

オブジェクト指向プログラミング 第5回

アクセス修飾子

担当：高橋、佐藤聖也

1 アクセス修飾子

変数を定義する際、変数を表す文字・文字列の前に置くキーワード（予約語）を修飾子（modifier）といいます。int, doubleなどは、変数の型を決めるので型修飾子です。型修飾子の前に付ける public, protected, privateなどのキーワードをアクセス修飾子（access modifier）といいます。アクセス修飾子は、変数の値を書き換えるなどの操作ができるプログラムの範囲¹を決めるためのものであり、フィールド、メソッド、コンストラクタ、クラスに対して付けることができます。アクセス修飾子の具体的な働きは順を追って説明していきませんが、差し当たり、アクセス制限として以下のような順序があると考えておいてください。

[制限弱い] public > protected > （アクセス修飾子なし） > private [制限強い]

2 パッケージ

アクセス修飾子のスコープを調べるためには、まずパッケージのインポートについて確認する必要があります²。

これまでもパッケージは利用してきました。たとえば、java.awt パッケージの中に含まれる Frame クラスを利用する際には

```
import java.awt.Frame;
```

のような import 文を使う必要がありました。また、Frame クラスや Color クラス、Graphics クラスなど java.awt パッケージの中に含まれる任意のクラスを利用する際には

```
import java.awt.*;
```

のように書きました。ちなみに java.lang パッケージは import 文によってインポートしなくても、自動的にインポートされます。

2.1 自分でパッケージを定義する

Java にあらかじめ用意されているパッケージを使う以外に、自分でパッケージを定義することもできます。いま、カレントディレクトリが「H:\CompExB\05」であるとしましょう。ここに、ひとつのフォルダを作り、フォルダ名を「p051」などとしておきます。この「p051」がパッケージ名となります。

1 行目に

```
package p051;
```

という package 文を書いた以下のような BaseSample.java を「p051」フォルダに保存し、「p051」パッケージに含まれるクラスを作ります。

¹この範囲のことをスコープといいます。

²パッケージについては第3回のレジュメにも記述があります。

BaseSample.java

```
1 package p051;
2
3 public class BaseSample{
4     public int w;
5     protected int x;
6     int y;
7     private int z;
8 }
```

このクラスはメインメソッドを含まないなので、コンパイルのみ可能で、実行はできません。コンパイルはコマンドプロンプトから以下のように行います³。

```
H:\¥CompExB¥05 > javac p051/BaseSample.java
```

2.2 同一パッケージ・子クラスからのアクセス

次に、BaseSample クラスを継承した ChildSample クラスをつくります。

ChildSample.java

```
1 package p051;
2
3 class ChildSample extends BaseSample{
4     //コンストラクタ
5     ChildSample(){
6         this.w = 1;
7         this.x = 2;
8         this.y = 3;
9         //this.z = 4;
10    }
11 }
```

このクラスをコンパイルすると、BaseSample クラスで定義したアクセス修飾子が子クラスに対して行うアクセス制限を知ることができます。実際にコンパイルしてみると

```
this.z = 4;
```

でコンパイルエラーが起きます(サンプルコードはコメントアウト済み)。これは BaseSample クラスで、変数 z が private 修飾子でアクセス制限されているからです。

練習問題

ChildSample.java に w, x, y の値を標準出力させるメインメソッドを書き加えた ChildSample02.java を作成し実行しなさい。

³ここでは eclipse を使わずにコマンドプロンプトからの操作を行ってみてください。

2.3 同一パッケージ・他クラスからのアクセス

次に、BaseSample クラスを継承していない、NotChildSample クラスをつくります。

NotChildSample.java (これはコンパイルエラーが出ます)

```
1  package p051;
2
3  class NotChildSample{
4      NotChildSample(){
5          BaseSample bs = new BaseSample();
6          bs.w = 10;
7          bs.x = 20;
8          bs.y = 30;
9          bs.z = 40;
10     }
11 }
```

NotChildSample クラスは BaseSample クラスの子クラスではないので、BaseSample クラスのインスタンスを生成しなければ w や x などの変数は使えないことに注意しましょう。

練習問題

NotChildSample.java をコンパイルしたときエラーとなるのはどのコードが確かめなさい。また、NotChildSample.java を書き換え、新しい private 変数 private int w,x,y; を定義し、これらに bs.w, bs.x, bs.y の値を代入した上で標準出力させる NotChildSample02.java を作成し実行しなさい。

2.4 アクセサメソッドによる private 変数へのアクセス

private 修飾子によってアクセス制限されている変数にアクセスするためのメソッドを用意することがあります。値を代入するメソッドを setter メソッド、値を読み出すメソッドを getter メソッドといい、これらを合わせてアクセサメソッドといいます。BaseSample クラスにアクセサメソッドを追加したのが以下の BaseSample02 クラスです。

BaseSample02.java

```
1  package p051;
2
3  public class BaseSample02{
4      public int w;
5      protected int x;
6      int y;
7      private int z;
8
9      public void setZvalue(int z){
10         this.z = z;
11     }
12
13     public int getZvalue(){
14         return this.z;
15     }
16 }
```

練習問題 (→ 課題1)

ChildSample02.java を書き換え, BaseSample02 クラスを親クラスに持つ ChildSample03_00rd000 クラス (ChildSample03_00rd000.java ファイル) を作成し, アクセサメソッドを用いて z の値を読み書きさせなさい. ただし, 00rd000 は各自の学籍番号とする.

練習問題

BaseSample02 クラスの public 修飾子を消して

```
package p051;
class BaseSample02{
    ... (省略) ...
}
```

としたとき, 上の練習問題で作成した ChildSample03 クラスは正常にコンパイルと実行ができるか確かめなさい.

2.5 他パッケージ・子クラスからのアクセス

今度は, ディレクトリ「H:\¥CompExB¥05」に別のパッケージ用フォルダ「p052」を作成し, 以下の ChildAnotherPackageSample.java ファイルを保存します. 「p051」パッケージにある BaseSample02 クラスを継承するために「import p051.*;」で「p051」パッケージをインポートしていることに注意してください.

ChildAnotherPackageSample.java (これはコンパイルエラーが出ます)

```
1 package p052;
2
3 import p051.*;
4
5 class ChildAnotherPackageSample extends BaseSample02{
6     ChildAnotherPackageSample(){
7         this.w = 111;
8         this.x = 222;
9         this.y = 333;
10        this.z = 444;
11    }
12 }
```

練習問題

ChildAnotherPackageSample.java をコンパイルしたときエラーとなるのはどのコードが確かめなさい. また, この ChildAnotherPackageSample.java を書き換え, アクセサメソッドで z に値を代入するコンストラクタを持ちメインメソッドで w, x, z の値を標準出力する ChildAnotherPackageSample02.java を作成し実行しなさい.

練習問題

BaseSample02 クラスの public 修飾子を消して

```
package p051;
class BaseSample02{
    ... (省略) ...
}
```

としたとき，上の練習問題で作成した ChildAnotherPackageSample02 クラスは正常にコンパイルと実行ができるか確かめなさい．

2.6 他パッケージ・他クラスからのアクセス

練習問題（→ 課題 2）

今日のサンプルコードを参考にしながらフォルダ「p052」の中に，他パッケージ・他クラスからのアクセスを確かめるための「NotChildAnotherPackageSample_00rd000 クラス」を作成し，コンパイルしなさい．ただし，00rd000 は各自の学籍番号とする．

課題

課題 1

練習問題で作成した ChildSample03_00rd000.java を提出しなさい．ただし，00rd000 は各自の学籍番号とする．

課題 2

練習問題で作成した NotChildAnotherPackageSample_00rd000.java を提出しなさい．ただし，コンパイルエラーが出る箇所はコメントアウトしておくこと．また，00rd000 は各自の学籍番号とする．

提出の際の注意点：

- ・ コンパイルすると実行できる形式で提出すること。
- ・ 学籍番号のフォルダに提出ファイルを全て入れ、そのフォルダをzipファイルにして提出すること。

提出先：<https://tdu.app.box.com/f/fd7eeb40e32f44958019ef68edd4d46a>

課題の提出期限：6/14 23:59