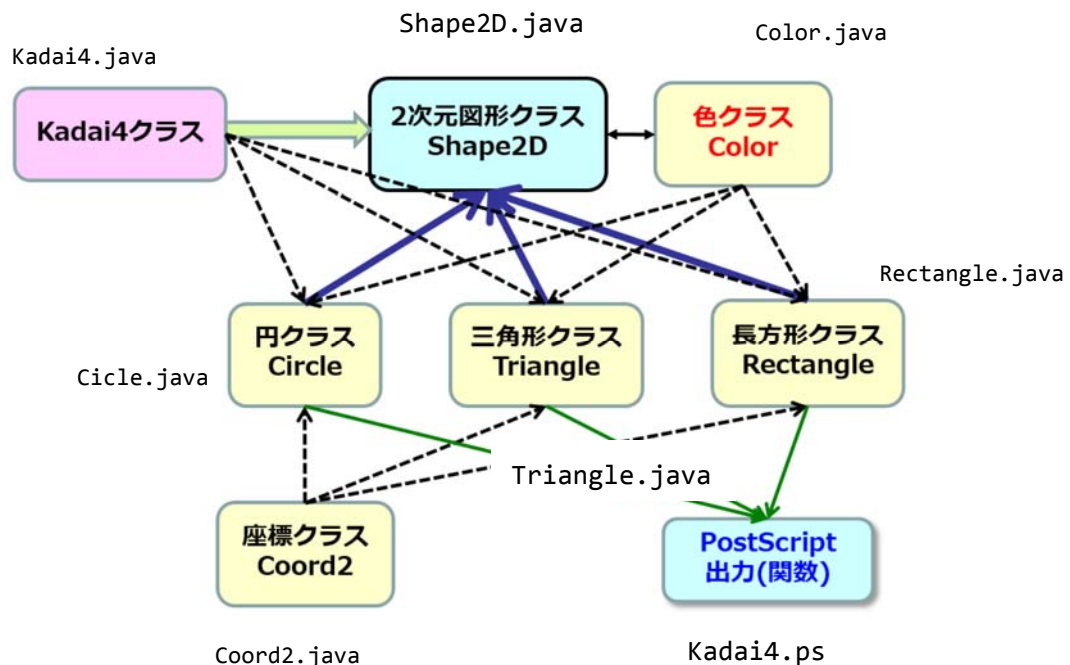


ソフトウェア演習ⅡB〔課題4：抽象クラスによる継承〕担当：青野雅樹

イラストを参考に、基底クラスが **Shape2D** という名前の 2 次元図形の抽象的なクラスで、派生クラスとして、円、三角形、(x 軸と y 軸と平行な辺からなる)長方形を有する Java プログラム群を作成せよ。実行にあたっての **main** 関数を含むクラスは **Kadai4.java** とし、ここで各種図形の発生や **PostScript** への出力を制御すること。実行結果はこれまでと違い、**PostScript** データファイルとなる。そこで、Java プログラムと **PostScript** データファイルを **ZIP** でまとめて Moodle にアップせよ。〆切は **7 月 20 日 (火)** の夜までとする。後述するすべての条件を満たすようにプログラムすること。条件以外の変数や関数を追加することは構わない。



【条件1】(基底クラス Shape2D) Shape2D.java

クラスのタイプ: `public abstract class Shape2D`

メンバー変数:

`private Color color; //Color クラスは後述`

メンバー関数:

`public void setColor(Color c); //Color 変数に値をセットする関数`

`public Color getColor(); //戻り値として Color 変数を返す関数`

`abstract double area(); //面積を計算する抽象関数`

`abstract double perimeter(); //周囲長を計算する抽象関数`

```
abstract void psPrint(PrintStream cout);//プリント関数
```

【条件2】(Color クラス) Color.java

メンバー変数:

```
private double r,g,b;//r(赤), g (緑), b(青), 0.0<=r,g,b <=1.0
```

メンバー関数:

```
public double getR(); //赤色を返す関数
```

```
public double getG(); //緑色を返す関数
```

```
public double getB(); //青色を返す関数
```

【条件3】(Coord2 クラス) Coord2.java 2次元座標を表現するクラス

メンバー変数:

```
private double x, y;
```

メンバー関数:

```
public double getX();//x座標を返す
```

```
public double getY();//y座標を返す
```

```
public void setCoord2(double x, double y);//座標値をセット
```

```
public static double distance(Coord v1, Coord v2);//ユークリッド距離
```

【条件4】(Triangle クラス) Triangle.java (Shape2D クラスの派生クラス)

メンバー変数:

```
private Coord2 v1, v2, v3;//又は private Coord2 v[3];三頂点座標
```

メンバー関数:

```
public Triangle(Coord2 v1, Coord2 v2, Coord2 v3, Color c);
```

```
public Triangle(Coord2[] v, Color c);
```

```
public Coord2[] getV(); // 3 頂点を返す(v1,v2,v3 の順)
```

```
public double perimeter();//三角形の周囲長を返す関数
```

```
public double area();//三角形の面積を返す関数
```

```
public void psPrint(PrintStream cout);//PostScript で出力
```

【条件5】(Rectangle クラス) Rectangle.java 長方形クラス

メンバー変数とメンバー関数は、三角形クラスに準じるが、メンバー変数では、長方形の左上座標(v1)と右下座標(v2)を private 変数でもたせる

【条件6】(Circle クラス) Circle.java 円クラス

メンバー変数とメンバー関数は、三角形クラスに準じるが、メンバー変数では、円

の中心座標(v)と半径(r)を **private** 変数でもたせる

【条件 7】 (Kadai4 クラス) Kadai4.java メインクラス

実行にあたっての **main** 関数を含む **Kadai4.java** には、**Color** クラスの生成に必要な赤、緑、青をランダムに発生する部分と、**Triangle** クラス、**Rectangle** クラス、**Circle** クラスをランダムに生成する部分を加える。

- (1) 実行にあたっては、プログラムの外から、**PostScript** ファイル名と図形総数を下図の太枠内に示すように与える。ただし、図形総数は 10~50 の間の数とする。
- (2) **PostScript** のキャンバスは **X 方向サイズを 600, Y 方向サイズを 800** とする。(資料のプログラム 4-6 参照。変数名やデータ型をプログラム 4-6 と同じとしてよい)
- (3) **PostScript** の末尾に総面積の値、総周囲長、総面積/総長の比、ならびに図形総数の値を **PostScript コメント** (行の先頭文字に%)として書き出すこと。(ファイルの先頭に学籍番号、氏名、日付を入れること)。なお、小数点以下は 3 桁程度でよい。
- (4) **PostScript** の名前は、**Kadai4.ps** とし、以下のように実行する

PostScript 末尾例 :

```
% 総面積 = 922620.930
% 総長 = 430420.910
% 総面積/総長比 = 2.140
% 図形総数 = 21
showpage
```

プログラム実行コマンド例

```
$ java(u) Kadai4 Kadai4.ps 30
```

コメントとヒント :

PostScript のサンプルは、授業の資料内 (プログラム 4-6 の実行コメント) にも記述していますが、<https://www.kde.cs.tut.ac.jp/~aono/2021/output.ps> にあります。ただし、こちらは、本課題とは条件が違います (**Color** 情報がないこと、図形として長方形がないこと、末尾に総面積等の計算結果がないことなどの点で異なります)。

以下は、あくまで例ですが、**Color** 変数を各図形のコンストラクタ (のパラメータ) に含ませる場合の例 (ここでは図形は円) を紹介します。

```
public Circle(Coord2 v, double radius, Color c){
    this.radius = radius; /* 半径 radius */
    this.v = v; /* 中心座標 */
    super.setColor(c); /* 基底クラスの setColor メソッド */
}
```

ポイントは、基底クラスの setColor メソッドを呼出すとき、super という Java 特有の親クラス変数を示すキーワードを使用する点です。色を与える PostScript コマンドは、"r g b setrgbcolor" (例: 0.31 0.75 0.10 setrgbcolor) のようなコマンドで与えると色として解釈してくれます。r, g, b 部分の書き出しは、たとえば String の format メソッドで %6.2f (あるいは %6.3f) 程度の変換で実数値を書き出せば結構です。円を描画する PostScript は以下のような命令としてください。

```
%%円
0.1 0.2 0.91 setrgbcolor
newpath
151.0 400.1 124.5 0 360 arc
stroke
```

多角形(三角形、長方形)の描画は、出だしの色と最後の stroke は円と同じで、違うのは、最初の頂点に x y moveto で移動し、以降、x y lineto で線分を結ぶことで行います。最後の点と最初の点を結ぶ線分は、最後に closepath で図形を閉じることで描画できます。

多角形の符号付き面積は、たとえば $Area = \frac{1}{2} \sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i$ で計算できます。ただし、

$x_n = x_0$ で $y_n = y_0$ です。正負の値があり得ますので、最終的な総和が負の場合もあります (三角形の場合も、最初に定義した頂点の順序によっては負もあります)。したがって、上記の式で面積を計算する場合は絶対値をとるようにしてください)

右図は n=21 でこの課題を実行した PostScript を表示した例です。(PostScript の表示方法は、たとえば GSView (Ghostscript) などのフリーソフトで表示できます。OS によっては、直接表示できる場合もあります (たとえば Mac では直接表示できると思います)。また、有償の Acrobat Pro をお持ちの場合は、PDF に変換してから表示確認することもできます。

