

2023 年 11 月 30 日 (木) 実施

オブジェクト指向

Java 言語がオブジェクト指向言語であることは初回の授業で触れた。ここでは、**クラス**と**インスタンス**について学習する。

クラスとインスタンス

初回の授業では、クラスの定義の一般形は次の様になることを学んだ。

```

修飾子 class クラス名 {
    型 フィールド名;
    修飾子 戻り値のデータ型 メソッド名( 引数 ) {
        メソッドの定義
    }
}

```

この様にクラスが定義されているだけでは、このクラスを利用することは出来ず、**インスタンス化**という手続きで、**設計図に相当するクラス定義からクラスの実体としてのインスタンスを作成**して初めて利用可能となる。なお、一つのクラスからは複数のインスタンスを作成することが可能である。

インスタンスの作成は **new 演算子**によって行い、**インスタンス変数**に代入して利用する。

```

クラス名 変数名 = new クラス名( 引数 );

```

コンストラクタ

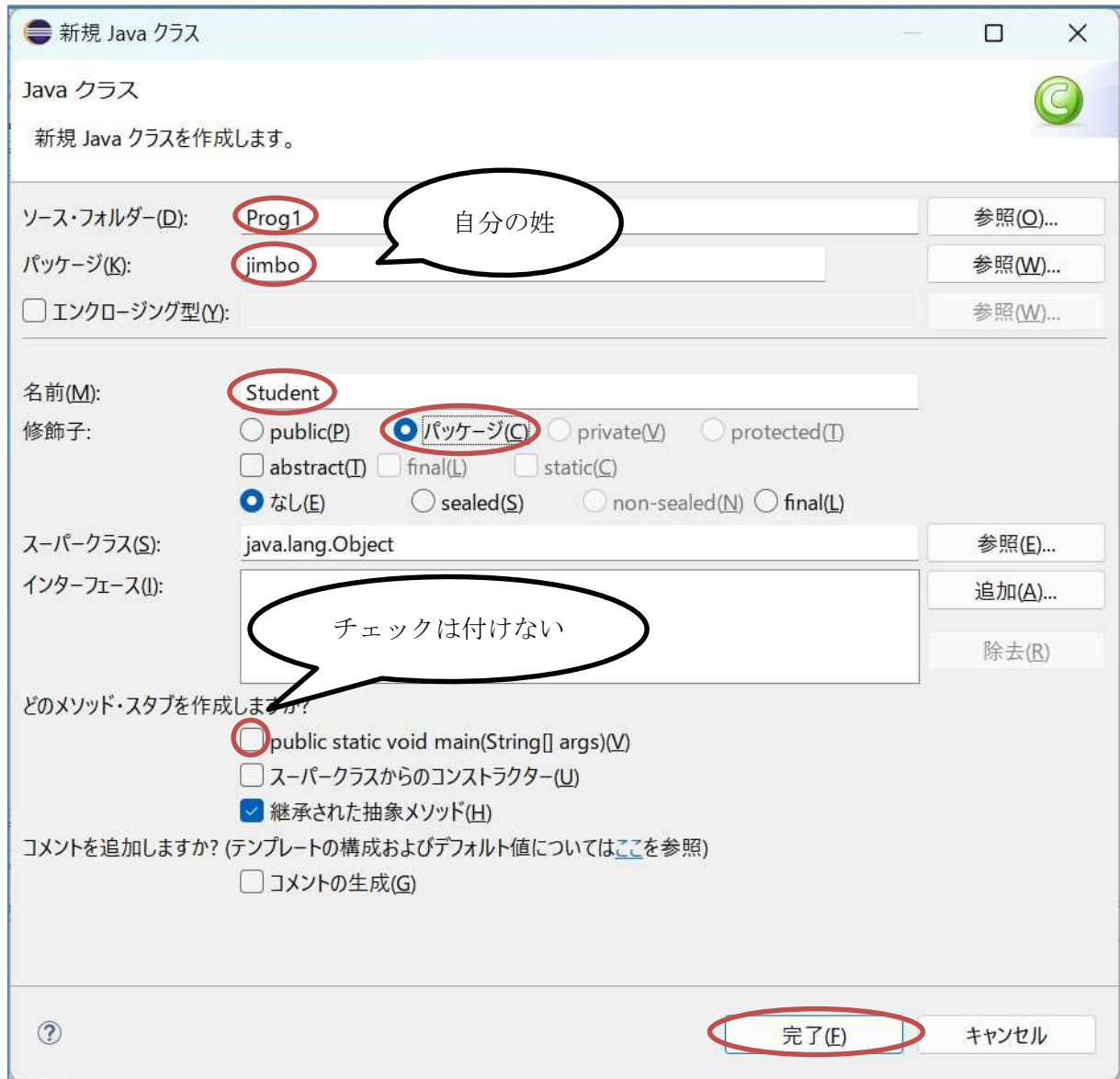
Java 言語では、クラスの定義にフィールド、メソッドの他に、**クラスのインスタンス化の際に呼び出されて実行される処理**を記述する**コンストラクタ**を設けることが出来る。

コンストラクタの特徴としては、次の様なものが挙げられる。

- 1) 名称は**クラス名と同一**である。
- 2) 戻り値はない。
- 3) **インスタンスの初期設定**に利用する。
- 4) コンストラクタからメソッドを呼び出すことも可能である。

Student クラス・Student2 クラス・Student3 クラス

Eclipse で **jimbo** の様に『**自分の姓**』のパッケージを指定して、次の **Student** クラス、**Student2** クラス及び **Student3** クラスを作成せよ。(これは直接実行出来ないクラス ⇒ 後の例題及び演習に利用する)



```

package jimbo;

class Student {
    String id = "00x0000";
    String name = "none";
    String code = "x0000000";
    int point = 0;
    char eval = ' ';

    void setParam(String pid, String pname, String pcode, int ppoint, char peval) {
        id = pid;
        name = pname;
        code = pcode;
        point = ppoint;
        eval = peval;
    }
}
    
```

```
void dataDisp() {
    System.out.println("学籍番号:"+id+"¥t 氏名:"+name+"¥t 授業コード:"+code);
    System.out.println("¥t 点数:"+point+"¥t 評価:"+eval);
}
}
```

【解説】

1. **setParam** メソッドの 5 個の引数は先頭から順に、学籍番号、氏名、授業コード、点数、評価のデータに対応している。
2. ¥t はタブコードのエスケープ文字を表す。
3. **public** 修飾子の付いていないクラスは、同一パッケージ内だけで利用可能となる。

```
package jimbo;

class Student2 {
    String id, name, code = "x0123456";
    int point;
    char eval;

    Student2(String cid, String cname, int cpoint, char ceval) {
        id = cid;
        name = cname;
        point = cpoint;
        eval = ceval;
    }

    void dataDisp () {
        System.out.println("学籍番号:"+id+"¥t 氏名:"+name+"¥t 授業コード:"+code);
        System.out.println("¥t 点数:"+point+"¥t 評価:"+eval);
    }
}
```

コンストラクタ

```
package jimbo;

class Student3 {
    String id, name, code = "x0123456";
    int point;
    char eval;

    Student3() {
        setParam("00x0000", "none", "x0000000", -1, ' ');
    }

    Student3(String cid, String cname, int cpoint) {
        char ev = eva(cpoint);
        setParam(cid, cname, code, cpoint, ev);
    }

    Student3(String cid, String cname, String ccode, int cpoint) {
        char ev = eva(cpoint);
        setParam(cid, cname, ccode, cpoint, ev);
    }
}
```

```

private static char eva(int p) {
    int rank;
    char c;

    if (p < 0 || p > 100)
        c = 'X';
    else {
        rank = p/10;

        switch (rank) {
            case 6: c = 'C'; break;
            case 7: c = 'B'; break;
            case 8: c = 'A'; break;
            case 9:
            case 10: c = 'S'; break;
            default: c = 'D';
        }
    }

    return c;
}

void setParam(String pid, String pname, String pcode, int ppoint, char peval) {
    id = pid;
    name = pname;
    code = pcode;
    point = ppoint;
    eval = peval;
}

void dataDisp() {
    System.out.println("学籍番号:"+id+"¥t 氏名:"+name+"¥t 授業コード:"+code);
    System.out.println("¥t 点数:"+point+"¥t 評価:"+eval);
}
}

```

【解説】複数のコンストラクタが記述されている場合には、インスタンス化で呼び出される際の引数に応じて、対応するコンストラクタが選択される。

例題 1

次のプログラムは **Student** クラスの **setParam** メソッド及び **dataDisp** メソッドの利用の仕方を確認すると共に **Student** クラスのインスタンスの参照について確認するものである。これを入力、ビルドして、実行せよ。ここで、クラス名は **Sample7_1**、ソースファイル名は **Sample7_1.java** とする。

```

package jimbo;

public class Sample7_1 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
    }
}

```

```

String s1 = "23a0345", s2 = "大島優人", s3 = "z4567890";
int po = 88;
char ev = 'A';

Student astudent = new Student();

astudent.setParam(s1, s2, s3, po, ev);
System.out.println("astudent の表示");
astudent.dataDisp();

System.out.println("-----");

Student student = astudent;
student.code = "z5678901";
student.point = 80;

System.out.println("student を変更後:");
System.out.println("student の表示");
student.dataDisp();
System.out.println("astudent の表示");
astudent.dataDisp();
}
}

```

【解説】

1. メソッドは、astudent.dataDisp();の様に、インスタンス名+ドット演算子+メソッド名で実行出来る。なお、setParam メソッドの様に引数がある場合には、必要な数の実引数を記述する。
2. student に代入されるのはインスタンス astudent の格納場所を表す参照である。
3. インスタンスのフィールドは、student.code の様にインスタンス名+ドット演算子+フィールド名で指定される。
4. student のフィールドの値を変更すると、参照先の astudent のフィールドの値が変更される。

例題 2

次のプログラムは Student クラスのインスタンスの配列を作成し、これを利用して 3 人分の学籍番号、氏名、授業コード、点数を入力し、これらに評価を付加して画面に表示するものである。これを入力、ビルドして実行せよ。ここで、クラス名は Sample7_2、ソースファイル名は Sample7_2.java とする。

```

package jimbo;

import java.util.Scanner;

public class Sample7_2 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM = 3;
    }
}

```

```
String s1, s2, s3;
int po;
char ev;
Scanner sc = new Scanner(System.in);

Student[] students = new Student[NUM];

for (int i=0; i<NUM; i++)
    students[i] = new Student();

for (int i=0; i<NUM; i++) {
    System.out.println((i+1)+"人目の");
    System.out.print("学籍番号を入力してください:");
    s1 = sc.next();
    System.out.print("氏名を入力してください:");
    s2 = sc.next();
    System.out.print("授業コードを入力してください:");
    s3 = sc.next();
    System.out.print("点数を入力してください:");
    po = sc.nextInt();
    ev = eva(po);

    students[i].setParam(s1, s2, s3, po, ev);
}

System.out.println("-----");

for (int i=0; i<NUM; i++) {
    System.out.println((i+1)+"人目のデータ:");
    students[i].dataDisp();
}

private static char eva(int p) {
    int rank;
    char c;

    if (p < 0 || p > 100)
        c = 'X';
    else {
        rank = p/10;

        switch (rank) {
            case 6: c = 'C'; break;
            case 7: c = 'B'; break;
            case 8: c = 'A'; break;
            case 9:
            case 10: c = 'S'; break;
            default: c = 'D';
        }
    }

    return c;
}
```

【解説】 students はクラス型インスタンスの配列で、各要素のインスタンス化が必要である。

* **eva** メソッドは第 6 回に配布済み

例題 3

次のプログラムは **Student2** クラスのインスタンスの配列を作成し、**Student2** クラスのコンストラクタの働きを確認するものである。これを入力、ビルドして、実行せよ。ここで、クラス名は **Sample7_3**、ソースファイル名は **Sample7_3.java** とする。

```
package jimbo;

import java.util.Scanner;

public class Sample7_3 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM = 3;
        String[] s1 = new String[NUM];
        String[] s2 = new String[NUM];
        int[] po = new int[NUM];
        char[] ev = new char[NUM];
        Scanner sc = new Scanner(System.in);

        Student2[] students = new Student2[NUM];

        for (int i=0; i<NUM; i++) {
            System.out.println((i+1)+"人目の");
            System.out.print("学籍番号を入力してください:");
            s1[i] = sc.next();
            System.out.print("氏名を入力してください:");
            s2[i] = sc.next();
            System.out.print("点数を入力してください:");
            po[i] = sc.nextInt();
            ev[i] = eva(po[i]);
        }

        for (int i=0; i<NUM; i++)
            students[i] = new Student2(s1[i], s2[i], po[i], ev[i]);

        System.out.println("-----");

        for (int i=0; i<NUM; i++) {
            System.out.println((i+1)+"人目のデータ:");
            students[i].dataDisp();
        }
    }

    private static char eva(int p) {
        int rank;
        char c;

        if (p < 0 || p > 100)
```

```
        c = 'X';
    else {
        rank = p/10;

        switch (rank) {
            case 6: c = 'C'; break;
            case 7: c = 'B'; break;
            case 8: c = 'A'; break;
            case 9:
            case 10: c = 'S'; break;
            default: c = 'D';
        }
    }

    return c;
}
}
```

演習

次のプログラムは **Student3** クラスの **4 個の引数を持つコンストラクタ** を利用する **インスタンスの配列を作成** し、これを利用して **3 人分の学籍番号、氏名、授業コード、点数** を入力し、これらに **評価を付加した結果** を画面に表示するものである。このプログラムリストの空欄 1 ~ 4 _____ に **適切な語句を埋めて** 完成させたプログラムを入力し、ビルドして実行せよ。ここで、クラス名は **Ex7**、ソースプログラム名は **Ex7.java** とする。

```
package jimbo;

import java.util.Scanner;

public class Ex7 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM = 3;
        String[] s1 = new String[NUM];
        String[] s2 = new String[NUM];
        String[] s3 = new String[NUM];
        int[] po = new int[NUM];
        Scanner sc = new Scanner(System.in);

        Student3[] students = new Student3[NUM];

        for (int i=0; i<NUM; i++) {
            System.out.println((i+1)+"人目の");
            System.out.print("学籍番号を入力してください:");
            s1[i] = sc.next();
            System.out.print("氏名を入力してください:");
            s2[i] = sc.next();
```



```
System.out.print("授業コードを入力してください:");
s3[i] = sc.next();

System.out.print("点数を入力してください:");
po[i] = sc.nextInt();
}

for (int i=0; i<NUM; i++)
    students[i] = new Student3( 1 , 2 , 3 , 4 );

System.out.println("-----");

for (int i=0; i<NUM; i++) {
    System.out.println((i+1)+"人目のデータ:");
    students[i].dataDisp();
}
}
```

提出物:

- 1) **例題 1**, **例題 2**, **例題 3** 及び**演習**のプログラムの**コンソールへの出力結果**をコピーして貼り付けた**テキストファイル res7.txt** をメールに添付する。
 - 2) **演習**の**ソースプログラムのファイル Ex7.java** をメールに添付する。
 - 3) 第 7 回の理解度確認用の**質問ファイル Prog1_Questions_7th.txt** に解答を記入して、メールに添付する。
- * メールのはじめは『**プログラミング 1 第 7 回課題**』（鍵括弧は要らない）とする。