

プログラミング演習2 課題 補足資料

第1週: JAVAプログラミングの復習

練習課題1-1

Point2Dクラスの作成①クラスの枠組み作成

※プログラミング言語の教科書 8-2-1も参照のこと

- クラスの作成手順

- エディタを開き、以下を入力する

```
public class Point2D {  
}
```

- 作成したいクラスの名前がPoint2Dなので
Point2D.java というファイル名で保存する
- 以上で中身は空のPoint2Dクラスが完成する
- その後、中身を書いていく

練習課題1-1

Point2Dクラスの作成②フィールド変数の定義

※プロ言教科書 8-2-1も参照のこと

- フィールド変数: クラスが保持する値を保存する変数のこと
ここでは、double型の座標値
- フィールド変数を追加すると

```
public class Point2D {  
    private double x; // x座標を保持  
    private double y; // y座標を保持  
}
```

- 一般にフィールド変数はprivateで定義する
 - クラス外から、変数の値の参照・変更をできなくしておく
→そのため、アクセサが必要となる(練習課題1-3)

練習課題1-1

Point2Dクラスの作成③コンストラクタの作成

※プロ言教科書 8-2-4も参照のこと

- コンストラクタ: インスタンス生成時に実行される特別なメソッド
- コンストラクタを追加すると、

```
public class Point2D {  
    private double x; // x座標を保持  
    private double y; // y座標を保持  
  
    public Point2D(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

- コンストラクタの引数(x, y)をフィールド変数(x, y)に代入している
- this.xは、そのクラスのフィールド変数の変数 x という意味

練習課題1-1

thisとは

- 作成したクラスには、それぞれ2種類の変数 x, y が存在
 - フィールド変数である x, y
 - コンストラクタのローカル変数である x, y
- メソッド(コンストラクタ)内で x とすると、ローカル変数 x が参照される
→ フィールド変数が参照できないと困るので
 $this.x$ として、フィールド変数の x であることを明示する
- 同じ名前のローカル変数が存在しないときは、
 $this$ を付けなくてもOK

```
public class Point2D {  
    private double  $x$ ; // x座標を保持  
    private double  $y$ ; // y座標を保持  
  
    public Point2D(double  $x$ , double  $y$ ) {  
        this. $x$  =  $x$ ;  
        this. $y$  =  $y$ ;  
    }  
}
```

練習課題1-2

printメソッドの追加

※プロ言教科書 8-2-4も参照のこと

- Point2Dクラスが保持する座標値x, yを表示したいので、printメソッド内で、フィールド変数を画面表示すればよい
- 具体的にはthis.x, this.y を表示する

```
public class Point2D {  
    :  
    練習課題1-1  
    :  
    public void print() {  
        System.out.println("(" + this.x + ", " + this.y + ")");  
    }  
}
```

練習課題1-3

アクセサの追加

- アクセサ: フィールド変数の値取得・設定用のメソッド
 - フィールド変数x, yはprivateなので、アクセサを通じてアクセス
- メソッドの作成方法自体は練習課題1-2と同様
 - getX(), getY()では、this.x, this.yの値を返す
 - setX(), setY()では、this.x, this.yに値を代入

```
public class Point2D {  
    :  
    public double getX() {  
        return this.x;  
    }  
    public void setX(double x) {  
        this.x = x;  
    }  
}
```

フィールド変数x (this.x)に
引数xを代入

※場所の都合で、
xに関してのみ

練習課題1-3

動作確認用プログラムの作成

※プロ言教科書 8-2-2も参照のこと

- Point2Dクラスにはmainメソッドがなく、実行できないので、以下のいずれかの方法で、mainメソッドを用意する
 - 動作確認用クラス(例えばTestEx0103クラス)を作成する(以下の例)
 - Point2Dクラス内に動作確認用のmainメソッドを作成してもよい

```
public class TestEx0103
{
    public static void main(String[] args)
    {
        //Point2Dクラスのインスタンスを生成
        Point2D p1 = new Point2D(0.0, 0.0);

        p1.setX(2.4);
        p1.setY(10.7);
        p1.print();
    }
}
```

この例では、
setX(), setY(), print()メソッドの
動作を確認している

(2.4, 10.7)

と画面に出力されればOK

必須課題1-4

Line2Dクラスの作成

- Point2Dクラスと異なる点は、以下の通り
Line2Dクラスは、線分は2点で定義されるので、
 - フィールド変数として、2点を保持するためのPoint2Dクラスを2つを持つ
 - コンストラクタの引数も、2点 (Point2Dクラス2つ)

```
public class Line2D {  
    private Point2D p1; //線分の始点  
    private Point2D p2; //線分の終点  
  
    //コンストラクタ  
    public Line2D(Point2D p1, Point2D p2) {  
        // 中身を書く  
    }  
    // アクセサとprintメソッドを作成  
}
```

コンストラクタの引数は
Point2Dクラスを2つ

- アクセサとprintメソッドについては、練習課題1-2,1-3を参考に

必須課題1-5

lengthメソッドの追加

- 線分の長さを計算するには、2点の座標があればよく
両方の座標を既にLine2Dクラスは保持しているため、
長さの計算には、他の情報は必要ない(= 引数を必要としない)