



Principet e Avancuara të Gjuheve të Programimit

Lamir Shkurti, Can. PhD.
lamir.shkurti@gmail.com

INTERFACE



- Është e ngjajshme me klasën.
- Është një koleksion i metodave abstrakte.
- Një klasë zbaton – “implementon” një interface, duke trashëguar kështu metodat abstrakte të interface-it.
- Një Interface mund të përmbajë gjithashtu edhe konstante, metoda default, metoda statike etj.
- Metodat në një interface mund të kenë implementim (body) vetëm në rast që janë metoda default ose metoda statike.

INTERFACE



- Shkrimi i një Interface është i ngjajshëm me shkrimin e një klase.
- Një klasë përshkruan atributet dhe sjelljet e një objekti, ndërsa një Interface përmban sjellje që një klasë implementon.
- Në rast që klasa që implementon një Interface nuk është abstrakte, atëherë duhet të implementohen të gjitha metodat e Interface-it në klasë.

INTERFACE



Një Interface është e ngjashme me një klasë në mënyrat e mëposhtme:

- Një Interface mund të përmbajë metoda.
- Një Interface shkruhet në një file me extension.java, me emrin e Interface-it që përputhet me emrin e file-it.
- Pasi që të bëhet compile file i Interface-it, krijohet një file me extension.class.

INTERFACE



Një Interface është e ndryshme nga një klasë në disa mënyra, duke përfshirë:

- Ju nuk mund të krijoni (inicializoni) një Interface. Një Interface nuk përmban konstruktor.
- Të gjitha metodat në një Interface janë abstrakte.
- Një Interface nuk mund të përmbajë attribute të instancës. Atributet e vetme që mund të shfaqen në një interface duhet të deklarohen si statike apo final (përfundimtare).
- Një Interface nuk bëhet extends nga një klasë; por implementohet (implements) nga një klasë.
- Një Interface mund të bëjë extend shumë Interface-a.

DEKLARIMI I INTERFACE



Fjala e cila përdorët për të deklaruar një interface është fjala **interface**.

Shembull se si deklarohet një interface:

```
public interface EmriInterface {  
    // Çdo atribut final apo static  
    // Çdo metodë abstrakte  
}
```

DEKLARIMI I INTERFACE



Interface-at kanë këto veti -

- Një interface është në mënyrë implicite abstrakte. Ju nuk keni nevojë të përdorni fjalën abstrakte ndërsa deklaroni një interface.
- Çdo metodë në një interface është gjithashtu abstrakte, kështu që fjala kyçe abstrakte nuk është e nevojshme.

Shembull:

```
interface Animal {  
  
    public void eat();  
    public void travel ();  
  
}
```

IMPLEMENTIMI I INTERFACE

```
interface Animal {  
    public void eat();  
    public void travel();  
  
    default void metoda () {  
        System.out.println("Test metoda me body");  
    }  
    static void metodastatike () {  
        System.out.println("Test metoda statike me body ");  
    }  
}
```


SHEMBULL



```
public class Dog implements Animal {  
    public void eat() {  
        System.out.println("Dog eats");  
    }  
    public void travel () {  
        System.out.println("Dog travels");  
    }  
    public int noOfLegs () {  
        return 4;  
    }  
    public static void main(String args[]) {  
        Dog d = new Dog();  
        d.eat();  
        d.travel();  
        d.metoda();  
        Animal.metodastatike();  
    }  
}
```

IMPLEMENTIMI I INTERFACE



Janë disa rregulla kur dëshirojmë të implmentojmë interface-in:

- Një klasë mund të implementojë më shumë se një interface në të njëjtën kohë.
- Një klasë mund të bëjë extend vetëm një klasë, por mund të implementojë shumë interface.
- Një interface mund të bëjë extend një interface tjetër, në një mënyrë të ngjashme si kur një klasë bën extend një klasë tjetër.

INTERFACE - EXTEND



- Një Interface mund të bëjë extend një interface tjetër në të njëjtën mënyrë që një klasë mund të bëjë extend
- një klasë tjetër.

Shembull

```
public interface Sports {  
    public void setHomeTeam(String name);  
    public void setVisitingTeam(String name);  
}  
public interface Football extends Sports {  
    public void homeTeamScored(int points);  
    public void visitingTeamScored(int points);  
    public void endOfQuarter(int quarter);  
}
```

```
public interface Hockey extends Sports {  
    public void homeGoalScored();  
    public void visitingGoalScored();  
    public void endOfPeriod(int period);  
    public void overtime Period(int ot);  
}
```

Interface Hockey ka katër metoda, por trashëgon dy nga Sportet; kështu, një klasë që implementon Hokej duhet të zbatojë të gjashtë metodat. Në mënyrë të ngjashme, një klasë që implementon Futbollin duhet të përcaktojë tre metodat nga Futbolli dhe dy metodat nga Sportet.

INTERFACE TË SHUMËFISHTË - EXTENDS



- Një klasë mund të bëjë extend vetëm një klasë prind.
- Trashëgimia e shumëfishtë nuk lejohet tek klasat.
- Interface-at nuk janë klasa dhe një interface mund të bëjë extend më shumë se një interface prindërore.
- Fjala extends përdoret një herë dhe interface-at prindërore deklarohen në një listë të ndarë me presje.

Shembull:

```
public interface Hockey extends Sports, Event
```

