Java 8 Interface yangiliklari

PERSONAL
DEVELOPMENT PROCESS
stay weird, stay different

Reja

- ✓ **Default method**
- ✓ **Static method**
- ✓ **Functional interface**

Default methods

- **default** kalit so’zi bilan e’lon qilinadi.
- Body qismi mavjud bo’ladi;
- Impliment qilingan class ga voris bo’lib o’tadi;
- Impliment qilgan classdan olingan object orqali murojaat qilinadi;
- Override qilish mumkin (majburiy emas)
- Ikkita aynan (nomi va argumentlari) bir xil default method ga ega bolgan interfacelarni bitta classga implement qilinganda ushbu default method override qilinishi shart

Default method

```
interface MyInterface {

    void abstractMethod();

    default void defaultMethod() {
        System.out.println("Default method is called");
    }
}
```

Default method nima uchun kerak?

(Interface) Static method

- **static** kalit so’zi bilan e’lon qilinadi.
- Faqat Interfacega tegishli bo’ladi;
- Impliment qilingan class ga voris bo’lib o’tmaydi;
- Impliment qilgan classdan olingan object orqali murojaat qilib bo’lmaydi;
- Override qilinmaydi;

Static methods

```
interface MyInterface {

    void abstractMethod();

    default void defaultMethod() {
        System.out.println("Default method is called");
    }

    static void staticMethod(){
        System.out.println("Static method is called");
    }
}
```

Functional interface

- ✓Bittadan ko’p bo’lmagan abstract methodga ega bo’maydi (Single Abstract Method SAM)
- ✓default method ga ega bo’lishi mumkin;
- ✓static method ga ega bo’lishi mumkin;
- ✓(public static final) o’zgaruvchilarga ega bo’lishi mumkin;
- ✓@FunctionalInterface annotasiyasi bilan belgilab qo’yiladi.

Functional interface

```
interface MyInterface {
    double G = 9.8;

    void abstractMethod();

    default void defaultMethod() {
        System.out.println("Default method is called");
    }

    static void staticMethod() {
        System.out.println("Static method is called");
    }
}
```

Functional interface ni (amalga oshirish) implement qilish:

Class orqali;

Anonim class orqali;

Lambda Expression orqali

Functional Interface ni class orqali (amalga oshirish) implement qilish

```
public class Main {
    public static void main(String[] args) {
        Addition addition = new Addition();// Interfacedan voris olgan aniq klass
        int a = addition.calculate(4, 5);
    }
}
interface Operationable {
    int calculate(int x, int y);
}
class Addition implements Operationable {
    @Override
    public int calculate(int x, int y) {
        return x + y;
    }
}
```

Functional Interface ni anonim class orqali amalga oshirish qilish

```
public class Main {
    public static void main(String[] args) {
        Operationable addition = new Operationable() {
            //Anonim class body qismi
            @Override
            public int calculate(int x, int y) {
                return x + y;
            }
        };
        int a=addition.calculate(4,5);
    }
}

interface Operationable {
    int calculate(int x, int y);
}
```

Functional Interface ni Lambda Expression orqali amalga oshirish qilish

```
public class Main {
    public static void main(String[] args) {
        Operationable addition=(a,b)->a+b;
        int c=addition.calculate(4,5);
    }
}

interface Operationable {
    int calculate(int x, int y);
}
```

E’tiboringiz uchun raxmat

PERSONAL
DEVELOPMENT PROCESS
stay weird, stay different