

2018 年 5 月 31 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できず、**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、次の様に行う。

```

[クラス名] [変数名] = new [クラス名]([引数]);

```

Student クラス

Eclipse で jimbo の様に『自分の名前』のパッケージを指定して、次の **Student** 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;
    }
}

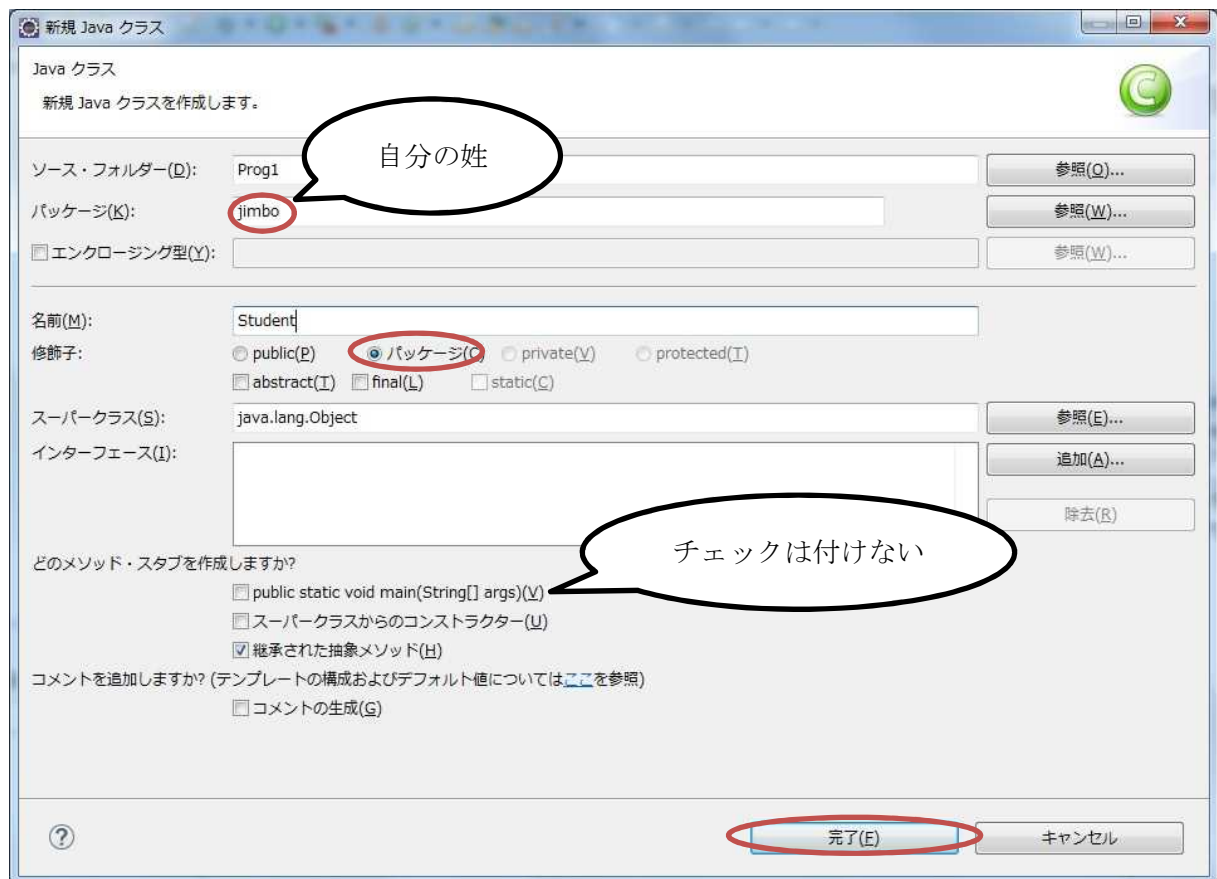
```

```

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



例題 1

次のプログラムは **Student** クラスのインスタンスを複数作成し、**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
        Student astudent = new Student();
        Student bstudent = new Student();
        Student cstudent = new Student();

        astudent.dataDisp();
        bstudent.dataDisp();
        cstudent.dataDisp();

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

        astudent.setParam("18a0321", "上田奈緒子", "z1234567", 95, 'S');
        bstudent.setParam("18b0248", "仲間寛", "w2345678", 70, 'B');
        cstudent.setParam("18c0432", "松雪隆子", "y3456789", 83, 'A');

        astudent.dataDisp();
        bstudent.dataDisp();
        cstudent.dataDisp();
    }
}

```

【解説】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。

例題 2

次のプログラムは 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_2, ソースファイル名は Sample7_2.java とする。

```

package jimbo;

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
        String s1 = "18a0345", 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. 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。
2. 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。
3. 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。

演習

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。このプログラムリストの空欄 1) ~ 5) 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, 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: ");

```

```
1) _____ = sc.next();
System.out.print("氏名を入力してください：");
2) _____ = sc.next();
System.out.print("授業コードを入力してください：");
3) _____ = sc.next();
System.out.print("点数を入力してください：");
4) _____ = sc.nextInt();
5) _____ = 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。

提出物：

- 1) 例題 1, 例題 2 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 6 回の授業の復習の内容を埋めた ファイル Review_6th.txt をメールに添付する。
- * 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。