

オブジェクト指向プログラミング 第3回

フレームの表示とグラフィックス クラスの継承

担当：高橋、佐藤聖也

先週の復習：クラスとインスタンス

先週の練習問題の解答例 (FrameSample02.java) を以下に示します。クラスとは「設計図のようなもの」であり、インスタンスとは「設計図からできる具体的なモノのようなもの」でした。フレームを複数表示させるためには、インスタンス (frA と frB)¹ を2つ生成し、それぞれのフレームを表示するためのメソッドを書きます。

また、

```
Frame frA = new Frame();
```

などで出てくる、右端の「Frame()」はコンストラクタといいます。新たなインスタンスを生成する際に呼び出されて内容の初期化などを行う、特殊なメソッドがコンストラクタです²。コンストラクタに引数が入ることもあります。

FrameSample02.java

```
import java.awt.*;
class FrameSample02{
    public static void main(String [] args){

        Frame frA = new Frame();
        Frame frB = new Frame();

        frA.setSize(500, 400);
        frA.setTitle("フレーム A");
        frA.setVisible(true);

        frB.setSize(300, 200);
        frB.setTitle("フレーム B");
        frB.setVisible(true);

        //このフレームの×ボタンは無効なので
        //Ctrl+C を押してコマンドプロンプトから強制終了します。
    }
}
```

一行目で「import java.awt.*;」とありますが、これは「java.awt パッケージに含まれるすべてのクラスをインポートする」という意味です。これに対して「import java.awt.Frame;」と書けば Frame クラスしかインポートされません。

¹先週も説明したとおり、frA と frB は正確には「Frame クラスのインスタンスのアドレスを格納する参照型変数 (ポインタ)」です。

²「コンストラクタ」≒「初期化用のメソッド」

1 クラスの継承

1.1 Frame クラスと JFrame クラス

Frame クラスを使ったプログラムである FrameSample01.java や FrameSample02.java では、フレームの×ボタンを押してもプログラムを終了させることができませんでした。これに対して、Frame クラスにいくつかのメソッドを追加するなどしてより高度な処理ができるようになった JFrame クラスでは、×ボタンを押してプログラムを終了させることができます。

JFrameSample01.java

```
import javax.swing.*;
class JFrameSample01{
    public static void main(String [] args){

        //JFrame クラスのインスタンスを生成
        JFrame fr = new JFrame();
        //フレームサイズの指定（横，縦）[ピクセル]
        fr.setSize(500, 400);
        //フレームタイトルの表示
        fr.setTitle("フレームのタイトルが書けます！");
        //×ボタンをおすとプログラムを終了します
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //フレームをディスプレイ画面に表示します
        fr.setVisible(true);
    }
}
```

「setDefaultCloseOperation(...)」は、JFrame クラスのメソッドで、×ボタンが押されたときにデフォルトで実行される処理を設定します。引数には次のうちの1つを指定する必要があります：DO_NOTHING_ON_CLOSE（何も行わない）/ HIDE_ON_CLOSE（終了動作があるとウィンドウを非表示にする）/ DISPOSE_ON_CLOSE（終了動作があるとウィンドウを破棄する）/ EXIT_ON_CLOSE（終了動作があるとアプリケーションを終了する）。これらは public static final int EXIT_ON_CLOSE のように定義されているクラスフィールドなので、JFrame. をつけて呼び出しています。

このように、元のクラスを拡張 (extend) して新しいクラスを作り出すことを継承 (inheritance) といいます。また、元のクラス（ここでは Frame クラス）のことをスーパークラス (super class)、親、先祖などといい、拡張された新しいクラス（ここでは JFrame クラス）のことをサブクラス (subclass)、子、子孫などといいます。

JFrame クラスは、AWT (java.awt パッケージ) ではなく Swing (javax.swing パッケージ) に含まれるクラスであることに注意してください³。Java には、AWT や Swing の他にも機能ごとに様々なパッケージが用意されています。代表的な標準パッケージを Table 1.1 に示します。

³Java では、ボタンやチェックボックスなどの GUI 用のオブジェクトのことをコンポーネント (components) といいます。特に、プラットフォーム (アプリケーションが動作する環境のこと。ここでは OS のこと) 固有の機能を利用するものを重量コンポーネント (heavyweight components) といい、Java のみによって記述されプラットフォーム固有の機能を利用していないものを軽量コンポーネント (lightweight components) といいます。AWT は重量コンポーネントであり、Swing は軽量コンポーネントです。

Table 1.1 代表的な標準パッケージ

パッケージ名	内容
java.lang	Java 言語機能の持つ基本機能．標準でインポートされる
java.awt	GUI やグラフィックに関する機能
java.awt.event	メニューやボタンが押された時の処理
java.io	入出力に関する機能
java.net	ネットワークに関する機能
java.sql	SQL データベースに関する機能
java.applet	アプレットに関する機能
java.text	テキスト，日付，時刻などの表示形式に関する機能
java.security	セキュリティに関するクラスで，公開鍵や証明書に関する機能
java.util	その他の機能
javax.swing	高度な GUI に関する機能

JFrame クラスのスーパークラスが Frame クラスであるのと同様に，Frame クラスにもスーパークラスが存在しています．もっとも基本的なクラスであるクラスである Object クラスから JFrame クラスまでの継承関係は以下のようになっています．

```

java.lang.Object
├ java.awt.Component
│   └ java.awt.Container
│       └ java.awt.Window
│           └ java.awt.Frame
│               └ javax.swing.JFrame

```

Figure 1.2 JFrame クラスの継承関係

1.2 クラスの継承とグラフィックス

ここからは，出来上がったフレームの中に絵を描いていきます．差し当たり，円をひとつ描くことを目標にしましょう．OvalSample01.java というプログラムに少しずつコードを書き加えながら，目標に近づいていくことにします．

OvalSample01.java

```

import java.awt.*;
import javax.swing.*;

public class OvalSample01{
    public static void main(String[] args) {

        // フレームを生成する
        JFrame fr = new JFrame();
        // パネルを生成する
        JPanel panel = new JPanel();

        //パネルのサイズを決める
        panel.setPreferredSize(new Dimension(500, 400));
        //Dimension dim = new Dimension(500, 400);
        //panel.setPreferredSize(dim);
        panel.setBackground(Color.white);

        // フレームを設定する
        fr.add(panel); //パネル panel をフレーム fr にはめ込む
        fr.pack();    //パネル panel の大きさに合わせてフレーム fr の大きさを決める
    }
}

```

```

        fr.setTitle("タイトルの表示");//タイトルの表示
        //×ボタンでプログラムを終了させる
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setResizable(true);//フレームサイズを可変(true) / 不変(false) にする
        fr.setVisible(true);//フレームを画面に表示させる
    }
}

```

OvalSample01.java では、フレームの中に絵を描くための「パネル」を生成し、
`fr.add(panel);`
 でパネルをフレームにはめ込んでいます⁴。フレームのサイズを決めるときには、これまで
`fr.setSize(500,400);`
 などとしていましたが、ここではパネルの大きさを先に決めています。すなわち：
`panel.setPreferredSize(new Dimension(500, 400));`
 とします⁵。setPreferredSize メソッドの引数には Dimension クラスのコンストラクタが入って
 いますが

```

        Dimension dim = new Dimension(500, 400);
        panel.setPreferredSize(dim);

```

と書きなおすこともできるでしょう。

```

        fr.pack();

```

は、パネルの大きさに合わせてフレームの大きさを適当に決めてくれるメソッドです。もしパネル
 がはめ込まれていなければ、最小サイズのフレームが表示されます (`fr.add();` の行をコメントア
 ウトすれば分かります)。また、

```

        panel.setBackground(Color.white);

```

によって、パネルの背景色を白色にしています。Color クラスは white, red, blue, ... などの既定
 の RGB を返すクラスフィールドを持っていますので、ここではそれを使っています。

```

        Color cw = new Color(255,255,255);

```

のように、RGB で指定された特定の色をインスタンスとして生成し、setBackground メソッドの
 引数とすることもできます。

Dimension クラスおよび Color クラスは AWT に含まれており、JFrame クラスと JPanel クラス
 は Swing に含まれているため、import java.awt.*; と import javax.swing が 1 行目と 2 行目に
 書かれています。

さて、この OvalSample01.java に、円を描くためのメソッドを付け加え、OvalSample02.java
 とします。

OvalSample02.java

```

import java.awt.*;
import javax.swing.*;
public class OvalSample02 extends JPanel{

    // 描画するときの処理
    // paintComponent メソッドのオーバーライド
    public void paintComponent(Graphics g){
        //javax.swing.JComponent クラスのメソッド
        super.paintComponent(g);
    }
}

```

⁴add メソッドはもともとは java.awt.Component クラスで定義されていますが、JFrame クラスは Component クラス
 を先祖とするクラスなので、JFrame クラスのインスタンスメソッドとして呼び出すことができます。

⁵setPreferredSize メソッドも、もともとは javax.swing.JComponent クラスで定義されているメソッドですが、JPanel
 クラスは JComponent クラスを先祖とするクラスなので、JPanel クラスのメソッドとして呼び出せます。

```

        g.setColor(Color.blue);
        g.fillOval(100,100,150,150);
    }

    public static void main(String[] args) {

        // フレームを生成する
        JFrame fr = new JFrame();
        // パネルを生成する
        //
        JPanel panel = new JPanel();
        OvalSample02 panel = new OvalSample02();

        //パネルサイズを決めるため
        panel.setPreferredSize(new Dimension(500, 400));
        //Dimension dim = new Dimension(500, 400);
        //panel.setPreferredSize(dim);
        panel.setBackground(Color.white);

        // フレームを設定する
        fr.add(panel); //パネルをフレームにはめ込む
        fr.pack(); //パネル panel の大きさに合わせてフレーム fr の大きさを決める
        fr.setTitle("円の描画"); //タイトルの表示
        // × ボタンでプログラムを終了させる
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setResizable(true); //フレームサイズを可変 (true) / 不変 (false) にする
        fr.setVisible(true); //フレームを画面に表示させる

    }
}

```

最初に、OvalSample02 の後ろに extends JPanel と加えられている点に注意してください。これは、「OvalSample02 クラスは JPanel クラスを拡張 (extend) したものである」という意味です。言い換えれば、OvalSample02 クラスは JPanel クラスを継承したことになります。

public void paintComponent(Graphics g){...} というのが、円 (や他の図形) を描くためのメソッドになります。このメソッドの中の最初の行で、super.paintComponent(g){...} としていますが、これは「super.」が「JPanel クラスより先祖側のクラス」を指し (最初に paintComponent メソッドを定義しているのは javax.swing.JComponent クラス)、そのスーパークラスの paintComponent メソッドを呼び出すとともに、g.setColor(Color.blue); で青色を設定し、g.fillOval(100,100,150,150); によって (100,100) を左上の角とする 50 × 50 ピクセルの矩形に内接する円を描くようになっています⁶。

しかし、この2点の変更だけでは、パネルに円を描かせることはできません。最後に、main メソッド内で

```

        JPanel panel = new JPanel();

```

となっていたインスタンス生成を

```

        OvalSample02 panel = new OvalSample02();

```

と書き換えます。OvalSample02 クラスは JPanel クラスを継承しているので、OvalSample01.java で書いた分は問題なく動作し、それに加えて、OvalSample02 クラスに追加されたメソッド (public void paintComponent(Graphics g){...}) が呼び出され、実行されます。この結果として青色の円が表示されるようになります。

⁶ このように、サブクラス (この場合 OvalSample02 クラス) にスーパークラス (この場合 JPanel クラス) と同じ引数パターンのメソッドを作成して別の処理内容に変えることをメソッドのオーバーライド (override) といいます。また、このメソッドの引数のパターンのことをシグニチャといいます。

課題

下の Figure 2.1 のように，色と大きさと位置がランダムな半透明の円を，30 個表示するプログラム `ManyOvalSample_00rd000.java` を書け．表示上の細かい差異は問わない．ただし，RGB 値を格納するために整数型の配列（配列サイズは 3）を，円を繰り返し描くために `for` 文を，円の位置や大きさ，RGB 値をランダムに決めるために `Math.random()` を用いること．また，プログラム名の `00rd000` は各自の学籍番号とする．

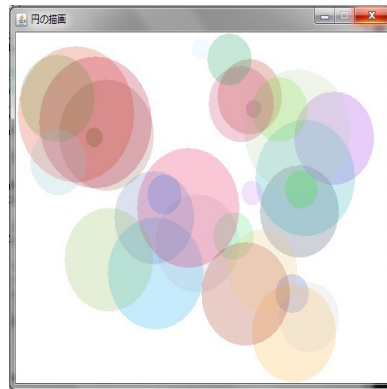


Figure 2.1

提出期限：5/31 23:59まで

提出場所：<https://tdu.app.box.com/f/2bb10a44a3d64d78b1209d32dc67293a>