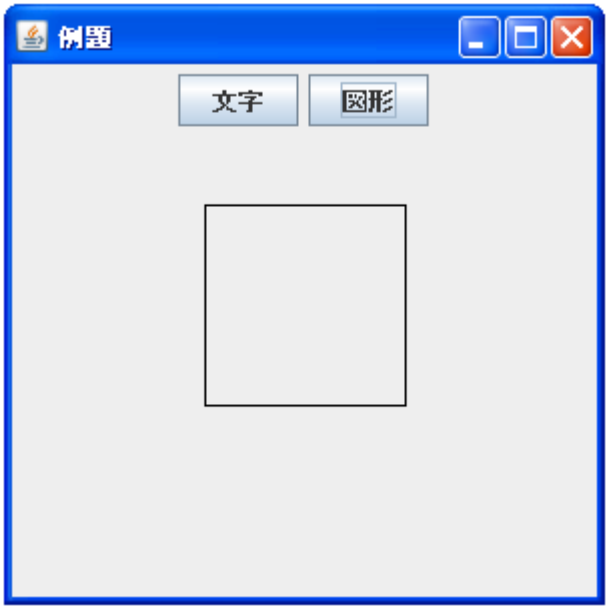
例題

「文字」というボタンをクリックすると「Hello Java !!」を表示し、「図形」というボタンをクリックすると正方形が表示されるプログラムを作りなさい。



Java のプログラムは一般に java コマンドを入力しないと実行できないが、jar ファイルを作成することによって、ダブルクリックで実行させることができるようになる。

テキストファイルに 「Main-Class: メインメソッドがあるクラス名」 と 1 行記述し、改行した後に manifest.mf で保存する。

コマンドプロンプトから以下のコマンドを入力する。

```
> jar cvmf manifest.mf Reidai.jar *.class
```

作成された 

をダブルクリックするとプログラムが実行される。

プログラム例
太字はウィンドウ作成に必須の文

```
import java.awt.*;                          //awt パッケージのインポート
import java.awt.event.*;                    //event パッケージのインポート
import javax.swing.*;                       //swing パッケージのインポート
                                            //JFrame の継承と ActionListener の実装
public class Reidai extends JFrame implements ActionListener {
   JButton btn1, btn2;                      //ボタンのオブジェクト宣言
   JLabel lbl;                              //ラベルのオブジェクト宣言
   JPanel panel;                            //パネルのオブジェクト宣言
   String action = "";                      //ボタンイベント判定用の文字列宣言

   Reidai() {                               //コンストラクタの作成
      setTitle("例題");                     //ウィンドウのタイトル
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  //プログラム終了の組込み
      setSize(300, 300);                    //ウィンドウの大きさ設定
      setLocationRelativeTo(null);          //プログラム開始時に画面中央に表示
      setLayout(new BorderLayout());        //ウィンドウレイアウトの設定
      panel = new JPanel();                 //パネルの生成
      lbl = new JLabel();                   //ラベルの生成
      btn1 = new JButton("文字");           //ボタンの生成
      btn2 = new JButton("図形");
      btn1.addActionListener(this);         //ボタンへアクションリスナー付設
      btn2.addActionListener(this);
      panel.add(btn1);                      //ボタンをパネルへ設置
      panel.add(btn2);
      Container cont = getContentPane();     //コンテントペインの取得
      cont.add(panel, BorderLayout.NORTH);   //パネルをコンテントペイン上部へ設置
      cont.add(lbl, BorderLayout.CENTER);    //ラベルをコンテントペイン中央へ設置
      lbl.setFont(new Font("MS ゴシック", Font.PLAIN, 32));  //フォント設定
      lbl.setHorizontalAlignment(JLabel.CENTER);             //センタリング
   }

   public static void main(String[] args) {       //メインメソッドの作成
      Reidai r = new Reidai();                     //クラスのインスタンス化
      r.setVisible(true);                          //ウィンドウの表示
   }

   public void actionPerformed(ActionEvent e) {   //アクションイベントの定義
      action = e.getActionCommand();               //クリックボタンの文字列取得
      if(action.equals("文字")) {                  //ボタン文字列の判定
         lbl.setText("Hello Java !!");             //ラベルへの文字列設定
      }
      else if(action.equals("図形")) {
         lbl.setText("");
         repaint();                                //再描画
      }
   }

   public void paint(Graphics g) {             //paint メソッドのオーバーライド
      super.paint(g);                          //スーパークラスのメソッド実行
      if(action.equals("図形"))
         g.drawRect(100,100,100,100);          //図形描画
   }
}
```

問題１

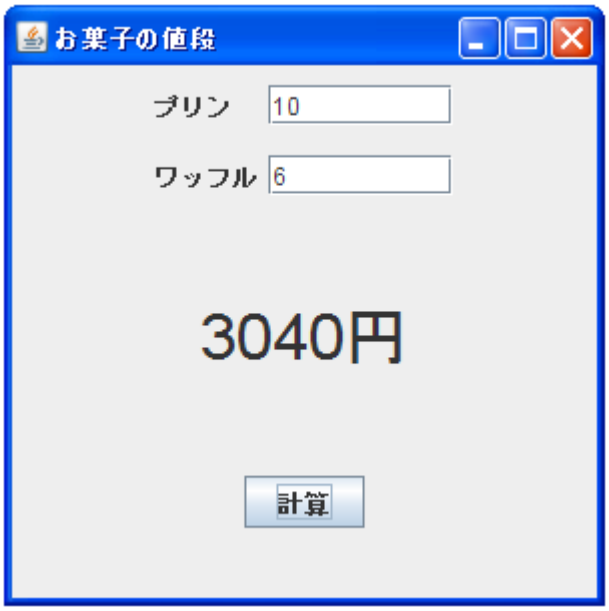
ボタンをクリックするとボタン文字の英語訳が表示されるプログラムを作りなさい。

問題２

ボタンをクリックするとテキストボックスに入力された文字列が表示されるプログラムを作りなさい。

問題３

プリンは１つ 220 円、ワッフルは１つ 140 円である。テキストボックスにそれぞれの個数を入力してボタンをクリックすると合計金額が表示されるプログラムを作りなさい。



問題４

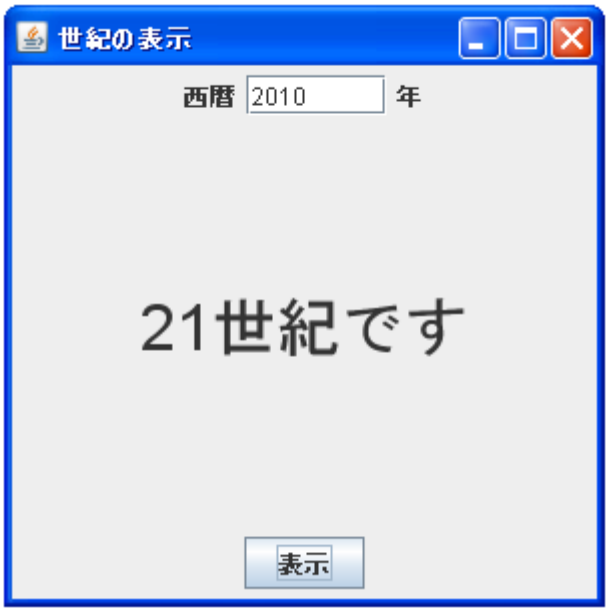
小児運賃は普通運賃の半額で 10 円未満に端数が出たときは切り上げられる。テキストボックスに普通運賃を入力してボタンをクリックすると小児運賃が表示されるプログラムを作りなさい。

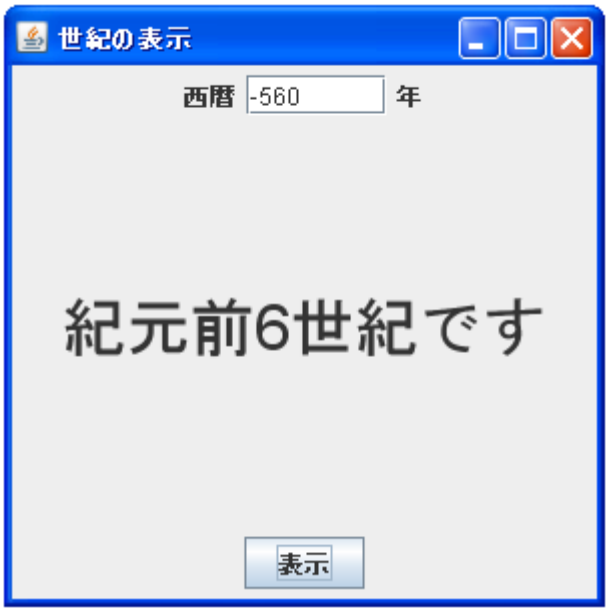
問題５

テキストボックスにテストの点数を入力してボタンをクリックすると 60 点未満で「不合格です」、60 点以上で「合格です」と表示されるプログラムを作りなさい。

問題６

テキストボックスに西暦年を入力してボタンをクリックすると対応する世紀が表示されるプログラムを作りなさい。西暦年にマイナスを付けて入力したときは紀元前が付記されるようにしなさい。



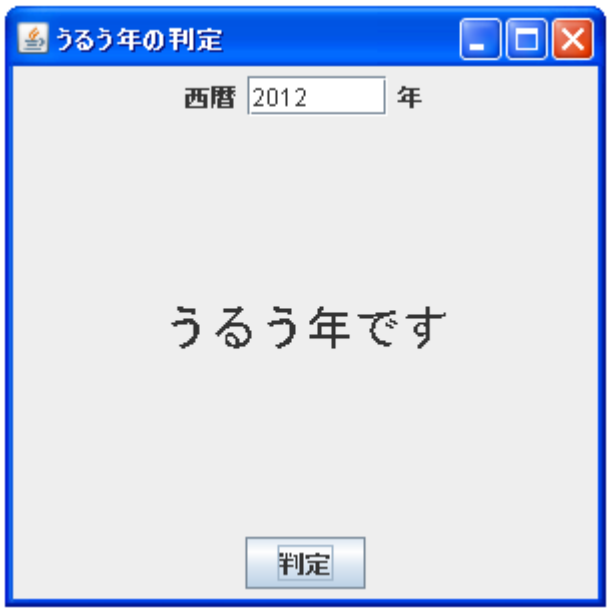


問題７

テキストボックスに西暦年を入力してボタンをクリックするとうるう年かどうかが表示されるプログラムを作りなさい。

（ヒント）うるう年は以下の条件で判定できる。

4 で割り切れて 100 で割り切れない　または　400 で割り切れる。

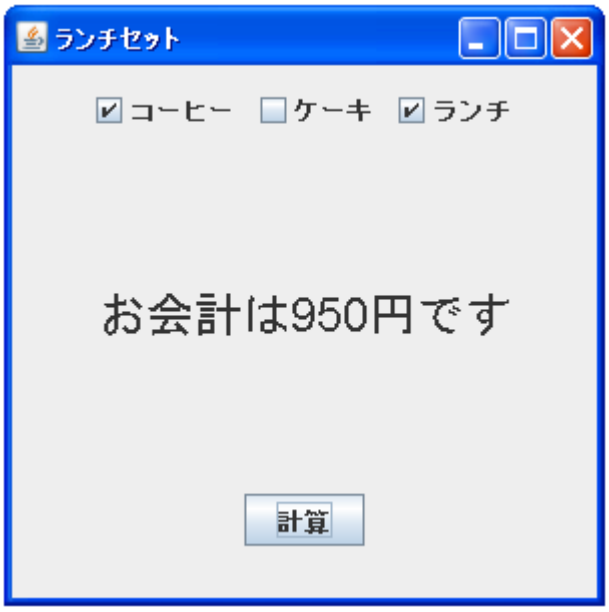


問題８

テキストボックスに 1 から 12 までの月を入力してボタンをクリックすると、3～5 月は「春です」、6～8 月は「夏です」、9～11 月は「秋です」、12～2 月は「冬です」と表示されるプログラムを作りなさい。

問題９

コーヒーは 300 円、ケーキは 500 円であるが、セットで注文すると 650 円になる。また、850 円のランチを注文すると、ケーキ、コーヒー、ケーキセットがそれぞれ 200 円引きになる。チェックボックスでオーダーした後、ボタンをクリックすると支払う金額が表示されるプログラムを作りなさい。



問題１０

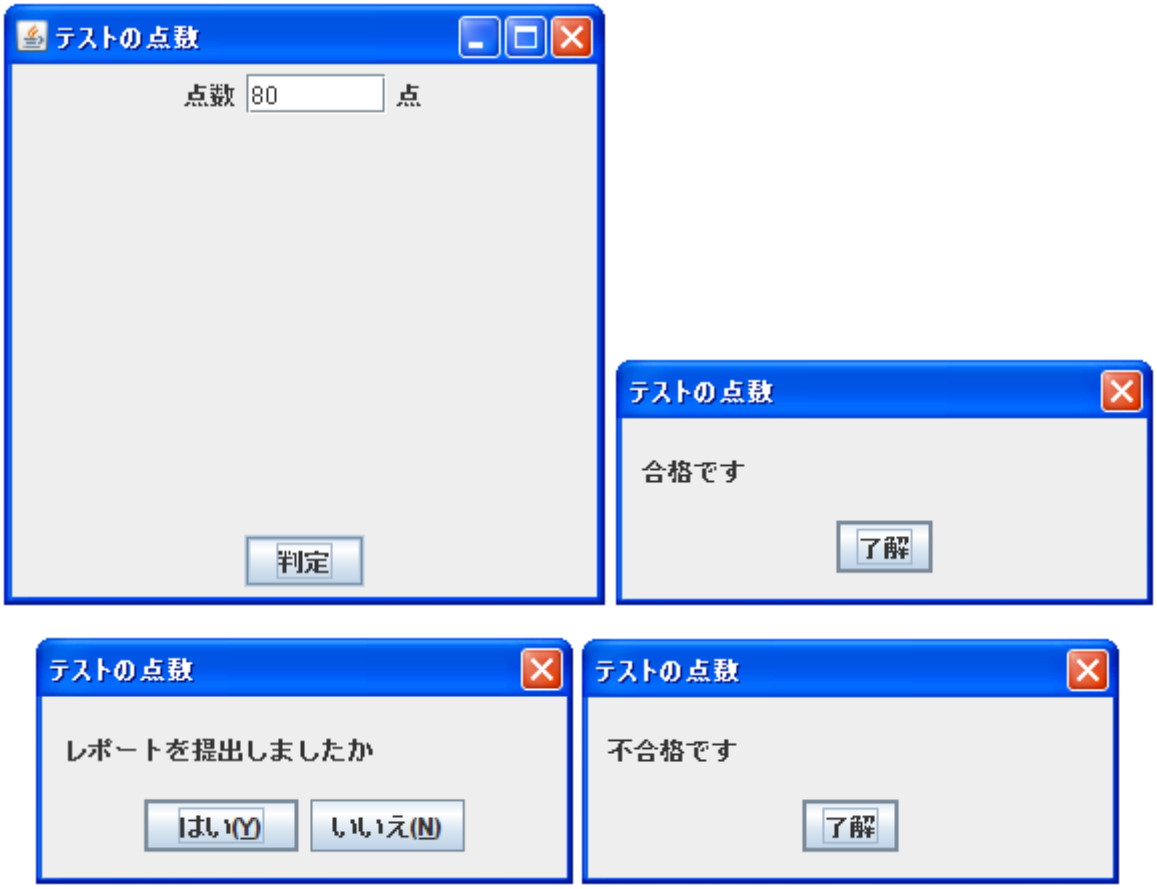
ラジオボタンで文字を選択してボタンをクリックすると文字の英語訳が表示されるプログラムを作りなさい。

問題１１

色と言語をそれぞれラジオボタンで選択してボタンをクリックすると色の各国語訳が表示されるプログラムを作りなさい。

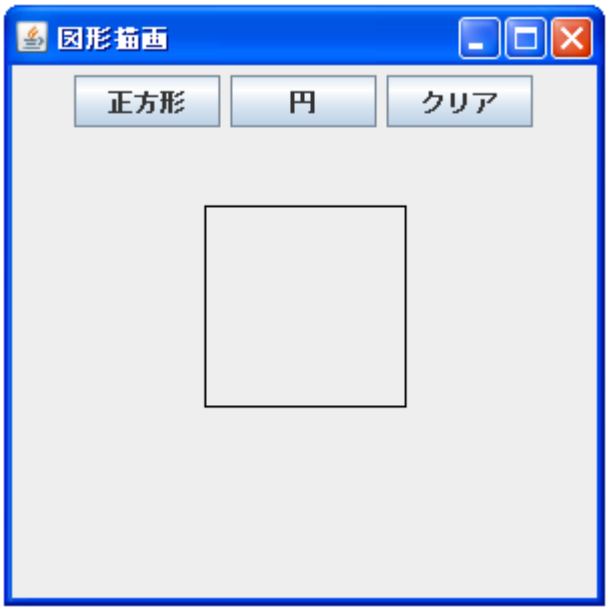
問題１２

テキストボックスにテストの点数を入力してボタンをクリックすると合格か不合格かが判定されるプログラムを作りなさい。ここで、70 点以上は合格、50 点未満は不合格であるが、50 点以上 70 点未満の場合はレポート提出の確認メッセージが現れて、レポートを提出していれば合格となる。



問題１３

「正方形」というボタンをクリックすると正方形を、「円」というボタンをクリックすると円をそれぞれ表示し、「クリア」というボタンをクリックすると図形が消去されるプログラムを作りなさい。

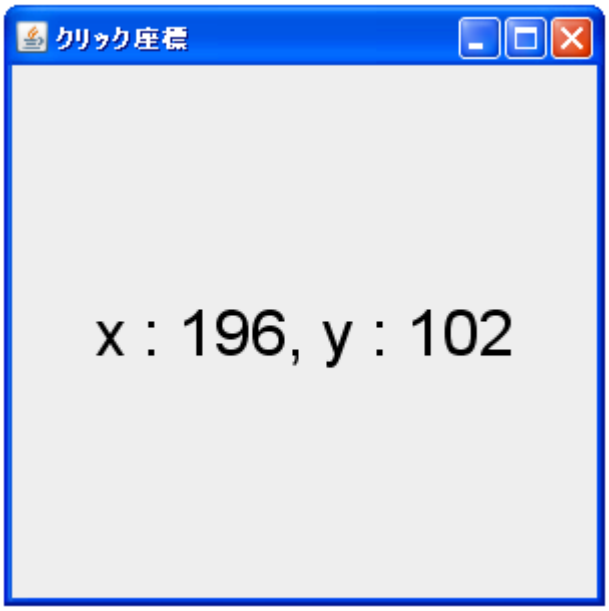


問題１４

「表示」というボタンをクリックすると画像が現れ、「消去」というボタンをクリックすると画像が消去されるプログラムを作りなさい。

問題１５

マウスでクリックした位置をウィンドウ座標で表示するプログラムを作りなさい。ここで、ウィンドウの下半分をクリックしたときは文字の色が変わるようにしなさい。



問題１６

スタートボタンをクリックすると１秒ごとに数字が増加していき、ストップボタンをクリックすると止まるプログラムを作りなさい。また、リセットボタンをクリックすると数字が 0 となるようにしなさい。

問題１７

コンボボックスで西暦年と月を選択してボタンをクリックすると、その月のカレンダーが表示されるプログラムを作りなさい。

（ヒント１）y 年 m 月 1 日の曜日は次式から求められる。

```
bias = y + (y - 1) / 4 - (y - 1) / 100 + (y - 1) / 400
```

s = 1 月から m - 1 月までの日数　として

week = (bias + s) % 7　を計算する。

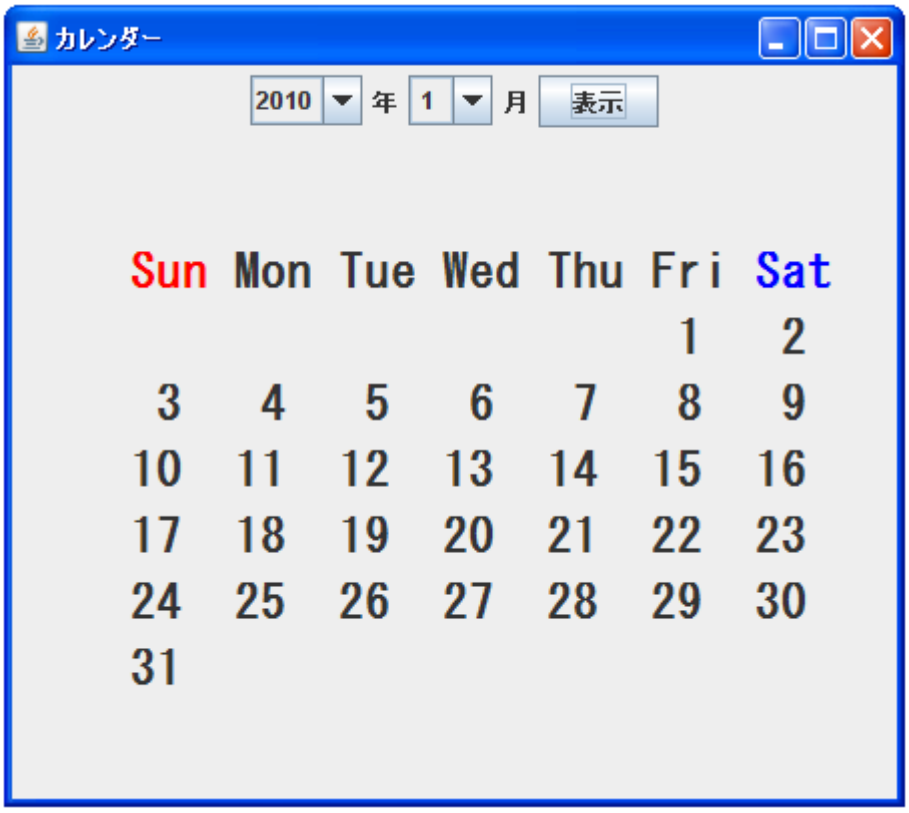
week = 0 ならば日曜で、6 ならば土曜である。

（ヒント２）カレンダーの数字列はひとつの文字列変数で構成し、その文字列をラベルで表示する。

文字列は<html><pre>で始め、</pre></html>で終わるようにする。

改行は
タグを使う。

<font color='　'>タグを使えば、Sun を赤で、Sat を青で表示させることもできる。



問題１８

円がウィンドウの左から右へ水平に移動するプログラムを作りなさい。ウィンドウの右側で円が消えたとき、再び左側から円が現れ、移動が無限に続くようにしなさい。

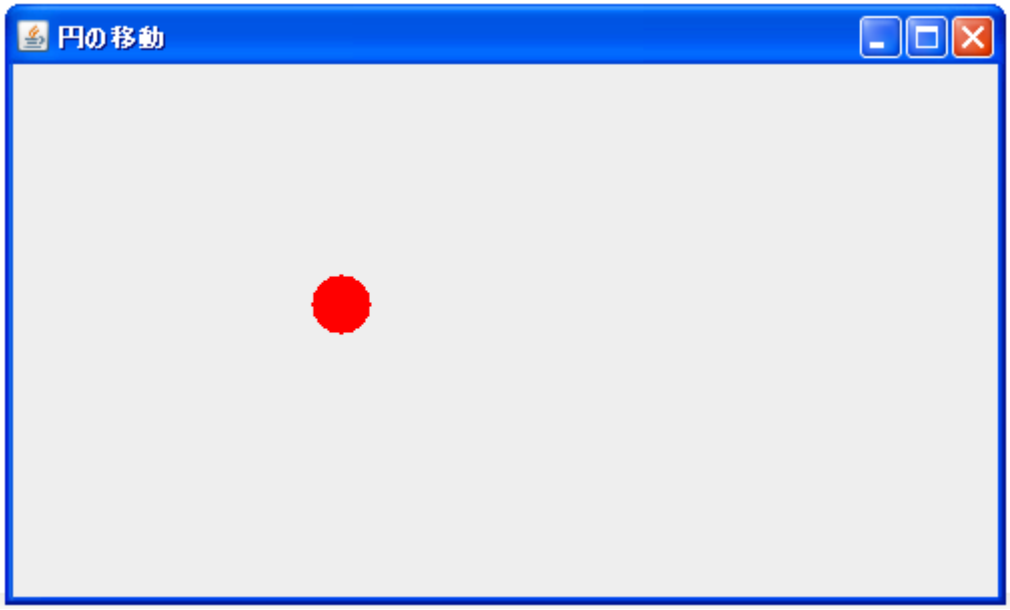
（ヒント１）`Circle` クラスを以下の要領で作成しなさい。

フィールド変数：ウィンドウの幅と高さ（`int width, int height`）、
円の半径（`int radius`）、円の中心の位置（`int x, int y`）、
移動距離（`int dx`）、速度（`int speed`）、円の色（`Color color`）

コンストラクタ：インスタンス化のときに `radius` と `color` を設定する。

メソッド：位置の設定（`void setXY(int x, int y)`）
移動距離の設定（`void setMove(int dx)`）
ウィンドウの大きさの設定（`void setCage(int width, int height)`）
速度の設定（`void setSpeed(int speed)`）
スレッドスリープタイムの計算（`void getSleeptime()`）　速度から計算する
円の描画（`void paint(Graphics g)`）
円の移動（`void nextPosition()`）　円が消えた後の処理を含める

（ヒント２）スレッド処理を含んだ `Problem18` クラスを添付するので利用すること。

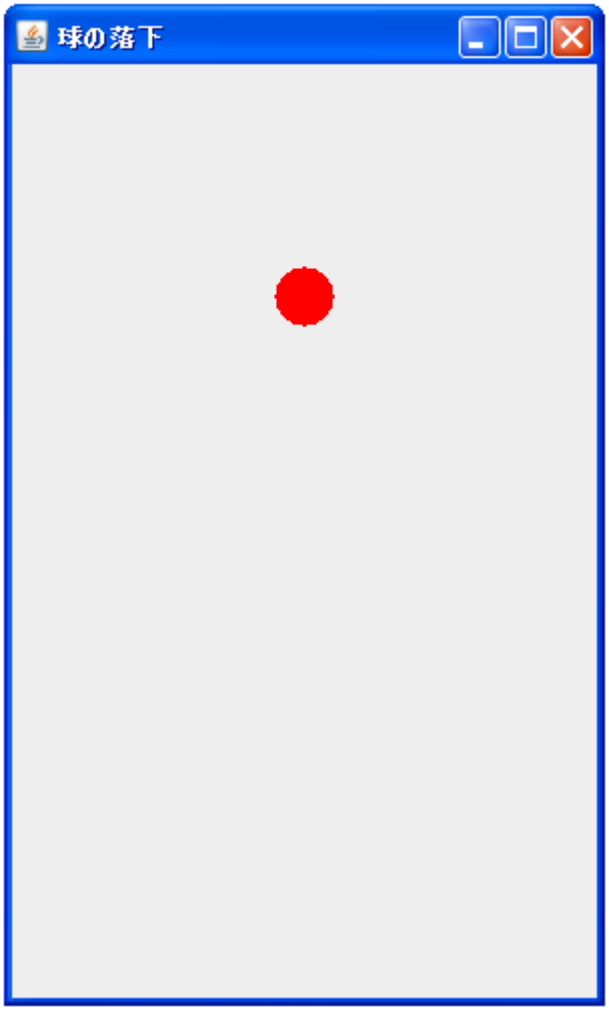


```
import java.awt.*;
import javax.swing.*;

public class Problem18 extends JFrame {
    static final int WIDTH = 500;                       //ウィンドウの幅
    static final int HEIGHT = 300;                      //ウィンドウの高さ
    static final int OBJECTS = 1;                       //円オブジェクトの数
    MyThread[] mythread = new MyThread[OBJECTS];        //スレッドを入れる配列
    Circle[] circle = new Circle[OBJECTS];              //円オブジェクトを入れる配列

    Problem18() {
        int radius = 15;                                //円の半径
        circle[0] = new Circle(radius, Color.red);      //円オブジェクトの生成
        circle[0].setXY(0, HEIGHT / 2);                 //初期位置
        circle[0].setMove(8);                           //オブジェクトの移動距離
        circle[0].setSpeed(60);                         //オブジェクトの速度
        circle[0].setCage(WIDTH, HEIGHT);               //ウィンドウの大きさの設定
        for(int i = 0; i < OBJECTS; i++) {      //オブジェクトごとにスレッドを作成
            mythread[i] = new MyThread(i, circle[i].getSleeptime());
            mythread[i].start();                        //スレッドの開始
        }
        setTitle("円の移動");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(WIDTH, HEIGHT);
        setLocationRelativeTo(null);
    }

    public static void main(String[] args) {
        Problem18 p = new Problem18();
        p.setVisible(true);
    }

    public void paint(Graphics g) {
        super.paint(g);
        for(int i = 0; i < OBJECTS; i++) {      //すべての円オブジェクトを描画
            circle[i].paint(g);
        }
    }
                                                //Threadクラスを継承した
    class MyThread extends Thread {             //内部（インナー）クラスを作成
        int index, sleeptime;                   //スレッドの識別子とスリープ時間

        MyThread(int index, int sleeptime) {    //インスタンス化の時にメンバー変数設定
            this.index = index;
            this.sleeptime = sleeptime;
        }

        public void run() {                             //runメソッドのオーバーライド
            Thread thisThread = Thread.currentThread(); //実行中のスレッドを取得
            while(mythread[index] == thisThread) {      //スレッドの識別
                circle[index].nextPosition();           //円オブジェクトの移動
                repaint();                              //再描画
                try {
                    this.sleep(sleeptime);              //スリープ処理
                } catch (InterruptedException e) { }
            }
        }
    }
}
```

コンパイルするとインナークラス部分は **Problem18$MyThread.class** として出力される。

問題１９

ある高さから垂直に落下した球が、下端で跳ね返って上向きに運動する様子をシミュレートするプログラムを作りなさい。ただし、落下も上向き運動も等速度運動とし、落下開始高さと下端との間で運動が無限に続くものとしなさい。



問題２０

　ウィンドウ内で複数の球が衝突を繰り返す様子をシミュレートするプログラムを作りなさい。ただし、衝突は球と壁のみを考慮し、球同士の衝突は考えないものとしなさい。また、球の初期位置、初期運動方向、運動速度は球ごとに異なる値を与えなさい。