

プログラミング演習2（第14回） 課題

2024年 1月 12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 Kadai14_1
{
    public static void main(String[] args){
        {
            ...
        }
    }
}
```

課題1. (Kadai14_1.java)

〔出題目的:抽象クラス〕(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 以下のフィールドおよびメソッドを持つ**抽象クラス Person**を作成する。
 - **サブクラスからのみアクセス可能な**、名前を収納する String 型のフィールド **name**,
 - **サブクラスからのみアクセス可能な**、身長, 体重を収納する double 型のフィールド **height**, **weight**
 - **setPerData**: データを代入する**抽象メソッド**
 - 引数 2 個: 1 目 int 型(データの種類,type)、2 目 double 型(データの値,value)
 - 戻り値 なし
 - **showPerData**: **public** メソッド。すべてのフィールドを実行例のように表示する。
- ② 以下の機能を有する、**Person** クラスのサブクラス **Student** を作成する。
 - すべてのフィールドを宣言時に設定する **public** 型のコンストラクタ(引数 3 つ)。
 - **setPerData**: **public** メソッド。type が 0 なら value を height に、1 なら value を weight に代入。
- ③ main 内で、**student** を **Student クラス** のオブジェクトとして作成し、名前、身長、体重の値を、**コンストラクタ** を用いて代入する(キーボードから入力する必要はない)。
- ④ **student** の内容を、**showPerData** を使い表示する。
- ⑤ **setPerData** を使い身長と体重を変更する。
- ⑥ **student** の内容を、**showPerData** を使い表示する。

〔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）

```
名前: Taro, 身長: 185.7 cm, 体重: 82.6kg
名前: Taro, 身長: 162.3 cm, 体重: 55.6kg
```

課題 2. (Kadai14_2.java)**〔出題目的: インターフェイス〕(10点)****宿題不可**

以下のような手順で、課題 1 の実行例と同じように実行するプログラムを作成せよ。

- ① 以下のメソッドを持つ **インターフェイス iPerson** を作成する。
 - **setPerData**: データを代入するメソッド
 - 引数 2 個: 1 目 int 型(データの種類,type)、2 目 double 型(データの値,value)
 - 戻り値 なし
 - **showPerData**: 情報を表示するメソッド。引数、戻り値無し
- ② 以下のフィールドおよびメソッドを有する、**iPerson** を実装したクラス **Student** を作成する。
 - **外部からアクセスできない**、名前を収納する String 型のフィールド **name**,
 - **外部からアクセスできない**、身長および体重を収納する double 型のフィールド **height**, **weight**
 - すべてのフィールドを宣言時に設定する **public** 型のコンストラクタ(引数 3 つ)。
 - **setPerData**: **public** メソッド。type が 0 なら value を height に、1 なら value を weight に代入。
 - **showPerData**: **public** メソッド。すべてのフィールドを実行例のように表示する。
- ③ main 内で、**student** を **Student クラス** のオブジェクトとして作成し、名前、身長、体重の値を、**コンストラクタ** を用いて代入する(キーボードから入力する必要はない)。
- ④ **student** の内容を、**showPerData** を使い表示する。
- ⑤ **setPerData** を使い身長と体重を変更する。
- ⑥ **student** の内容を、**showPerData** を使い表示する。

課題3. (Kadai14_3.java)**〔出題目的: 多重継承〕(10点)****宿題可**

課題 2 で作成したプログラムを用いて、以下のようなプログラムを作成せよ。

- ① 以下のフィールドおよびメソッドを持つ新たな **インターフェイス iSchool** を作成する。
 - **UNIV**: int 型のフィールド。大学の必要在籍年数(4)
 - **HIGH**: int 型のフィールド。高校の必要在籍年数(3)
 - **JUNI**: int 型のフィールド。中学校の必要在籍年数(3)
 - **ELEM**: int 型のフィールド。小学校の必要在籍年数(6)
 - **ENT**: int 型のフィールド。小学校 1 年生時の年齢(7)
 - **getSchool**: 現在在学してる学校についての情報を返すメソッド。引数 int 型 1 つ、戻り値 String 型。
 - **getScYear**: 現在の学年を返すメソッド。引数 int 型 1 つ。戻り値 int 型。
- ② 課題 2 で作成した **Student クラス** に、さらに **iSchool** を **追加実装** する。
- ③ **Student クラス** に以下のフィールドを **追加** する。
 - **外部からアクセスできない**、生まれた年(西暦)を収納する、**int 型** のフィールド **bYear**
- ④ すべてのフィールドを宣言時に設定できるように **コンストラクタ** を修正する(引数 4 つ)。
順番: 名前、身長、体重、生まれた年
- ⑤ 課題 2 で作成した **setPerData** に、**type が 2 なら bYear に代入** するよう、追記する(代入時、引数を int 型でキャストすること)。
- ⑥ 課題 2 で作成した **showPerData** を、実行例の 1 行目のように表示するよう、修正する。
- ⑦ **Student クラス** に必要な以下の 2 つのメソッドを追加する。必要不可欠なメソッド以外にメソッドは作成しないこと。下記のメソッド内での判断の際には **インターフェイス** で定義されている定数を活用すること。
 - getSchool**: **public** メソッド。引数には西暦を入力。その年に在学している学校の種別を、未入学、小学校、中学校、高校、大学、社会人から選んで返す。
 - getScYear**: **public** メソッド。引数には西暦を入力。その年の学年を整数で返す(社会人及び未入学の場合は -1 を返す)。

- ⑧ main 内は、実行例のように実行されるよう、課題2のプログラムに以下の点を追記する。すでに書かれている部分は削除しないこと。
1. main 内で西暦を保存する int 型の整数 year を用意し、2023 で初期化する。
 2. **student** を **Student クラス** のオブジェクトとして作成し、実行例 1 行目の値を**コンストラクタ**を用いて代入する。(コンストラクタへの引数部分のみ修正)
 3. **getSchool、getScYear** を使い、**実行例 2 行目のように、表示する。**
 4. **setPerData** を使い、生まれた年を実行例 3 行目のように変更する命令を追加する。
 5. **getSchool、getScYear** を使い、**実行例 4 行目のように、表示する。**

〔実行例〕（青字は変数もしくはメソッドの戻り値を表示した結果、その他は文字列リテラルによる表示である）

名前: Taro, 身長: 185.7 cm, 体重: 82.6kg, 生まれた年: 2003 年
西暦 2023 年のとき、大学 2 年生
名前: Taro, 身長: 162.3 cm, 体重: 55.6kg, 生まれた年: 2013 年
西暦 2023 年のとき、小学校 4 年生

課題 4. (Kadai14_4.java)

〔出題目的: サブインターフェイス〕(10点)

宿題可

課題 3 で作成したプログラムを用いて、以下のようなプログラムを作成せよ。

- ① 以下のメソッドを持つ新たな**インターフェイス iUniv** を **iSchool** を**拡張して作成する**。
 - **showUnivData**: 情報を表示するメソッド。引数、戻り値無し。
- ② **課題 2 で作成した Student クラス**に実装したインターフェイス **iSchool** を外し、代わりに **iUniv** を**実装**する。
- ③ Student クラスに以下のフィールドを**追加**する。
 - **外部からアクセスできない**、大学名を収納する **String 型**のフィールド **uName**
 - **外部からアクセスできない**、学部名を収納する **String 型**のフィールド **gakubu**
- ④ すべてのフィールドを宣言時に設定できるように**コンストラクタ**を修正する(引数 6 つ)。
順番: 名前、身長、体重、生まれた年、大学名、学部名
- ⑤ Student クラスに必要なメソッドを追加する。必要不可欠なメソッド以外にメソッドは作成しないこと。
showUnivData()による表示は、実行例の 3 行目である。
- ⑥ main 内は、課題 3 のプログラムに以下の点を追記する。**すでに書かれている部分は削除しないこと**。
 1. **student** を **Student クラス**のオブジェクトとして作成する際、**学校名と学部も設定する**(値は実行例を参考に)。(コンストラクタへの引数部分のみ修正)
 2. 課題 3 の実行例の 2 行目および 4 行目の表示をした行の次の行で、**現在の学校が大学であるかどうかを判断して、大学であれば、作成したメソッドを使い、大学名、学部名を表示する**。(2 つとも同じコードを記入する)。
注意: 実行例の 6 行目は何も表示されていないが、5 行目の後で判断した結果、表示してない状況。

〔実行例〕（青字は変数もしくは定数を表示した結果、その他は文字列リテラルによる表示である）

名前: Taro, 身長: 185.7 cm, 体重: 82.6kg, 生まれた年: 2003 年
西暦 2023 年のとき、大学 2 年生
所属: 大阪産業大学/デザイン工学部
名前: Taro, 身長: 162.3 cm, 体重: 55.6kg, 生まれた年: 2013 年
西暦 2023 年のとき、小学校 4 年生

- 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: **来週木曜日(1月 18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**

14-2 課題(宿題) Kadai14 3-4

注意: **14-2 課題(宿題) Kadai14 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す。

- 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 Kadai14_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。
- **次回は、これまでの全範囲の復習課題**を行うので、これまでの学習内容を復習しておくこと。