

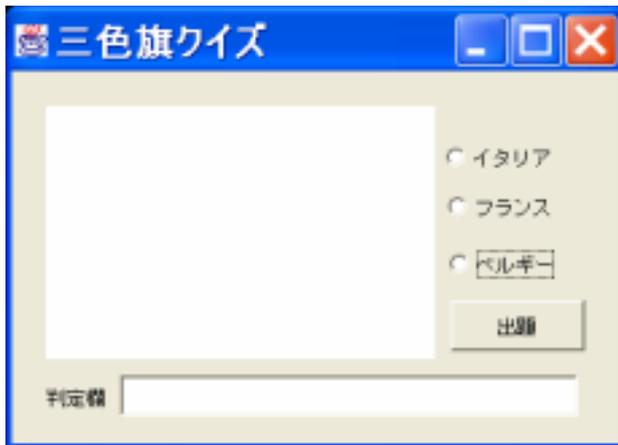
2003年度プログラミング応用編 - 課題

前回までと同様 HP に課題プログラムのクラスファイルを置いてあります。このクラスファイルをダウンロードすると、該当プログラムを実行し動作結果を確認できます。やり方は「クラスファイルからの実行の仕方」を参照して下さい。課題名横の()内に記してあるのがクラスフォルダの名前です。なお、課題名の右端に記してある数字は、解いた場合に成績に加点される得点です。

今回は、前回に引き続き CG の課題を扱いました。課題 を終了していることを前提に話を進めていますので、不明な点は適宜課題 の該当部分を参照して下さい。前半の 4-1～4-2 は CG の応用課題、そして 4-3～4-5 は画像ファイルの読み込みと表示を扱っています。

課題4 - 1 三色旗クイズ(flagquiz) 2

課題 3-8 の三色旗プログラムを発展させて、次のように、表示された三色旗がどの国のものをか答える三色旗クイズのプログラムを作成して下さい。



起動すると、左のような画面が現れます。



[出題] ボタンをクリックすると、(イタリア、フランス、ベルギーのいずれかの国の) 三色旗がランダムに表示されます。左はベルギーの国旗が表示された場合。



ここで、正解のベルギー欄をクリックすると、判定欄に「正解です。」と表示されます。

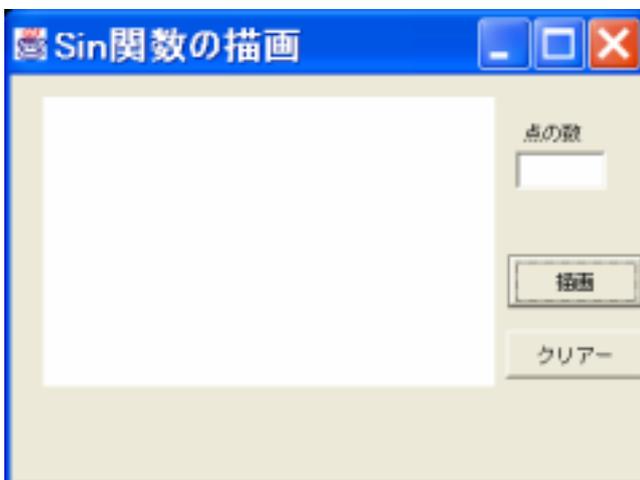


誤った国名欄、例えばイタリア欄をクリックすると、「残念！正解はベルギーです。」と表示されます。これは、フランス欄をクリックしても同様です。

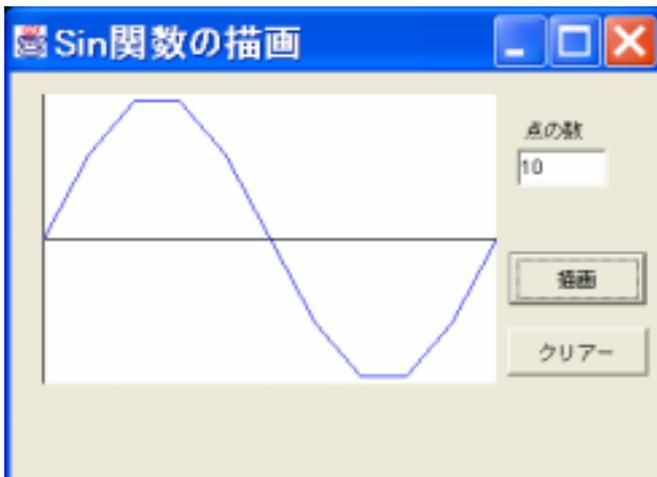
同様の処理を、イタリアおよびフランスの国旗が表示された場合についても行います。

課題4 - 2 グラフの描画 - sin 関数(graphsin) 3

課題 3-9 では、点画を描きました。ここでは、それを発展させて、次のように 1 周期分の sin 関数を表示するプログラムを作成しましょう。



プログラムを起動すると左のような画面が現れます。



($\sin(x)$ の値を求める) 点の数 n を入力して [描画] ボタンをクリックすると、1周期を n 等分した点 $\{x_1, x_2, x_i, \dots, x_n\}$ に対応する $\sin(x_i)$ を計算し、それらを直線で結ぶことによって、 $\sin(x)$ を描画します。

左は点数 10 の場合の例。



点の数を 20 にして [描画] ボタンをクリックすると、滑らかな曲線になります。

<ヒント>

x 座標の範囲は $0 \leq x < 2\pi$ 、 y 座標の範囲は $-1 \leq y \leq 1$ です。

円周率 π の値は、`Math.PI` で与えられます。

パネルコンポーネントの、横幅を w 、高さを h とすると、関数の原点座標 (x_0, y_0) は、 $x_0=0, y_0=h/2$ で与えられます。

x 軸と y 軸も描画して下さい。

$\sin(x)$ の値は `Math.sin(x)` で与えられます。

\sin 関数の描画は点の数を n とすると、点 $(x_i, \sin(x_i)) \ (i=1, 2, \dots, n)$ を求め、それらを順次線で結べばよいのです。

その際、 x 座標の範囲 2π がパネルコンポーネントの幅 w に対応し、 y 座標の範囲 2 がパネルコンポーネントの高さ h に対応するように、 x 座標、 y 座標それぞれについて縮尺率 c_x, c_y を求める必要があります (例えば $2\pi \times c_x = w$ より c_x が求まります)。

縮尺率を用いると、点 $(x_i, \sin(x_i))$ は、パネル上では、 $(x_0 + c_x \cdot x_i, y_0 - c_y \cdot \sin(x_i))$ に対応します。

なお、パネル上の座標は整数であることに注意して下さい。

画像ファイルの読み込みと表示

以下では、画像ファイルを読み込んで表示させる方法を学習しましょう。Java 言語では、{ gif,jpg,png } 形式の画像を読み込む機能があります。HP に掲載している Fig.lzh をダウンロードし、それを解凍して下さい。フォルダ Fig の中には{ Falcon.jpg, Grappa.jpg, Desk.jpg, Concert.jpg, Milano.jpg } の5つのサンプル画像を納めています。以下では、これを入力画像ファイルとして用います。以下の課題では、作成するプロジェクトのフォルダ内にフォルダ Fig を必ずコピーしておいて下さい。

まず、次のようなプログラムを作ってみましょう。



起動すると、左のような画面が現れます。



画像表示ボタンをクリックすると、左のような画像が現れます。

このプログラムは次のようになります。

```

void jButtonDraw_actionPerformed(ActionEvent e) {
    Graphics g=jPanell1.getGraphics();
    Image Fig1; //Image (画像) クラスのオブジェクトの宣言
    Toolkit toolkit=getToolkit(); //Toolkit クラスのオブジェクトの宣言
    Fig1=toolkit.getImage("Fig¥¥Falcon.jpg"); //指定したファイル内の画像
を Image オブジェクト Fig1 に読み込む
    g.drawImage(Fig1,0,0,this);
}

```

< 解説 >

Java 言語では、画像は Image オブジェクトとして扱われます。そして Image オブジェクト A を表示させるためには、次のように drawImage メソッドを用います。

```
drawImage(A,x,y,this)
```

ここに、(x,y)は表示させる画像オブジェクト(の左上隅)の表示位置です。第4引数については、通常は this と指定すると覚えておいて下さい。

画像ファイルから画像を読み込むためには、getImage メソッドを用います。このメソッドは getImage(ファイル名)の形で用います。上の例では、ファイル名として "Fig¥¥Falcon.jpg" と指定していますが、これは、Fig フォルダ内にある Falcon.jpg ファイルという意味です。本来は¥記号は一つで良いのですが、これは特別な意味を持つ記号なので、文字列定数の中で¥を表したい場合は¥¥の様に二つ続けて記述しなければなりません。1文字では、コマンドだと解釈されてしまうので注意して下さい。なお、使用しているフォントによっては、「¥」が「\」と表示されます。

getImage メソッドは Graphics クラスには用意されておらず、Toolkit クラスという別のクラスに用意されています。そこで、上のプログラムでは、

```
Toolkit toolkit=getToolkit();
```

によって、toolkit オブジェクトを用意し、そこに用意されている getImage メソッドを利用しています。

画像ファイルから読み込まれた画像データは、プログラム中では Image オブジェクトに代入して用います。上の例では、Fig1 という名前で Image オブジェクトを用意しています。

なお、用意した画像の大きさは、幅 390、高さ 250 程度なので、パネルの大きさもその程度の大きさにして下さい。パネルのサイズをこれより大きくすると、余白部分ができてしまいます。

プログラムを作成したら動作を確かめて下さい。

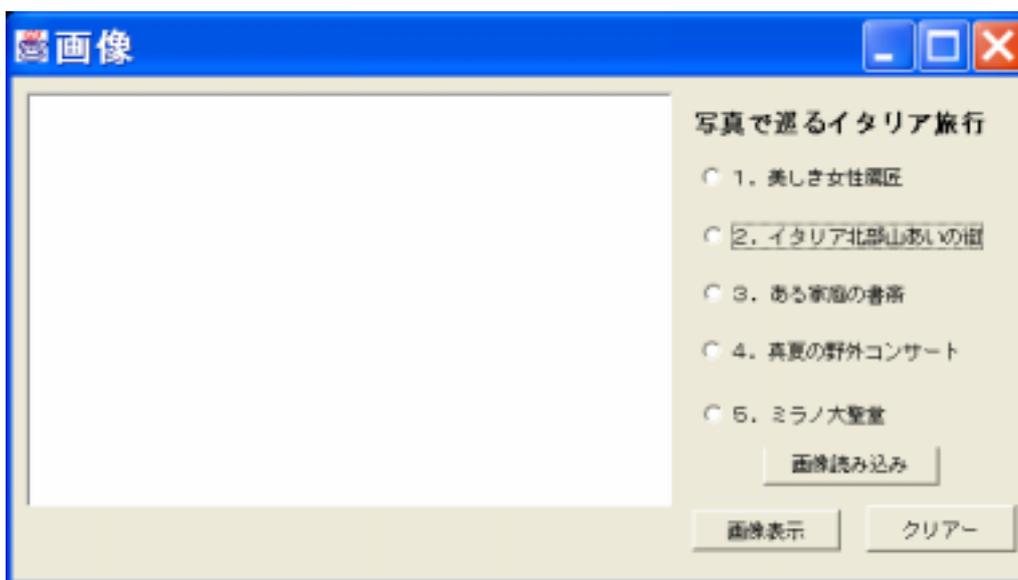
確かめてみると分かりますが、ボタンを1回押しただけでは画像は表示されず、2回目で表示されると思います。動作を見る限り、最初の drawImage メソッドが呼び出された段

階では画像をバッファ（一時的作業領域）に格納するだけでまだ（パネルに）表示せず、2回目に drawImage メソッドが呼び出されて始めて表示されるようになっているようです。

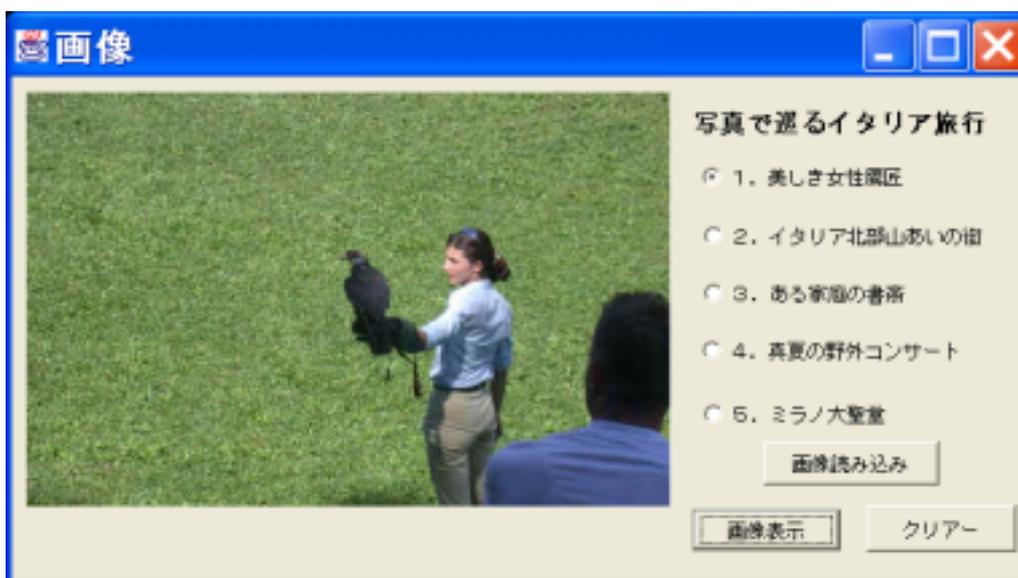
課題4 - 3 画像ファイルの読み込みと表示 写真で巡るイタリア旅行(image) 2

画像表示の技術を使って、次のように、指定した画像を表示させるプログラムを作成して下さい。

まず、プログラムを起動すると下の画面が現れます。



ここで、[画像読み込み] ボタンをクリックし、画像ファイルの画像データをバッファへ格納します。続いて画像リストの適当な欄をクリックして [画像表示] ボタンをクリックすると、該当する画像が表示されます。下は、1を選択したところ。



下は、5を選択したところ。



このプログラムでは、5つの画像を5つの Image オブジェクトに格納するようにします。そこで、下のように、Image オブジェクト (の配列) Fig[] をグローバル変数として宣言して下さい。点線枠内が入力部分です。

```
public class Frame1 extends JFrame {  
    private JPanel contentPane;  
    private JPanel jPanel1 = new JPanel();  
    . . .  
    Image Fig[]=new Image[5]; //Image オブジェクトを配列型で宣言  
    //フレームのビルド
```

次に、ボタン [画像読み込み] のイベントハンドラを次のように記述して下さい。

```
void jButtonRead_actionPerformed(ActionEvent e) {  
    Graphics g=jPanel1.getGraphics();  
    int Xmax=jPanel1.getWidth();  
    int Ymax=jPanel1.getHeight();  
    Toolkit toolkit=getToolkit();  
    Fig[0]=toolkit.getImage("Fig¥¥Falcon.jpg");  
    Fig[1]=toolkit.getImage("Fig¥¥Grappa.jpg");  
    Fig[2]=toolkit.getImage("Fig¥¥Desk.jpg");  
    Fig[3]=toolkit.getImage("Fig¥¥Concert.jpg");  
    Fig[4]=toolkit.getImage("Fig¥¥Milano.jpg");  
    for (int i=0;i<=4;i++) {  
        g.drawImage(Fig[i],0,0,Xmax,Ymax,this);  
    }  
}
```

< 解説 >

プログラムを見れば分かるように、このプログラムで行っている事は以下の2点です。

) Image オブジェクト Fig[0] ~ Fig[4] の定義 (画像データの代入)

) drawImage メソッドの呼び出しによる画像オブジェクトのバッファへの保管

その際、drawImage メソッドの引数が先程と変わっている事に気づいたでしょう。実は、drawImage メソッドは

```
drawImage(画像オブジェクト、x,y,w,h,this)
```

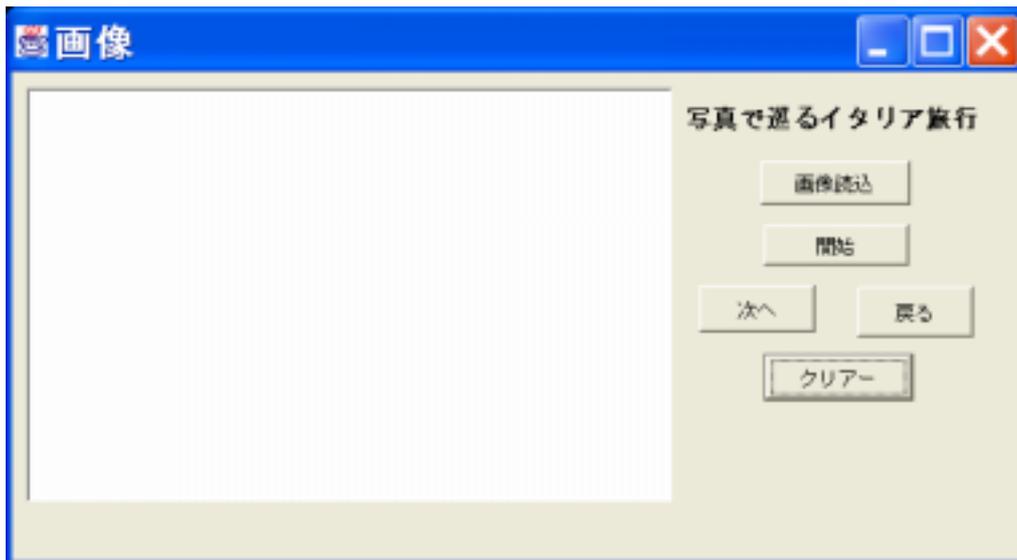
という形式で、指定した画像オブジェクトを、幅 w、高さ h の領域に収まるように (縮尺して) 表示させる、という事もできます。これは大変便利なのですが、本来の画像の縦横サイズ比と指定領域のそれが大きくずれてしまうと、画像がひずんでしまうので注意が必要です。

[画像表示] ボタンの処理は、問題なく記述できるはずですが、なお、このプログラムを実行するには、プロジェクトのフォルダ内にフォルダ Fig をコピーしておくことを忘れないように注意して下さい。

課題4 - 4 写真で巡るイタリア旅行2 - スライドショー (image2) 3

課題5 - 3 を改良して、次のように5枚の画像を (順番に) 次々と表示するプログラムを作成して下さい。

プログラムを起動すると、下のような画面が現れます。



ここで、まず [画像読み込み] ボタンをクリックし、画像データをバッファに格納します。

続いて[開始]ボタンをクリックすると、最初の画像(課題5-3の1の画像)が表示されます。



[次へ]ボタンをクリックすると、2番目の画像が表示されます。



以下、[次へ]ボタンをクリックすると、順番に次の画像が表示されます。

そして5番目の画像が表示された状態で[次へ]ボタンをクリックすると・・・



下のように最初の画像に戻ります。



このように、循環的に画像を表示できるようにします。[戻る]ボタンについても同様に、循環的に一つ前の画像を表示するようにして下さい。

課題4 - 5 キャプションの表示・消去(Image3) 2

課題5 - 4を改良して、次のように画像の説明(キャプション)を表示させるプログラムを作成して下さい。

前課題同様、プログラムを起動して[画像読み込] [開始ボタン]とクリックすることで最初の画像が現れます。



ここで、[キャプション表示]ボタンをクリックすると、画像の説明が表示されます。



また、[キャプション消去] ボタンをクリックすると、キャプションは消えます。



このように、キャプションの表示 / 消去を全ての画像について行うようにして下さい。
各画像のキャプションは次の通りとします。

画像の順番	キャプション
1	古都シエナに伝わる鷹匠の技 - 美しき鷹匠
2	イタリア北部の街 - バッサーノ・デル・グラッパ
3	書斎 - イタリア人はアンティークな家具がお好き
4	真夏の夜のコンサート - 古都ペルージャにて
5	ミラノ大聖堂 - 青空に映える大聖堂の尖塔

<ヒント>

パネル上の Graphics クラスのオブジェクトを g とするとパネル上の座標(x,y)に文字列を表示させるには次のようにします (文字列"abc"を表示させる例です)。

```
String Moji="abc";
```

```
g.drawString(Moji,x,y);
```

ただし、この(x,y)は表示させる文字列の左下隅の座標です。

表示させる文字列のフォントを指定するには次のようにします。

```
int Fsize=14; //フォントサイズの指定
```

```
g.setFont(new Font("Dialog",Font.PLAIN,Fsize));
```

これは、フォントとして Dialog フォントを選び、通常の文字スタイルを選んだ場合です。フォントの種類と文字スタイルは次の通りです。

フォント名	Dialog、DialogInput、Monospaced、Serif、SansSerif
フォントスタイル	Font.PLAIN、Font.BOLD、Font.ITALIC

スタイルを組み合わせる場合は「Font.BOLD|Font.ITALIC」などとします。