

სემინარი 5. ინტერფეისები და მემკვიდრეობა

სავარჯიშო 1

- შექმენით Device ინტერფეისი on და off მეთოდებით.
- შექმენით Device ინტერფეისის მემკვიდრე Electronic ინტერფეისი plugin() მეთოდით.
- შექმენით Phone აბსტრაქტული კლასი აბსტრაქტული makeCall(String number) და არააბსტრაქტული testCall(String number) მეთოდებით.
- Phone კლასი უნდა აკეთებდეს Electronic ინტერფეისის იმპლემენტაციას.
- შექმენით Phone კლასის მემკვიდრე Mobile კლასი და აღწერეთ მხოლოდ ერთი, ნებისმიერი მეთოდი.
- გადააკეთეთ Electronic ინტერფეისი აბსტრაქტულ კლასად.

სავარჯიშო 2

- აღწერეთ Parent მშობელი და Child შვილობილი კლასები.
- აღწერეთ go მეთოდი, რომელიც აბრუნებს void-ს და ბეჭდავს კონსოლში “go method in Parent class”
- გადაფარეთ go მეთოდი Child კლასში, რომელიც დაბეჭდავს “go method in Child class”.
- აღწერეთ Test კლასი
- აღწერეთ ორი სტატიკური invoke მეთოდი Test კლასში; პირველს პარამეტრად ჰქონდეს Parent ხოლო მეორეს Child ტიპი.
- ორივე invoke მეთოდში გამოიძახეთ go მეთოდი გადმოცემული პარამეტრიდან
- აღწერეთ Test კლასში main მეთოდი და შექმენით:
 - Parent კლასის ობიექტი Parent კლასის ტიპის მითითებით
 - Child კლასის ობიექტი Child კლასის ტიპის მითითებით
 - Parent კლასის ობიექტი Child კლასის ტიპის მითითებით
 - Child კლასის ობიექტი Parent კლასის ტიპის მითითებით
 - გადაეცით ყველა მითითება invoke მეთოდს ცალცალკე და დააკვირდით რა დაიბეჭდება.

სავარჯიშო 1

```
interface Device {
    void on();
    void off();
}

interface Electronic extends Device {
    void plugin();
}

abstract class Phone implements Electronic {
    abstract void makeCall(String number);
    boolean testCall(String number){
        return true;
    }
}

public class Mobile extends Phone {

    @Override
    void makeCall(String number) {
        System.out.println("Making call " + number);
    }

    @Override
    public void plugin() {

    }

    @Override
    public void on() {

    }

    @Override
    public void off() {

    }

    public static void main(String[] args) {
        Mobile mobile = new Mobile();
    }
}
```

```

        String number = "123456";
        if (mobile.testCall(number)) {
            mobile.makeCall(number);
        }
    }
}

```

სავარჯიშო 2

```

class Parent {
    public void go(){
        System.out.println("go method in Parent class");
    }
}

class Child extends Parent {
    @Override
    public void go(){
        System.out.println("go method in Child class");
    }
}

public class Test {
    static void invoke(Parent parent){
        parent.go();
    }

    static void invoke(Child child){
        child.go();
    }

    public static void main(String[] args) {
        Parent parent = new Parent();
        Child child = new Child();
        Parent parentRefChildObj = new Child();

        invoke(parent);
        invoke(child);
        invoke(parentRefChildObj);
    }
}

```