

2018年6月7日(木)実施

コンストラクタ

前回の授業では、クラスの定義とインスタンス化とを学んだ。Java 言語では、クラスの定義にフィールド、メソッドの他に**コンストラクタ**と呼ばれる、**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。

Student2 クラス・Student3 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 **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 ⇒ 後の例題に利用する)

```
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+"氏名:"+name+"授業コード:"+code);
        System.out.println("氏点:"+point+"氏評価:"+eval);
    }
}
```

```
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 + " 氏名:" + name + " 授業コード:" + code);
        System.out.println(" 点数:" + point + " 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 `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 `Sample8_1`、ソースファイル名は `Sample8_1.java` とする。

```
package jimbo;

import java.util.Scanner;

public class Sample8_1 {

    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;
}
}

```

例題 2

次のプログラムは `Student3` クラスのインスタンスを引数の個数を変えて複数作成し、`Student3` 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 `Sample8_2`、ソースファイル名は `Sample8_2.java` とする。

```

package jimbo;

public class Sample8_2 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        Student3 astudent = new Student3();
        Student3 bstudent = new Student3("17a0543", "向井進", 78);
        Student3 cstudent = new Student3("17e0256", "櫻井譲", "x1234567", 89);
        astudent.dataDisp();
        bstudent.dataDisp();
        cstudent.dataDisp();
    }
}

```

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 `Ex8`、ソースプログラム名は `Ex8.java` とする。

```

package jimbo;

import java.util.Scanner;

```

```
public class Ex8 {  
  
    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 及び 演習 の 出力結果 をコピーして貼り付けた テキストファイル res8.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 Ex8.java をメールに添付する。
- 3) 第 7 回の授業の復習の内容を埋めた ファイル Review_7th.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 8 回課題』(鍵括弧は要らない) とする。