独習Java第3版

13.1 アプレットの概要
13.2 最初の Java アプレット
13.3 アプレットのライフサイクル
13.4 Graphics クラス

アプレットの概要(1/3)

- WebページのHTMLソースコードから参照されるプログラム
- Webサーバーからブラウザに動的にダウンロードされる
- ダウンロードされたアプレットはブラウザの環境で実行される
- アプレットビューアなどのツールで実行することもできる

アプレットの概要(2/3)

- アプレットにはJavaアプリケーションと同等の機能を持たない
- 動作範囲はサンドボックスの中に限られる
  - 信頼されないコードは一定の境界の外では動作できない（例：ローカルディスクに対しての読み書き）
- アプレットはネイティブコードを実行できない
  - 実行できるとサンドボックスによる制限の裏をかけるため
- アプレットからローカルコンピュータのアプリケーションを起動することはできない

アプレットの概要(3/3)

- アプレットからはダウンロードもとのホストにのみソケット接続を開くことが出来る
  - →クラッカーの進入を防ぐため
- デジタル署名をアプレットに関連付けることでアプレットの制限を緩和できる

Javaアプレットの例

```
import java.applet.Applet;
import java.awt.Graphics;

/*
  <applet code="FirstApplet" width=200 height=200>
  </applet>
*/
public class FirstApplet extends Applet {
  public void paint(Graphics g) {
    g.drawString("This is my first applet!", 20, 100);
  }
}
```

全てのアプレットのスーパークラスである
Appletクラスをインポート

Graphicsクラスを使ってアプレットウィンドウに
文字列や線などを描画する

アプレットビューアで読み取るHTMLコード

Appletクラスを拡張しなければいけない

Graphicsオブジェクトが引数
アプレットの出力を生成するのが目的

文字列を表示させる
引数：文字列、座標（横）、座標（縦）

実行結果

- アプレットビューアを用いて実行
  - コマンド:
    appletviewer FirstApplet.java
  - アプレットビューアは
    JDKに属しているツール
- 線を描画する際は
  drawLine()メソッドを使う
  - 線の起点、終点を示す4つの
    int型変数を引数として受け取る

アプレットのライフサイクル

- アプレットはWebブラウザまたは
  アプレットビューアなどのツールが用意された
  環境で実行され、main()メソッドが存在しない

- 代わりにinit(),start(),stop(),destroy()の
  ４つのメソッドが呼び出される

- これらはjava.applet.Appletクラスに
  定義されている

呼び出されるタイミング

メソッド	呼び出されるタイミング
init()メソッド	アプレットの実行が開始されるときだけ
start()メソッド	・init()メソッドの実行が終わった後 ・アプレットの実行を再開するとき ・Webページに別のページが表示され、 元のページに戻るとき ・アプレットビューアを最小化してから最大化するとき
stop()メソッド	・アプレットの実行を中断するとき アプレットビューアを最小化するとき ・Webページに別のページが表示されたとき ・destory()が呼び出される前
destroy()メソッド	アプレットが終了する直前

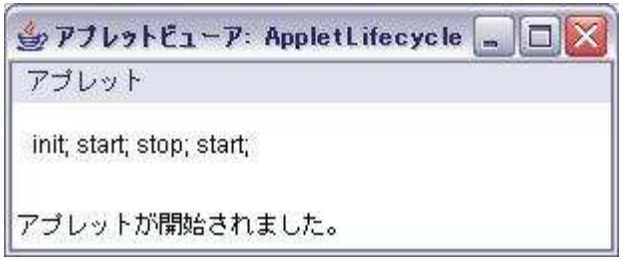
例

```
import java.applet.Applet;
import java.awt.Graphics;
/*
  <applet code="AppletLifecycle"
      width=300 height=50>
  </applet>
*/

public class AppletLifecycle extends Applet
     {
  String str = "";
  public void init() {
    str += "init; ";
  }
  public void start() {
    str += "start; ";
  }

  public void stop() {
    str += "stop; ";
  }

  public void destroy() {
    System.out.println("destroy");
  }

  public void paint(Graphics g) {
    g.drawString(str, 10, 25);
  }
}
```

実行結果

- 実行時

- 最小化→元のサイズに戻す の後



- 終了時
  - C:¥jv>Appletviewer AppletLifecycle.java
    destroy

Graphicsクラス

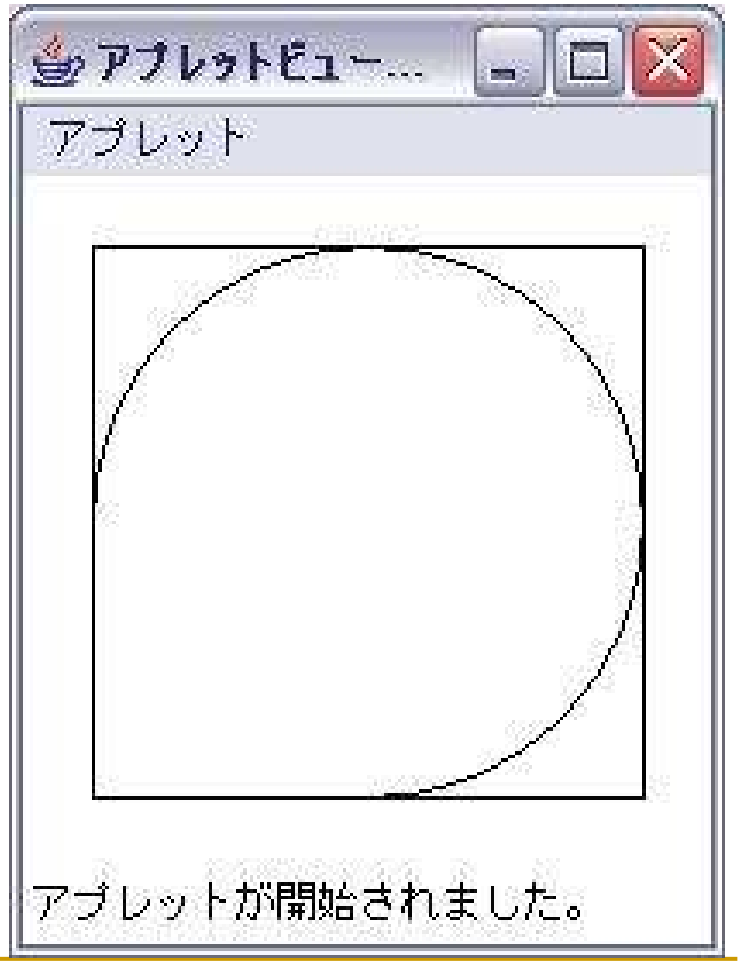
- Graphicsオブジェクトを出力するメソッド群がカプセル化されている

メソッド	説明
abstract void drawArc(int x, int y, int w, int h, int degrees0, int degrees 1)	degrees0とdegrees1の角度で弧を描画する。この中心は座標x,yに左上隅が配置される幅w、高さhの四角形の中心となる。角度は逆時計回りに増加し、0度は時計の午後3時に相当する
abstract boolean drawImage(Image img, int x, int y, ImageObserver io)	座標x,yに左上隅が配置されるようにイメージimgを描画する。描画処理の進行状況は、ioに送られる
abstract void drawLine(int x0, int y0, int x1, int y1)	座標x0,y0とx1,y1を結ぶ線を描画する
abstract void drawOval(intx, int y, int w, int h)	円を描画する。円の中心は、座標x,yに左上隅が配置される幅w,高さhの四角形の中心となる
abstract void drawPolygon(int x[], int y[], int n)	n個の頂点を持つ多角形を描画する。頂点の座標は、配列xとyの要素として引き渡す。最初の頂点と最後の頂点は、自動的に接続される。
abstract void drawPolyline(int x[], int y[], int n)	n個の頂点を持つ多角線（ポリライン）を描画する。頂点の座標は配列xとyの要素として引き渡す
void drawRect(int x, int y, int w, int h)	座標x,yに左上隅が配置される幅w,高さhの四角形を描画する。

メソッド	説明
abstract void drawString(String str, int x, int y)	strを座標x,yに描画する
abstract void fillArc(int x,int y,int w, int h, int degree0,int degree1)	degree0とdegree1の角度で弧を塗りつぶして描画する。弧の中心は、座標x,yに左上隅が配置される幅w, 高さhの四角形の中心となる。0度は時計の午後3時に相当する
abstract void fillOval(int x,int y,int w, int h)	円を塗りつぶして描画する。円の中心は、座標x,yに左上隅が配置される幅w,高さhの四角形の中心となる
abstract void fillPolygon(int x[],int y[], int n)	n個の頂点を持つ多角形を塗りつぶして描画する。頂点の座標は、配列xとyの要素として引き渡す。
void fillRect(int x,int y,int w,int h)	座標x,yに左上隅が配置される幅w,高さhの四角形を塗りつぶして描画する。
abstract Color getColor()	現在のオブジェクトのカラーを取得する。
abstract Font getFont()	現在のオブジェクトのフォントを取得する。
abstract FontMetrics getFontMetrics()	現在のオブジェクトのフォントメトリックスを取得する。
abstract void setColor(Color c)	グラフィックコンテキストの現在のカラーとしてcを設定する。
abstract void getFont(Font f)	現在のオブジェクトのフォントとしてfを設定する。

例

```
import java.applet.Applet ;
import java.awt.Graphics ;
/*
  <applet code="DrawTest" width=200 height=200>
  </applet>
*/
public class DrawTest extends Applet {
  public void paint(Graphics g) {
    int x[] = { 20, 20, 180, 180 } ;
    int y[] = { 180, 20, 20, 180 } ;

    g.drawArc(20, 20, 160, 160, -90, 270) ;
    g.drawPolygon(x, y, 4) ;
  }
}
```



課題

- 右の図のような
  渦を描くプログラムを
  作りなさい