    - 64-bitiline
- **double raadius = 7.8;**

# Tõeväärtus

- `boolean`
  - `true`
  - `false`
- `boolean tv = true;`



# Sümbolid

- **char**
  - 16-bitiline *Unicode*-sümbol
  - '\u0000' (0) kuni '\uffff' (65535)
- `char ch1 = 'a';`
- `char ch2 = 65;`
- `System.out.print("ch1 ja ch2: ");`
- `System.out.println(ch1 + " " + ch2);`

# Nimi

- on tõstutundlik
- võib sisaldada tähti, numbreid, `_`, `$`
- ei tohi alata numbriga
- ei tohi olla Java võtmesõna
- ei tohi olla **true, false, null**

# Kirjeldamine, väärtustamine

```
char a = 'e';  
double raadius;  
int x, y, z, k;  
raadius = 3.4;  
x = y = z = 1;  
int i = 1, j = 2;  
k = 1 + 5;  
x = y + z;  
long kood = 48007140350L;
```

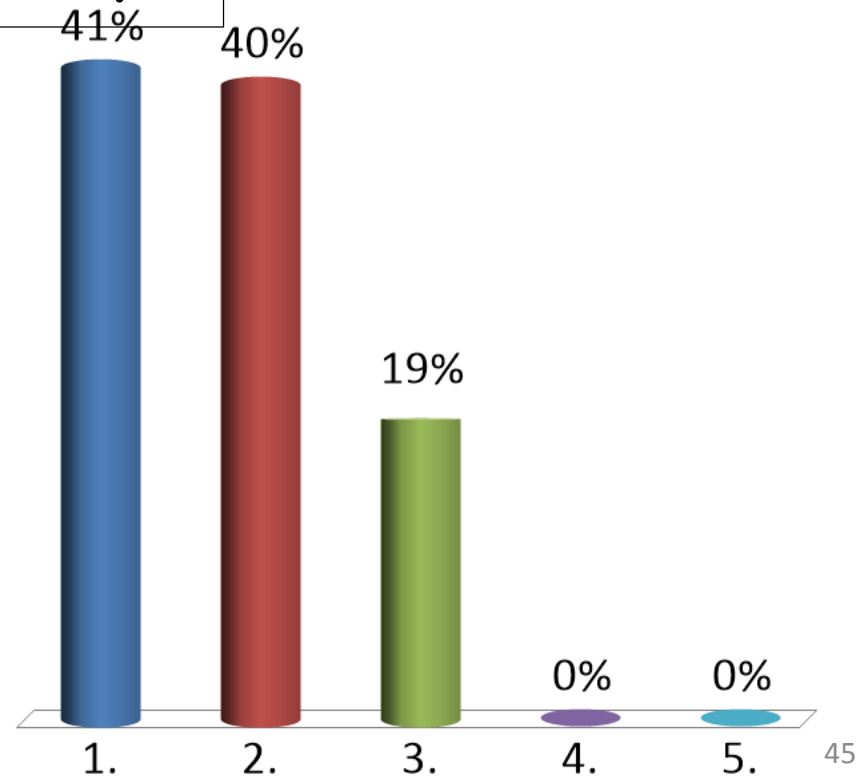
# Ühiksuurendamine, ühikvähendamine

<b>++muutuja</b>	Muutujat suurendatakse ühe võrra ja seejärel arvutatakse avaldise väärtus kasutades muutuja uut väärtust
<b>muutuja++</b>	Avaldise väärtus arvutatakse kasutades muutuja vana väärtust ja seejärel suurendatakse muutujat ühe võrra
<b>--muutuja</b>	Muutujat vähendatakse ühe võrra ja seejärel arvutatakse avaldise väärtus kasutades muutuja uut väärtust
<b>muutuja--</b>	Avaldise väärtus arvutatakse kasutades muutuja vana väärtust ja seejärel vähendatakse muutujat ühe võrra

# Mille toob järgmine programmi lõik ekraanile?

```
int i = 5;  
int uus = 5 * i++;  
System.out.println(uus);
```

- ✓ 1. 25
- 2. 26
- 3. 30
- 4. veateate
- 5. midagi muud



# Mille toob järgmine programmi lõik ekraanile?

```
int i = 5;  
int uus = 5 * ++i;  
System.out.println(uus);
```

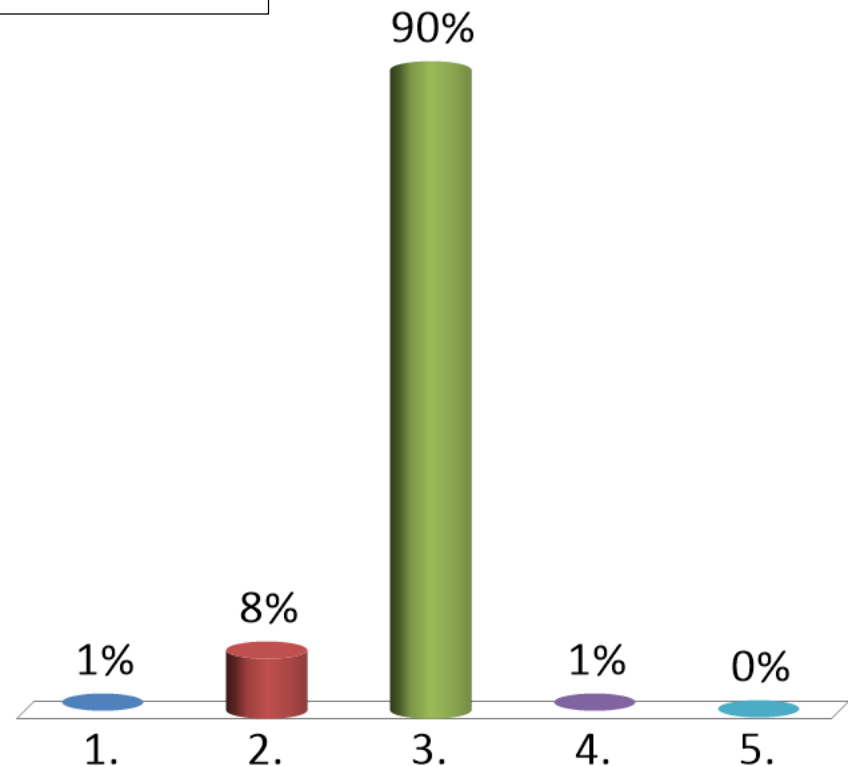
1. 25

2. 26

✓ 3. 30

4. veateate

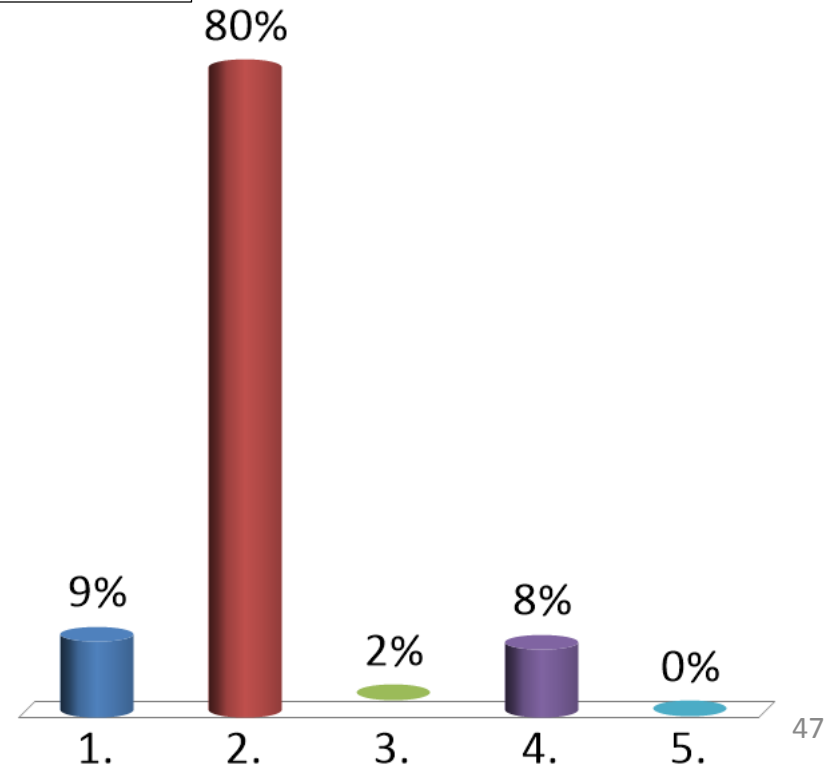
5. midagi muud



# Mille toob järgmine programmi lõik ekraanile?

```
int i = 5;  
int uus = (5 * i)++;  
System.out.println(uus);
```

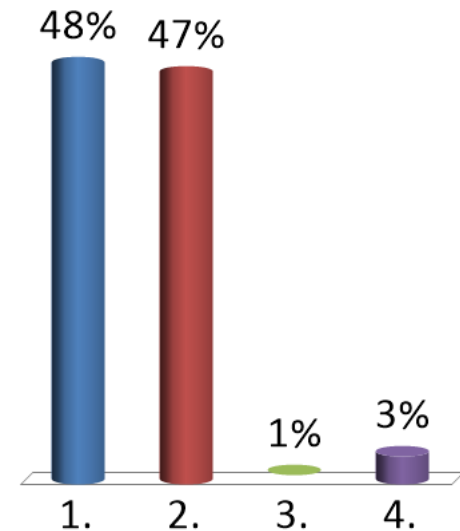
1. 25
2. 26
3. 30
- ✓ 4. veateate
5. midagi muud



# Mis ilmub ekraanile?

```
int d = 0;  
int c = 0;  
d = c++;  
System.out.println(c);
```

1. 0
- ✓ 2. 1
3. midagi muud
4. veateade

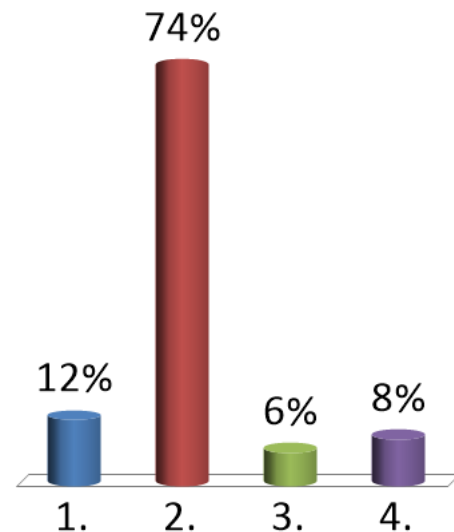




# Mis ilmub ekraanile?

```
int c = 0;  
c = c++;  
System.out.println(c);
```

- ✓ 1. 0
- 2. 1
- 3. midagi muud
- 4. veateade



C = C++

C = C

C++

```
int temp = c; // temp = 0
```

```
c++; // c = 1
```

```
c = temp; // c = 0
```

# javap -c Liitmine

Compiled from "Liitmine.java"

```
public class Liitmine {
```

```
  public Liitmine();
```

```
    Code:
```

```
      0: aload_0
```

```
      1: invokespecial #1          // Method java/lang/Object."<init>": ()V
```

```
      4: return
```

```
public static void main(java.lang.String[]);
```

```
  Code:
```

```
    0: iconst_0
```

```
    1: istore_1
```

```
    2: iload_1
```

```
    3: iinc      1, 1
```

```
    6: istore_1
```

```
    7: getstatic #2 //Field java/lang/System.out:Ljava/io/PrintStream;
```

```
   10: iload_1
```

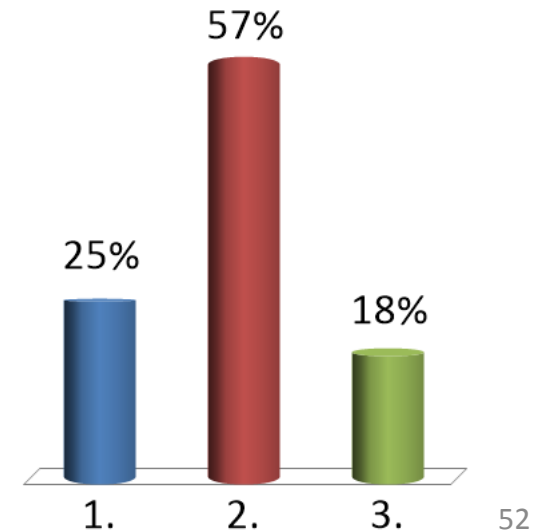
```
   11: invokevirtual #4 //Method java/io/PrintStream.println:(I)V
```

```
   14: return
```

```
}
```

Kas on lubatud  
`int Public = 50000;`

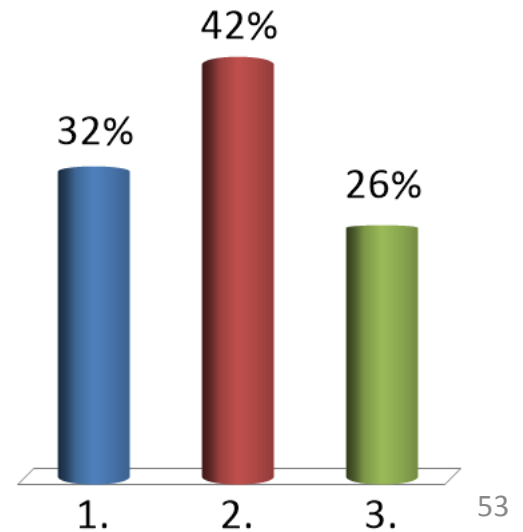
- ✓ 1. Jah
- 2. Ei
- 3. Ei oska öelda



Kas on lubatud

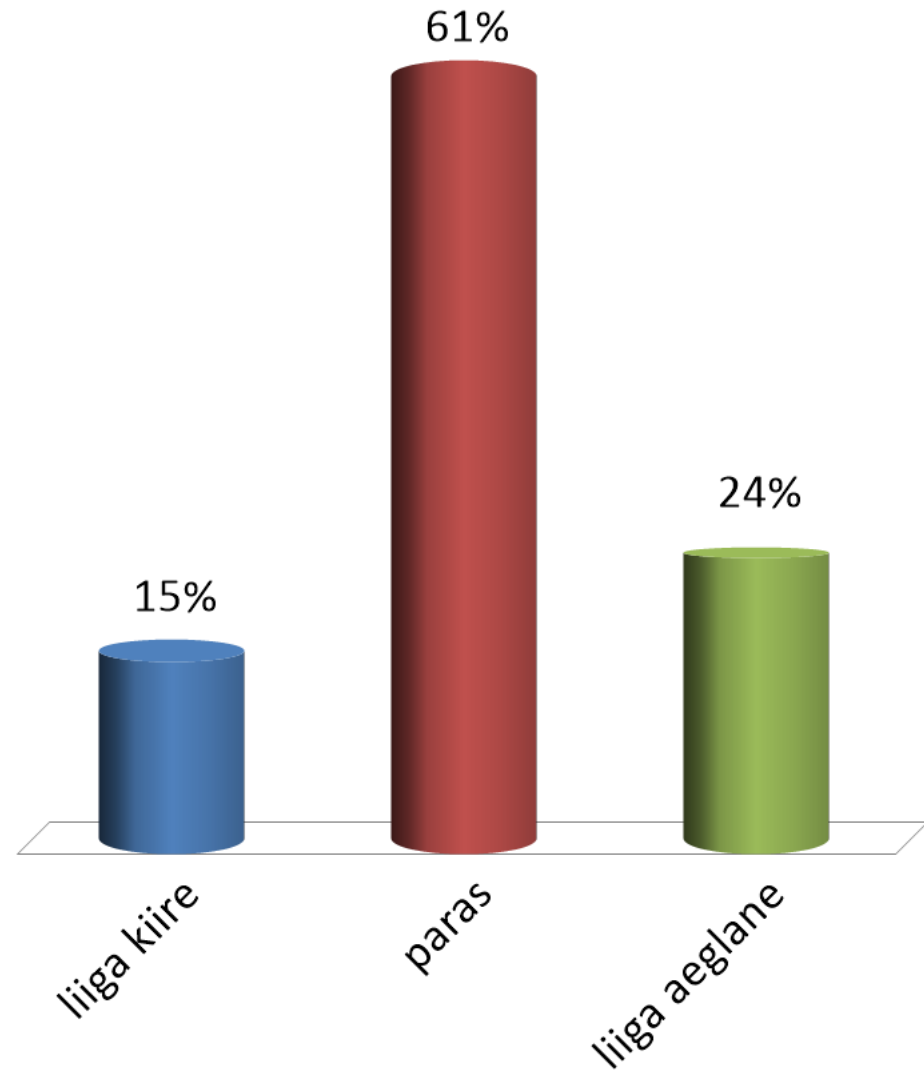
```
int x = 5_000_000;
```

- ✓ 1. Jah
- 2. Ei
- 3. Ei oska öelda



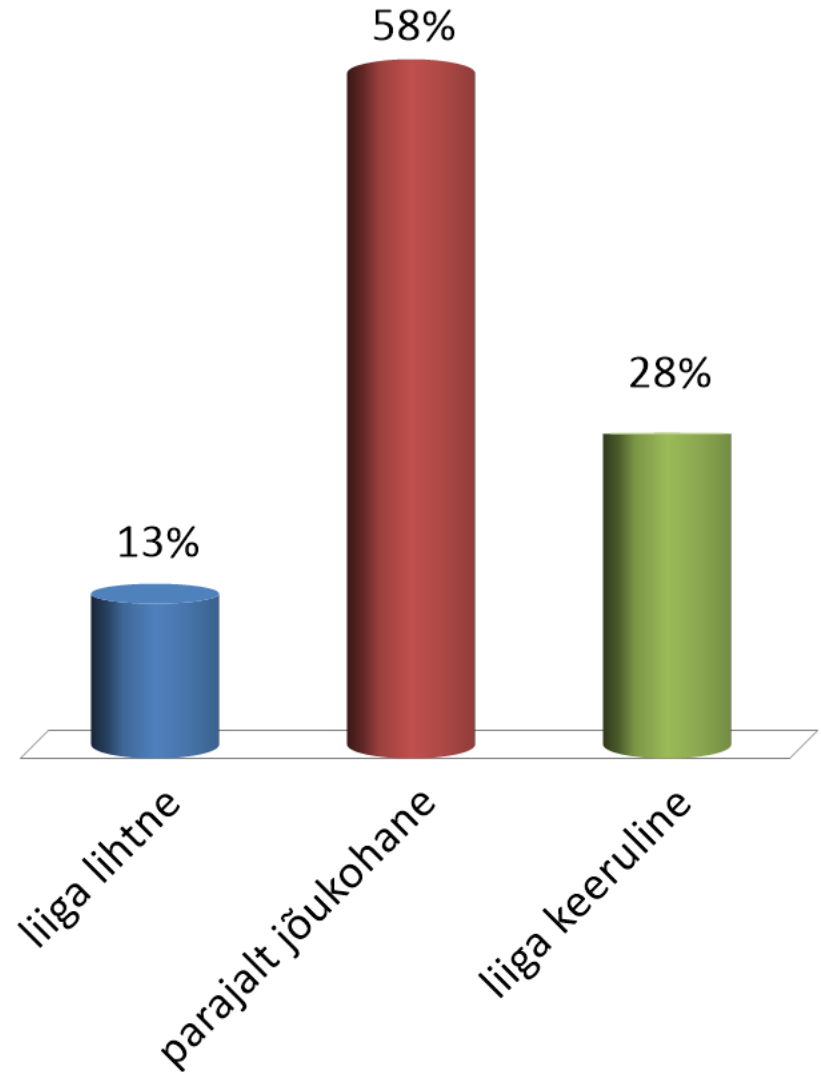
# Loengu tempo oli

1. liiga kiire
2. paras
3. liiga aeglane



# Materjal tundus

1. liiga lihtne
2. parajalt jõukohane
3. liiga keeruline



Suur tänu osalemast ja  
kohtumiseni!