# Objektorientētā programmēšana Java valodā

## polimorfisms un abstrakcija

Ainārs Skrubis 2PT

2025. gada jūnijs

# 1. TEORIJA

**1.1.** Objektorientētā programmēšana (OOP) OOP ir programmēšanas paradigma, kuras pamatā ir objekti. Galvenie OOP principi:

- Mantošana (Inheritance)
- Iekapsulēšana (Encapsulation)
- **Abstrakcija (Abstraction)**
- **Polimorfisms (Polymorphism)**

**1.2. Polimorfisms(Polymorphism)** ir koncepts kas nodrošina iespēju vienu darbību veikt dažādos veidos.

**1.2.1.** Polimorfisma galvenās iezīmes:

- **Vairākas uzvedības:** Viena un tā pati metode var uzvesties atšķirīgi atkarībā no objekta, kas to izsauc.
- **Metodes ignorēšana(Overriding):** Bērnu klase var pārdefinēt savas vecākklases metodi.
- **Metodes pārslodze(Overloading):** Var definēt vairākas metodes ar vienādu nosaukumu, bet atšķirīgiem parametriem.
- **Izpildes laika lēmums:** Izpildes laikā Java nosaka, kuru metodi izsaukt, atkarībā no objekta faktiskās klases.

**1.3. Abstrakcija (Abstraction)** ir process, kas progrmmas lietotājam parāda tikai tam nepieciešamo informāciju un slēpj datus, kas tam nav nepieciešamo. Galvenais abstrakcijas mērķis ir datu slēpšana.

**1.3.1.** Abstrakcijas galvenās iezīmes:

- Abstrakcija slēpj sarežģītas detaļas un parāda tikai būtiskās iezīmes.
- Abstraktām klasēm var būt metodes bez ieviešanas, un tās jāievieš apakšklasēs.
- Abstrahējot funkcionalitāti, izmaiņas ieviešanā neietekmē kodu, kas ir atkarīgs no abstrakcijas.

### 1.3.2. Saskarnes(Interface)

**Saskarnes** ir vēl viena abstrakcijas ieviešanas metode Java valodā. Galvenā atšķirība ir tā, ka, izmantojot saskarnes, Java klasēs var panākt **100% abstrakciju**. Java vai jebkurā citā valodā saskarnes ietver gan metodes, gan mainīgos, bet tām trūkst metodes pamatteksta. Papildus abstrakcijai saskarnes var izmantot arī mantojuma ieviešanai Java valodā.

## 2. KODA PIEMĒRI

**2.1.** Apstrakcijas piemērs

```java
//abstraktā klase ar abstraktām metodēm
abstract class Cilveks {
 abstract void ieslekt();
 abstract void izslekt();
}
class Pults extends Cilveks {
 @Override
 void ieslekt() {
     System.out.println("TV ir ieslekts.");
 }
 @Override
 void izslekt() {
     System.out.println("TV ir islekts.");
 }
}
//Main klase
public class Abstrakcija {
 public static void main(String[] args) {
     Cilveks remote = new Pults();
     remote.ieslekt();
     remote.izslekt();
 }
}
```

```
TV ir ieslekts.
TV ir islekts.
```

**2.2.** Polimorfisma piemērs

```java
class Virietis{
 void disk() {
     System.out.println("Es esmu vīrietis.");
 }
}

//Klase kas pārakstīs Cilveks klases metodi
class Tevs extends Virietis {

 // Pāraksta metodi (Overriding)
 @Override
 void disk() {
     System.out.println("Es esmu tēvs.");
 }
}

public class Polimorfisisms {
 public static void main(String[] args) {
     Virietis p = new Tevs();
     p.disk();
 }
}
```

```
Es esmu tēvs.
```

**2.3.** Saskarnes(Interface) piemērs

```java
//Definē interface vārdu Forma
interface Forma {
 double aprLauk(); //Abstraktā metode apreikina laukumu
}
//Izmanto mantošanu klasē Aplis
class Aplis implements Forma {
 private double r;    // rādius
 public Aplis(double r) {
   this.r = r;
 }
 //Implementē abstrakto metodi
 public double aprLauk()
 {
     return Math.PI * r * r;
 }
}
//Implementē interface clasē taisnstūris
class Taisnsturis implements Forma {
 private double garums;
 private double platums;
 public Taisnsturis(double garums, double platums)
 {
     this.garums = garums;
     this.platums = platums;
 }
 //Implementē apstrakto metodi
 public double aprLauk() {
   return garums * platums;
 }
}
//Main klase
public class Interface {
 public static void main(String[] args)
 {
     //Izveido apļa un taisnstūra objektus
     Aplis a = new Aplis(5.0);
     Taisnsturis t = new Taisnsturis(4.0, 6.0);
     System.out.println("Apļa laukums: "+ a.aprLauk());
     System.out.println("Taisnstūra laukums: "+ t.aprLauk());
 }
}
```

```
Apļa laukums: 78.53981633974483
Taisnstūra laukums: 24.0
```

# 3. Izmantotie avoti

- GeeksForGeeks (https://www.geeksforgeeks.org/java/polymorphism-in-java/)
- GeeksForGeeks (https://www.geeksforgeeks.org/java/abstraction-in-java-2/)
- W3School (https://www.w3schools.com/java/java_polymorphism.asp)
- W3School(https://www.w3schools.com/java/java_abstract.asp)
- ChatGPT
- Skolo.lv (Rāvalds java programmēšanas kursi)