

プログラミング演習2（第13回） 課題

2023年 12月 22日

注意事項

ソースファイルの先頭には必ず「学籍番号」「氏名」「課題番号」を下記の例のようにコメントとして入れること。mainメソッドのあるクラス内に他のメソッドを作らないこと。

```
// 学籍番号:77H000, 氏名:産大太郎, 課題番号:課題1

class Kadai13_1
{
    public static void main(String[] args){
        {
            ...
        }
    }
}
```

課題1. (Kadai13_1.java) 【出題目的:クラスの復習】(10点) 宿題不可

以下のような手順で実行例のように実行するプログラムを作成せよ。

- ① **Student** クラスを用意し、**学籍番号**と**名前**を収納する、**外部からアクセス可能**な String 型のフィールド **number** と **name** を作成する。
- ② **Student** クラスに、**number** に“00H000” **name** に“No name”を代入する、**外部からアクセス可能な、引数なしのコンストラクタ**を用意する。
- ③ **Score** クラスを用意し、**数学と英語の点数**を収納するための、**外部からアクセス可能な** integer 型のフィールド **math** と **english** を作成する。
- ④ **Score** クラスに、**math** と **english** に 0 を代入する、**外部からアクセス可能な、引数なしのコンストラクタ**を用意する。
- ⑤ main 内で、**Student** クラスの変数 **person** を用意する。
- ⑥ main 内で、**Score** クラスの変数 **data** を用意する。
- ⑦ **person** および **data** のそれぞれのオブジェクトの内容を、**各フィールドに直接アクセスして**実行例のように表示する。

【実行例】（青字は変数を表示した結果である。その他は文字列リテラルによる表示である）

```
学籍番号:00H000, 名前: No name
数学:0, 英語:0
```

課題2. (Kadai13_2.java) 【出題目的:継承、メンバへのアクセス】(10点) 宿題不可

課題1で作成したプログラムを修正する形で、以下の手順のプログラムを作成せよ。
main は新たに作成すること。

- ① **Score** クラスを、**Student** クラスをスーパークラスに持つ、**サブクラスに変更する**。
- ② main 内で、**サブクラス**の変数 **member** を用意する。
- ③ **member** のそれぞれのオブジェクトの内容を**各フィールドに直接アクセスして**実行例のように表示する。
- ④ main 内で、**member** の各フィールドに**直接アクセスして**、実行例のように学籍番号、名前、点数を代入する。
- ⑤ **member** のそれぞれのオブジェクトの内容を**各フィールドに直接アクセスして**実行例のように表示する。

【実行例】（青字は変数を表示した結果である。その他は文字列リテラルによる表示である）

```
学籍番号:00H000, 名前: No name, 数学:0, 英語:0
学籍番号:22H777, 名前:Taro, 数学:70, 英語:80
```

課題3. (Kadai13_3.java)

〔出題目的:継承、メンバへのアクセス、オーバーライド〕(10点)

宿題可

課題2で作成したクラスを以下の①～⑤のように追記・修正し、さらに⑥、⑦を加え、課題2の 2 行目の実行例と同じように表示するプログラムを作成せよ。main は新たに作成すること。

- ① **スーパークラスのフィールドおよび引数無しのコストラクタを、サブクラスだけから直接アクセスできるように変更する。**
- ② **スーパークラスに、クラス内のすべてのフィールドを希望の値(文字列を含む)に初期化するための外部からアクセス可能なコストラクタを用意する。**
- ③ **サブクラスのフィールドおよび引数無しのコストラクタを、外部から直接アクセスできないように変更する。**
- ④ **サブクラスに、スーパークラスのフィールドを含めたすべてのフィールドを希望の値(文字列を含む)に初期化する、外部からアクセス可能なコストラクタを用意する。その際 引数は、学籍番号、名前、数学の点数、英語の点数の順とする。スーパークラスのフィールドの初期化にはスーパークラスのコストラクタを利用すること。**
- ⑤ **サブクラスの toString() の戻り値を、課題 2 の実行例のような文字列にするよう、オーバーライドする。(青文字部分は各フィールドの内容が反映されるようにすること)。**
- ⑥ main 内で、**サブクラスの変数 member** を用意する。その際、**コストラクタを利用して、学籍番号、名前、数学の点数、英語の点数** を **member** の各フィールドに代入する。
- ⑦ **member** のオブジェクトの内容を **toString() の戻り値を利用して表示する。**

課題 4. (Kadai13_4.java)

〔出題目的:継承、メンバへのアクセス〕(10点)

宿題可

課題3で作成したクラスに①を追記し、さらに②～④を加えたプログラムを作成せよ。main は新たに作成すること。

- ① **スーパークラスの toString() の戻り値を、以下の実行例の2行目のような文字列 にするよう、オーバーライドする(青文字部分は各フィールドの内容が反映されるようにすること)。**
- ② main 内で、**スーパークラスの要素数 3 の配列変数 members** を用意する。
- ③ **実行例を参考に、3 人分の学籍番号、名前、数学の点数、英語の点数を、コストラクタを利用して members の各要素に代入する。(Taro と Hanako は Score クラス、Ichiro は Student クラスのオブジェクトである。)**
- ④ **for ループを 1 つだけ用いて、members のすべての要素の内容を実行例のように表示する。その際、toString() を利用すること。繰り返し回数は、配列の長さを読み取って設定すること。**

〔 実行例 〕 (青字は変数を表示した結果である。その他は文字列リテラルによる表示である)

```
学籍番号:22H777, 名前:Taro, 数学:70, 英語:80  
学籍番号:22H778, 名前:Ichiro  
学籍番号:22H779, 名前:Hanako, 数学:90, 英語:50
```

- 演習時間内に出来なかったプログラムのソースファイル(.java)は、**下記の期限までにWebClass** で提出すること。
- 提出されたプログラムは**すべてチェック**します。動作しないプログラムが提出されている場合、提出物をすべて**無効**とします。ファイル名が間違っている場合は**未提出と判断**します。

提出期限: **次回授業前日(1月 11日)18:40** まで

提出場所: **WebClass**: **金 4 後 プログラミング演習 2 H 松本 恵治 1 2023**

13-2 課題(宿題) Kadai13 3-4

注意: **13-2 課題(宿題) Kadai13 3-4** には、課題チェックシート(提出物)にチェック(合格)されたファイルを再度提出してはならない。提出している場合、課題としてチェックされたものが取り消しとなる。

- ソースファイルの先頭には必ず「学籍番号」「氏名」「**課題番号**」を下記の例のようにコメントとして入れること。

```
/* 学籍番号:77H777, 氏名:産大太郎, 課題番号:課題 3 */  
class Kadai13_3  
{  
    public static void main(String[] args){
```

- 期限に遅れた場合は、**評価外**とする。ただし、病気等の理由がある際には**診断書**あるいは**領収書**等を持参の上、**担当教員**に了解を得ること。
- **復習テスト**が再公開されているので、まだ完了していない学生は完了まで実施すること。
- **Lesson12** の Sample 1 (**Sample12_01.java**) から Sample 4 (**Sample12_04.java**) までのソースファイル(.java) 4 個を提出すること。

提出期限: **次回授業前日(1月 11日)18:40** まで

提出場所: **WebClass**: **金 4 後 プログラミング演習 2 H 松本 恵治 1 2023**

14-1 予習 Sample12_01-04

- ソースファイルの先頭には必ず「学籍番号」「氏名」「**Sample 番号**」を下記の例のようにコメントとして入れること。

```
// 学籍番号:77H777, 氏名:産大太郎, Sample 番号:1  
class Sample12_01  
{  
    ...
```

- **次回**は**Lesson12** の説明、課題を行うので、内容を熟読しておくこと。