

# Java プログラミング入門

— 継承・オーバーライド —

Waseda University

# 継承

## 継承

共通した処理をスーパークラス (親クラス) としてまとめ、サブクラス (子クラス) にコードの再利用や機能の追加を行うための技法

- スーパークラス (親クラス) の **フィールド**、**メソッド** を引き継ぐことができる
- コンストラクタは継承されない

## 継承の文法

```
修飾子 class サブクラス名 extends スーパークラス名 {  
    クラスの本体  
}
```

# スーパークラス (親クラス) の例

## Person クラス

“名前”、“身長”、“体重”、“年齢”を管理し、すべてのデータを表示するメソッドを持つ Person クラスを定義せよ。

## Person.java

```
public class Person {
    String name;
    double height, weight;
    int age;
    public Person(){
        :
    }
    public Person(String s, double h, double w, int a){
        :
    }
    public void print(){
        :
    }
}
```

# スーパークラス (親クラス) の例

## Person.java

```
public class Person {
    String name;
    double height, weight;
    int age;
    public Person(){
        name = ""; height = 0.0; weight = 0.0; age = 0;
    }
    public Person(String s, double h, double w, int a){
        name = s; height = h; weight = w; age = a;
    }
    public void print(){
        System.out.println("name : " + name);
        System.out.println("height : " + height + " cm");
        System.out.println("weight : " + weight + " kg");
        System.out.println("age : " + age + " years old");
    }
}
```

# Person クラスを継承

## Student.java

```
public class Student extends Person{
    String id;
    public static void main(String[] args){
        Student st1 = new Student();
        st1.name = "Frank";
        st1.height = 175.0;
        st1.weight = 63.0;
        st1.age     = 18;
        st1.id = "1w00B000";
        st1.print();
    }
}
```

## 実行例

```
name : Frank
height : 175.0 cm
weight : 63.0 kg
age : 18 years old
```

## 継承された個所

### Person.java

```
public class Person {
    String name;
    double height, weight;
    int age;
    public Person(){
        name = ""; height = 0.0; weight = 0.0; age = 0;
    }
    public Person(String s, double h, double w, int a){
        name = s; height = h; weight = w; age = a;
    }
    public void print(){
        System.out.println("name : " + name);
        System.out.println("height : " + height + " cm");
        System.out.println("weight : " + weight + " kg");
        System.out.println("age : " + age + " years old");
    }
}
```

# オーバーライドと super 参照

## print メソッドの拡張

Person クラスの print メソッドは名前、身長、体重、年齢を表示させる。  
Student クラスの print メソッドは名前、身長、体重、年齢、ID を表示させたい

- メソッドの書き換え ⇒ **オーバーライド**
- Person クラスの print メソッドの参照 ⇒ **super** を使った参照

## メソッドのオーバーライド

サブクラスのメソッドと同じ名前、同じ引数を持つメソッドをスーパークラスで定義し、**書き換える機能**

## super

サブクラスがスーパークラスのインスタンスのフィールド、メソッドを参照するとき用いる

# オーバーライドの例

Student.java

```
public class Student extends Person{
    String id;
    @Override
    public void print(){
        super.print();
        System.out.println("id : "+ id );
    }
    public static void main(String[] args){
        Student st1 = new Student();
        st1.name = "Frank";
        st1.height = 175.0;
        st1.weight = 63.0;
        st1.age     = 18;
        st1.id = "1w00B000";
        st1.print();
    }
}
```

# コンストラクタの例

## Student.java

```
public class Student extends Person{
    String id;

    public Student(){
        super();
    }
    public Student(String n, double h, double w, int a){
        super(n, h, w,a);
    }
    public Student(String n, double h, double w, int a, String sID){
        super(n, h, w,a);
        id =sID;
    }
    @Override
    public void print(){
        super.print();
        System.out.println("id : "+ id );
    }
}
```

# コンストラクタの例

## SampleStudent.java

```
public class SampleStudent {
    public static void main(String[] args){
        Student st1 = new Student("Frank", 175.0, 63.5,18);
        st1.id      = "1w00B000";
        st1.print();

        Student st2 = new Student("Thomas", 177.0, 72.0,18,"1w00C000");
        st2.print();
    }
}
```

# super でコンストラクタを参照する場合の注意点

## コンストラクタでの注意点

サブクラスでスーパークラスのコンストラクタを参照する場合  
必ず一行目に書かなければならない。

```
public Student(String n, double h, double w, int a, String sID){  
    super(n, h, w,a);  
    id =sID;  
}
```