

オブジェクト指向プログラミング 第4回

コンストラクタ オーバーロードとオーバーライド

担当：高橋、佐藤聖也

1

1.1

数学で、一般に x の関数を

$$f(x) \quad [\text{数学}] \quad (1)$$

のように書いたことを思い出しましょう．引数 (argument) x に対する関数 $f(x)$ の返り値が y のときは

$$y = f(x) \quad [\text{数学}] \quad (2)$$

のように書きました．さて、 x の値が与えられたとき y の値が具体的に決まるためには、例えば、

$$y = 3x + 1 \quad [\text{数学}] \quad (3)$$

などとしなければなりません．まずはこの (3) を以下のようなプログラムに直すところから始めましょう．

FunctionMain01.java

```
class FunctionMain01{
    public static void main(String [] args){
        double x,y;
        x = 3.0;
        y = 3.0 * x + 1.0;
        System.out.println("f("+x+") = "+y);
    }
}
```

FunctionMain01.java の実行結果

f(3.0) = 10.0

次に、FunctionMain01.java を書き換えて

$$\begin{array}{lcl} f(x) & = & 3x + 1 \\ y & = & f(x) \end{array} \quad [\text{数学}] \quad (4)$$

のような構造を持たせたプログラム FunctionMain02.java を作成しましょう．すなわち、 $f(x) = 3x + 1$ の部分をメソッドとして独立させてみます．

練習問題

FunctionMain02.java は未完成で、そのままではコンパイルできない．関数 $f(x)$ がインスタンスメソッドとなるよう FunctionMain02.java にコードを加筆し、FunctionMain01.java と同様の出力を得られるようにせよ．

FunctionMain02.java (未完成)

```
class FunctionMain02{
    double f(double x){
        return 3.0 * x + 1.0;
    }
    public static void main(String [] args){
        double x,y;
        x = 3.0;
        y = f(x);
        System.out.println("f("+x+") = "+y);
    }
}
```

1 次関数の係数 (パラメータ) を a, b とし , 一般形を $ax + b$ で表した上で $a = 3, b = 1$ を代入する構造

$$\begin{array}{rcl} a & = & 3 \\ b & = & 1 \\ f(x) & = & ax + b \quad [\text{数学}] \\ y & = & f(x) \end{array} \quad (5)$$

をプログラムに実装したものが **FunctionMain04.java** です .

この場合 , a, b は **FunctionMain04** クラスのインスタンスフィールドとなり , $f(x)$ は **FunctionMain04** クラスのインスタンスメソッドとなります .

FunctionMain04.java

```
class FunctionMain04{
    double a = 3.0;
    double b = 1.0;

    double f(double x){
        return a * x + b;
    }

    public static void main(String [] args){
        FunctionMain04 fm = new FunctionMain04();
        double x,y;

        x = 3.0;
        y = fm.f(x);
        System.out.println("f("+x+") = "+y);
    }
}
```

FunctionMain04.java をさらに書き換えてみましょう。パラメータ a, b と関数 $f(x)$ は FunctionSet05.java の中に書き、メインメソッドは FunctionMain05.java の中に書きます。

FunctionSet05.java

```
class FunctionSet05{
    double a = 3.0;
    double b = 1.0;

    double f(double x){
        return a * x + b;
    }
}
```

FunctionMain05.java

```
class FunctionMain05{
    public static void main(String [] args){
        FunctionSet05 fs = new FunctionSet05();
        double x,y;

        x = 3.0;
        y = fs.f(x);
        System.out.println("f("+x+") = "+y);
    }
}
```

FunctionMain05.java には

```
FunctionSet05 fs = new FunctionSet05();
```

という1行があります。右端の FunctionSet05() はコンストラクタといいます。コンストラクタは、コンストラクタと同じ名前のクラスに対するインスタンスを生成する際に呼び出されます。FunctionSet05.java の中にコンストラクタの定義は明示的に書かれていませんが、明示的に書くとすれば

FunctionSet05.java

```
class FunctionSet05{
    double a = 3.0;
    double b = 1.0;

    FunctionSet05(){
    }

    double f(double x){
        return a * x + b;
    }
}
```

のようになります。

コンストラクタに関して、まず次の点を覚えておいてください。

- コンストラクタはインスタンスを初期化するためのものです。メソッドに似ていますが、メソッドとは異なり、`void` や `int` などの型修飾子は付けません。
- コンストラクタを定義する際は

クラス名.java

```
class クラス名{  
    ///////////////  
    //コンストラクタ  
    クラス名(引数){  
        処理内容  
    }  
    ///////////////  
}
```

のような書式で書きます。

- コンストラクタがプログラムコード中で明示的に定義されない場合、Java はコンストラクタを自動生成します。この、Java によって自動生成されるコンストラクタをデフォルトコンストラクタといいます。デフォルトコンストラクタは引数を持ちません。

コンストラクタの意義を理解するために，FunctionSet05.java を FunctionSet06.java のように書き変えてみましょう．

FunctionSet06.java

```
class FunctionSet06{
    double a;
    double b;
    //コンストラクタ 6-1
    FunctionSet06(){
        this.a = 1.0;
        this.b = 0.0;
    }
    //コンストラクタ 6-2
    //コンストラクタ 6-1 のオーバーロード
    FunctionSet06(double a, double b){
        this.a = a;
        this.b = b;
    }
    //メソッド 6-1
    double f(double x){
        return a * x + b;
    }
}
```

FunctionMain06.java

```
class FunctionMain06{
    public static void main(String [] args){
        FunctionSet06 fs_def = new FunctionSet06();
        FunctionSet06 fs1 = new FunctionSet06(3.0,1.0);
        FunctionSet06 fs2 = new FunctionSet06(4.0,2.0);
        double x,y;

        x = 3.0;
        y = fs_def.f(x);
        System.out.println("f("+x+") = "+y);
        y = fs1.f(x);
        System.out.println("f("+x+") = "+y);
        y = fs2.f(x);
        System.out.println("f("+x+") = "+y);
    }
}
```

5～8行目は引数がないのでデフォルトコンストラクタであり，a と b の値を初期化しています．10～13行目もコンストラクタで，a と b の値を引数で初期化しています．これらコンストラクタは同じ名前を持っていますが，引数のパターンが異なるので区別することができます．この引数のパターンをシグニチャといいます．シグニチャによって区別されるメソッドやコンストラクタを作成することをメソッドのオーバーロード (overload) / コンストラクタのオーバーロードといいます．

FunctionSet07.java

```
class FunctionSet07 extends FunctionSet06{
    //double a;
    //double b;
    //親クラス (FunctionSet06) で定義されている
    //インスタンスフィールド a,b は
    //子クラス (FunctionSet07) で改めて定義しなくても使える

    double c;

    //コンストラクタ 7-1
    FunctionSet07(double a, double b, double c){
        this.a = a;
        this.b = b;
        this.c = c;
    }
    //メソッド 7-1
    //メソッド 6-1 のオーバーライド
    double f(double x){
        return a*x*x + b*x + c;
    }
    //メソッド 7-2
    //メソッドのオーバーロード
    double f(double x, double z){
        return a*x + b*z + c;
    }
}
```

FunctionMain07.java

```
class FunctionMain07{
    public static void main(String [] args){
        //FunctionSet07 fs_def = new FunctionSet07();
        FunctionSet07 fs = new FunctionSet07(3.0,1.0,1.0);
        double x,y,z;

        x = 3.0;
        y = fs.f(x);
        System.out.println("f("+x+") = "+y);

        z = 2.0;
        y = fs.f(x,z);
        System.out.println("f("+x+", "+z+") = "+y);
    }
}
```

FunctionSet07 クラスは、FunctionSet06 クラスを継承して作られています。すなわち、FunctionSet07 クラスは FunctionSet06 クラスの子クラス（サブクラス）です。

子クラスの中に、親クラス（スーパークラス）と同じグニチャのメソッドを作ることによって親クラスのメソッドを実行できなくしたり、親クラスのメソッドと異なる処理を上書き定義したりすること

をメソッドのオーバーライド (override) といいます。

メソッド 7-1 は、FunctionSet06 クラスのメソッドをオーバーライドして作られています。また、メソッド 7-2 は、メソッド 7-1 をオーバーロードしたものです。

課題 1

FunctionMain05.java にコードを加筆し、 a, b の値を標準出力するプログラム

FunctionMain05_00**000.java を作成せよ。ただし、プログラム名の 00**000 は各自の学籍番号とする。

課題 2

FunctionSet06 クラスを継承して、関数 $f(x, z, w) = ax^2 + bz^2 + cw^2$ をメソッドとして実装した FunctionSet08-00**000 クラスを作成せよ。また、 $f(x, z, w) = ax^2 + bz^2 + cw^2$ の実装は、メソッドのオーバーロードにあたるかオーバーライドにあたるかをコメント「//...」として付記せよ。ただし、00**000 は各自の学籍番号とする。

課題 3

課題 2 で作成したメソッド (関数 $f(x, z, w) = ax^2 + bz^2 + cw^2$ の値を計算するメソッド) を実行するためのメインメソッドを FunctionMain08_00**000 クラスに作成し、プログラムを実行せよ。ただし、引数として $x = 2.0, z = 4.0, w = 1.0$ を代入し、 $a = 1.0, b = 2.0, c = 3.0$ としたインスタンスメソッドと $a = 3.0, b = 5.0, c = 7.0$ としたインスタンスメソッドの 2 つを実行するようにせよ。また、00**000 は各自の学籍番号とする。

提出の際の注意点：

- ・ コンパイルすると実行できる形式で提出すること (FunctionSet06.java も提出すること)。
- ・ 学籍番号のフォルダに提出ファイルを全て入れ、そのフォルダを zip ファイルにして提出すること。

提出先：<https://tdu.app.box.com/f/b4bc2daf357d4b9e99753fa5c48ebcce>

課題の提出期限：6/7 23:59