

Objektorientētā programmēšana Java valodā

Superklase un mantošana

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. Superklase un apakšklase Superklase ir klase, no kuras manto citas klases. Apakšklase jeb subklase izmanto "extends" atslēgvārdu, lai mantotu no superklases.

Piemērs:

```
class Animal {  
    void makeSound() {  
        System.out.println("Sound");  
    }  
}  
  
class Dog extends Animal {  
    void makeSound() {  
        System.out.println("Bark");  
    }  
}
```

1.3. Atslēgvārdi "extends" un "super"

- extends: Norāda, ka klase manto no citas
- super: Izsauc superklases metodi vai konstruktoru

1.4. Metodes pārrakstīšana (Overriding) Apakšklase var definēt to pašu metodi ar tādu pašu parakstu kā superklase, bet ar citu uzvedību.

2. KODA PIEMĒRI

2.1. Vienkārša mantošana

```
class Vehicle {  
    void move() {  
        System.out.println("Moving");  
    }  
}  
  
class Car extends Vehicle {  
    void move() {  
        System.out.println("Driving");  
    }  
}
```

2.2. Konstruktoru izmantošana ar super

```
class Person {  
    String name;  
    Person(String name) {  
        this.name = name;  
    }  
}  
  
class Student extends Person {  
    int grade;  
    Student(String name, int grade) {  
        super(name);  
        this.grade = grade;  
    }  
}
```

3. Izmantotie avoti

- GeeksForGeeks (<https://www.geeksforgeeks.org/inheritance-in-java/>)
- W3School (https://www.w3schools.com/java/java_inheritance.asp)
- ChatGPT
- Skolo.lv (Rāvalds java programmēšanas kursi)