

独習Java

第7章 7.3～7.4

7.3 インターフェイスの継承

7.4 instanceof演算子



インターフェイスの継承

- インターフェイス間の継承関係を定義する場合にも extends キーワードを使用する。
- インターフェイスは複数のインターフェイスを直接拡張できる。

- 宣言例

```
intfModifier interface intfName extends intfList {  
//インターフェイス本体  
}
```

*interfListはカンマで区切って指定する



インターフェイス継承例(1/2)

```
interface J {  
    int j = 200;  
    int j1();  
}
```

```
interface K {  
    double k1();  
}
```

```
interface L extends J, K {  
    boolean l1();  
}
```

```
class I implements L {  
    return 4;  
}
```

- public double k1() {
- return 6.8;
- }
- public boolean l1() {
- return true;
- }
- }
- class InterfaceInheritance {
- public static void main(String args[]) {
- I i = new I();
- System.out.println(i.j);
- System.out.println(i.j1());



インターフェイスの継承例(2/2)

```
System.out.println(i.k1());  
System.out.println(i.l1());  
}  
}
```

出力結果

200

4

6.8

true



instanceof演算子とは？

- オブジェクトクラスやオブジェクトに実装されているインターフェイスを確認するために使う演算子。
- 宣言例

`varName instanceof type`



incetanceof演算子使用例(1/4)

```
abstract class Fish {
    abstract void display();
}
abstract class FreshWaterFish extends Fish {
}
abstract class SaltWaterFish extends Fish {
}
class Nizimasu extends FreshWaterFish {
    void display() {
        System.out.println("Nizimasu");
    }
}
```



incetanceof演算子使用例(2/4)

```
class Same extends SaltWaterFish {  
    void display() {  
        System.out.println("Same");  
    }  
}  
  
class Kingyo extends FreshWaterFish{  
    void display() {  
        System.out.println("Kingyo");  
    }  
}
```



instanceof演算子使用例(3/4)

```
class InstanceofClass {  
    public final static int NUMFISH = 4;  
    public static void main(String args[]) {  
        //魚類の配列を作成する  
        Fish fishes[] = new Fish[NUMFISH]  
  
        fishes[0] = new Nizimasu();  
        fishes[1] = new Same();  
        fishes[2] = new Kingyo();  
        fishes[3] = new Nizimasu();  
    }  
}
```



instanceof演算子使用例(4/4)

```
//instanceof演算子を利用する
for (int i = 0; i < NUMFISH; i++) {
    Fish fish = fishes[i];
    if (fish instanceof SaltWaterFish) {
        fish.display();
    }
}
}
```

出力

Same



宿題

- 以下のプログラムの問題点を挙げよ。

```
interface Nom1{
```

```
    int nom1 = 0;
```

```
    int special = 100;
```

```
}
```

```
interface Nom2 extends Nom1 {
```

```
    int Nom2 = 1;
```

```
    int special = 100;
```

```
}
```

```
interface Nom3 extends Nom1 {
```

```
    int Nom3 = 2;
```

```
    float special = 100;
```

```
}
```



宿題（続き）

```
interface Sum extends Nom2 , Nom3 {  
    int Sum = 3;  
}
```

```
class Z implements Sum {  
}
```

```
class AllInterface {  
    public static void main(String args[]) {  
        Z z = new Z();  
        System.out.println(z.Nom1);  
        System.out.println(z.special);  
    }  
}
```