グラフの描画

プログラミング演習 I
L03

今週の目標

- キャンバスを使って思ったような図（指定された線＝グラフ）を描いてみる

- 今週は発展問題が三つあります

グラフの準備 ～値の算出～

値の表示

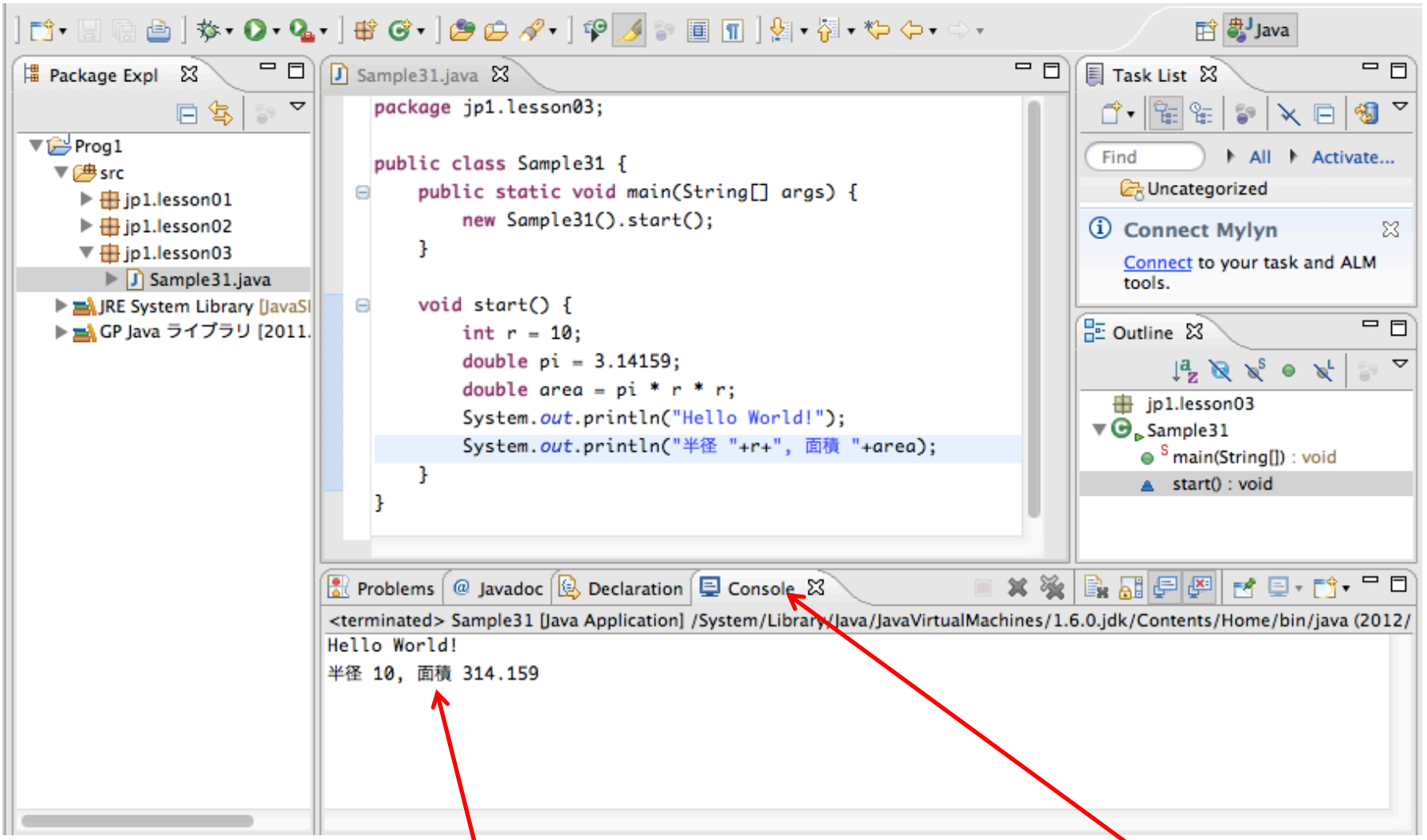
- これまでは、文字列や値を表示するのには、主に JOptionPane.showMessageDialog() を使っていましたが、ちょっとしたものを表示するのには、System.out.println() という方法もあります。
- 引数には文字列や変数を取ると適当な表示をしてくれます。
- Eclipse の場合、右下の Console というタブのところに表示されます。
  - コンソール出力といいます

例題3-1: コンソール出力

- System.out.println() を使って、コンソールに Hello World! という文字列と、半径10の円の面積を表示せよ。

- パッケージ
  - jp1.lesson03
- クラス名
  - Sample31

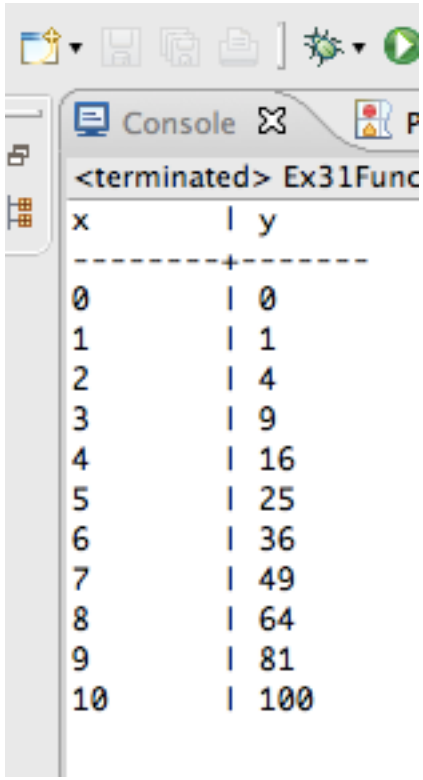
例題3-1: コンソール出力

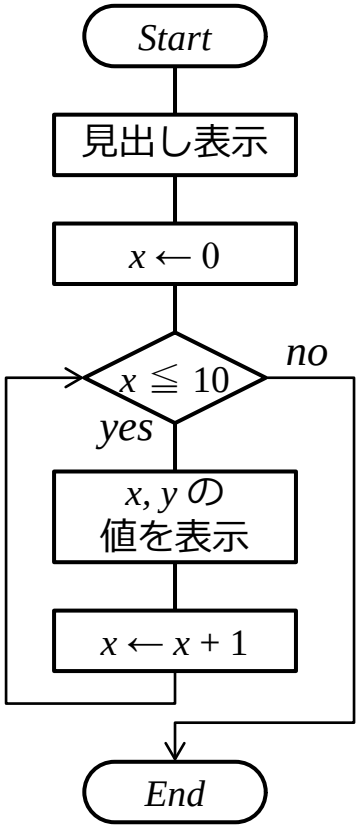


問題3-B-1：関数の値の表示

- $y=x^2$ について考える。xを0から10まで1ずつ増やしたときのyの値を、下のように整列して表示せよ。2列目の前にはタブ (“¥t”)を出力すると列が揃う。

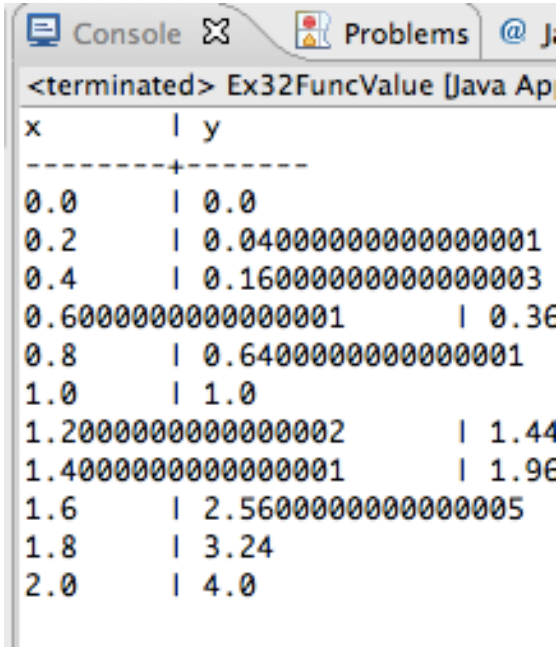
- パッケージ
  - jp1.lesson03
- クラス名
  - Ex31FuncValue

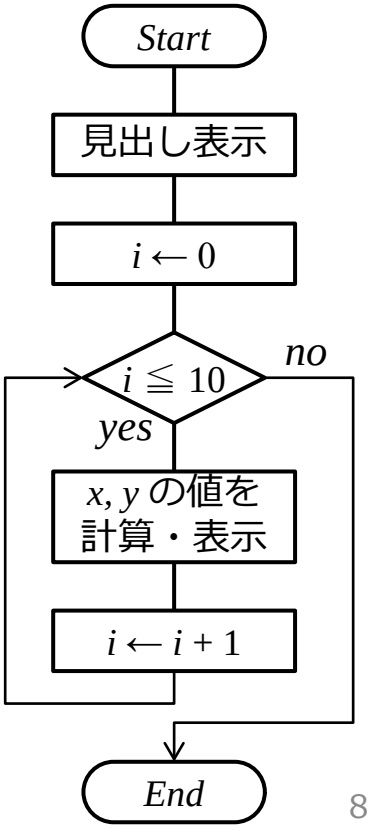




問題3-B-2：関数の値の表示

- $y=x^2$ について考える。x を 0 から 2 まで 10ステップで（つまり0.2刻みで）増やしたときのyの値を、整列して表示せよ。
- doubleの変数を表示すると、誤差が出る場合があるが、そのまま表示するので構わない。
  - 刻みの大きさなどは0, 2, 10 といった値からプログラム中で求めるのが望ましい。（難しければ0.2をプログラムに与えても良い）





- パッケージ
  - jp1.lesson03
- クラス名
  - Ex32FuncValue

問題3-A-1：関数の値の表示

- $y = x^2$ について考える。
  x を 0 から 3 まで 0.2刻みで増やしたとき、y が 7 より小さい範囲について、整列して表示せよ。この際、x の 最大値を与えてはならない。
  - プログラム中で、y が 7 を超えるかどうかを調べる
- 今回は単調増加関数で、いつか5を超えるが、挙動のわからない関数の場合のことを考えると、繰り返しの上限回数を設定することが望ましい。

- パッケージ
  - jp1.lesson03
- クラス名
  - Q31FuncValue

グラフの描画 ～キャンバスの利用～

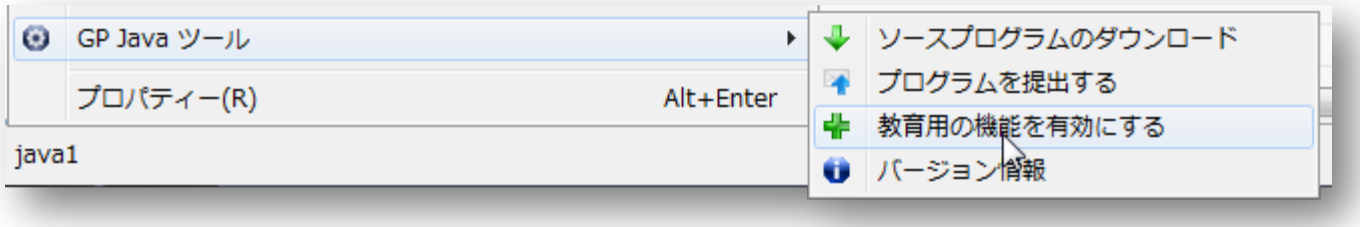
教材フレームワークの確認

- Eclipseを起動
  - プロジェクトを右クリックしたメニューに「GP Javaツール」という項目があるか？

無ければ、
プログラミング入門1の資料で「補足資料: 教材フレームワーク」に従い、教材フレームワークをインストールする。

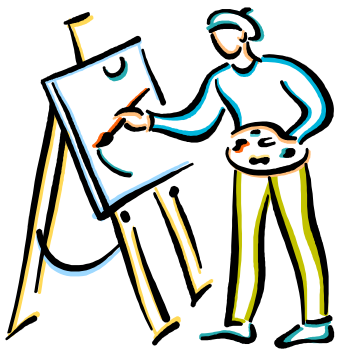
分化教材の有効化

- 「GP Javaツール」メニューから「教育用の機能を有効にする」を選択



キャンバスの利用

- 描画をする場所を用意する
  - 描画する機能も用意されている
  - 描画のモデルとしては一般的
  - GPJava.Canvas は、初学者でも使えるように少し簡単にしてある。



- 詳しくは、プログラミング入門1の資料（第1回 課題(01: ベーシック), 第2回 課題(02:キャンバス)) を参照のこと

Canvas の使い方の概要(1)

- package の次の行に、import 文を記入
  - import gpjava.Canvas;
- 実行の最初に、Canvas.show() を実行
  - 500 x 500 のキャンバスが出現
  - 大きさを指定する場合は、幅と高さを引数に与える

Canvas の使い方の概要(2)

- 表示用のメソッド群(上から、文字列、線、長方形、楕円)
  - Canvas.drawString(double x, double y, String message);
  - Canvas.drawLine(double x1, double y1, double x2, double y2)
  - Canvas.drawRect(double x, double y, double width, double height);
  - Canvas.drawOval(double x, double y, double width, double height);
- 塗りつぶし用のメソッド群(上から、長方形、楕円)
  - Canvas.fillRect(double x, double y, double width, double height);
  - Canvas.fillOval(double x, double y, double width, double height);

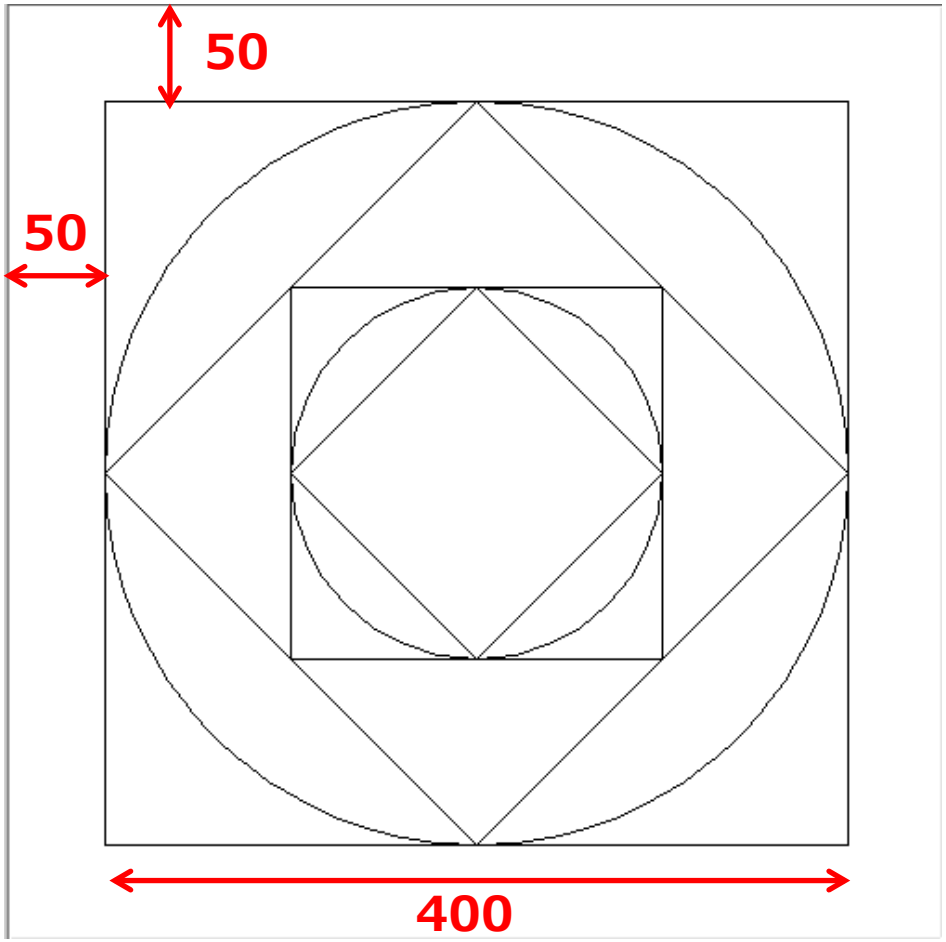
Canvas の使い方の概要(3)

- 色(RGB)の指定
  - Canvas.setColor(int red, int green, int blue);
- 画面表示の消去(初期化)
  - Canvas.clear();
- 一定時間の休止
  - Canvas.waitForCountdown(int msec);
- マウスのクリックの認識
  - Canvas.waitForPoint(String message);
    - Canvas.getPointedX(); // X座標を返却
    - Canvas.getPointedY(); // Y座標を返却

例題3-2：キャンバスの導入

- キャンバスを使って、下のような図を描け.

- パッケージ
  - jp1.lesson03
- クラス名
  - Sample32



注) 赤字・赤線は大きさなので表示しなくて良い

例題3-2：キャンバスの導入

```java
package jp1.lesson03;

import gpjava.Canvas;

public class Sample32 {
    public static void main(String[] args) {
        new Sample32().start();
    }

    void start() {
        Canvas.show();
        Canvas.drawRect(50, 50, 400, 400);
        Canvas.drawOval(50, 50, 400, 400);
        Canvas.drawLine(250, 50, 50, 250);
        Canvas.drawLine(250, 50, 450, 250);
        Canvas.drawLine(50, 250, 250, 450);
        Canvas.drawLine(450, 250, 250, 450);
        Canvas.drawRect(150, 150, 200, 200);
        Canvas.drawOval(150, 150, 200, 200);
        Canvas.drawLine(250, 150, 150, 250);
        Canvas.drawLine(250, 150, 350, 250);
        Canvas.drawLine(150, 250, 250, 350);
        Canvas.drawLine(350, 250, 250, 350);
    }
}
```

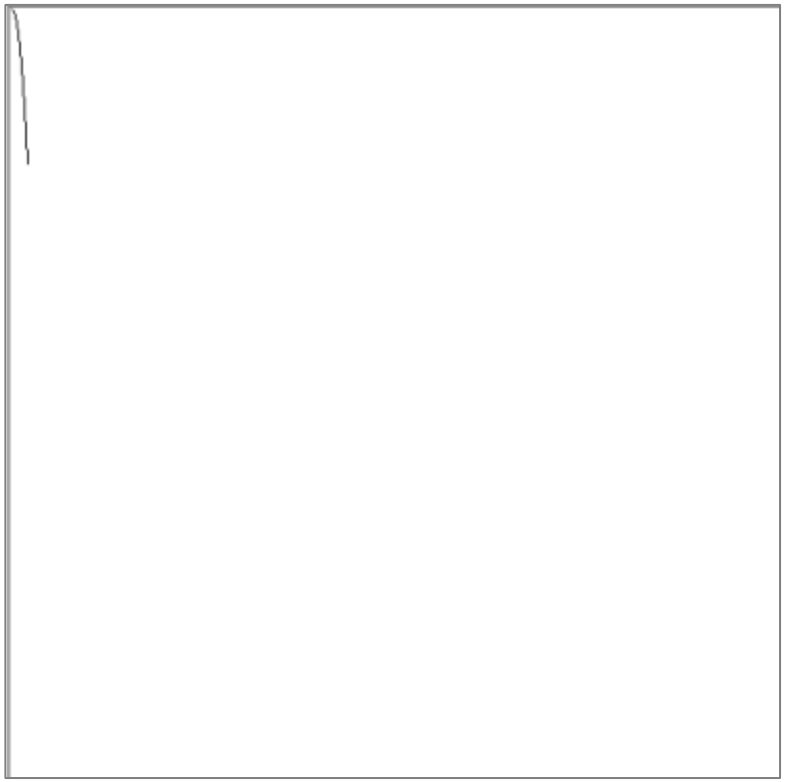
```java
package jp1.lesson03;

import gpjava.Canvas;

public class Sample32 {
    public static void main(String[] args) {
        new Sample32().start();
    }

    void start() {
        Canvas.show();
        drawPattern(50, 50, 400);
        drawPattern(150, 150, 200);
    }

    void drawPattern(int x, int y, int width) {
        int left=x, right=x+width, top=y, bottom=y+width;
        int widthCenter=(left+right)/2;
        int heightCenter=(top+bottom)/2;
        Canvas.drawRect(x, y, width, width);
        Canvas.drawOval(x, y, width, width);
        Canvas.drawLine(widthCenter, top, left, heightCenter);
        Canvas.drawLine(widthCenter, top, right, heightCenter);
        Canvas.drawLine(left, heightCenter, widthCenter, bottom);
        Canvas.drawLine(right, heightCenter, widthCenter, bottom);
    }
}
```

どちらでも同じ図が表示される。

問題3-B-3：値のグラフ化(1)

- $y = x^2$ に関して、x を 0 から 10 まで1刻みで増やしながら、グラフを描け。描画の座標値は、計算の結果得られる値をそのまま使って表示する。

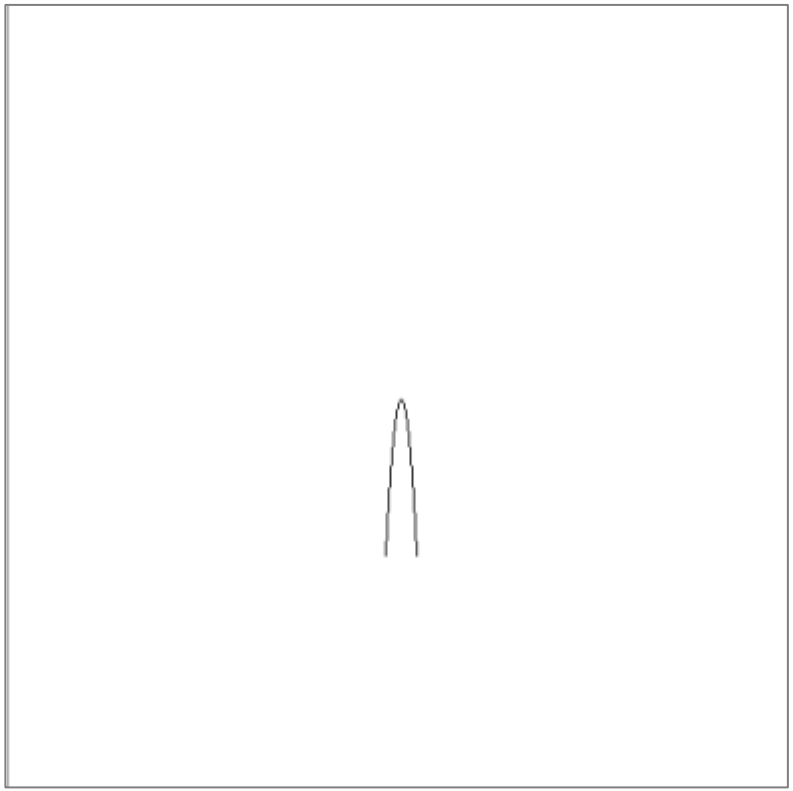
- パッケージ
  - jp1.lesson03
- クラス名
  - Ex33DrawGraph



問題3-B-4：値のグラフ化(2)

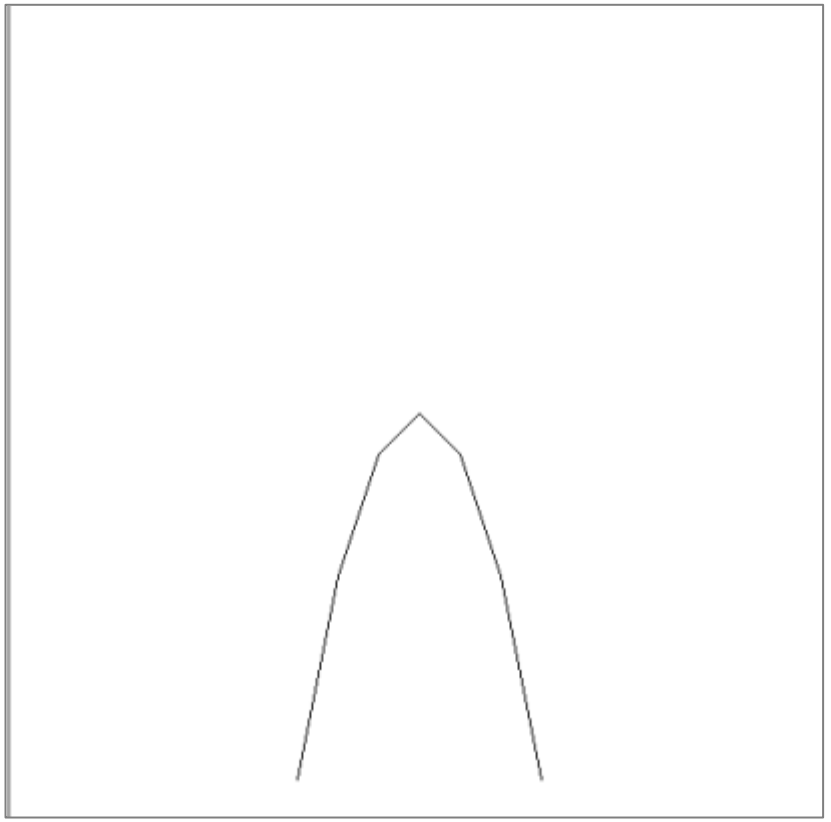
- 問題3-B-3 について、グラフの原点をキャンバスの中央(x=250, y=250)にし、x を −10 から 10 まで1刻みで増やしたグラフを描け。

- パッケージ
  - jp1.lesson03
- クラス名
  - Ex34DrawGraph



問題3-B-5：値のグラフ化(3)

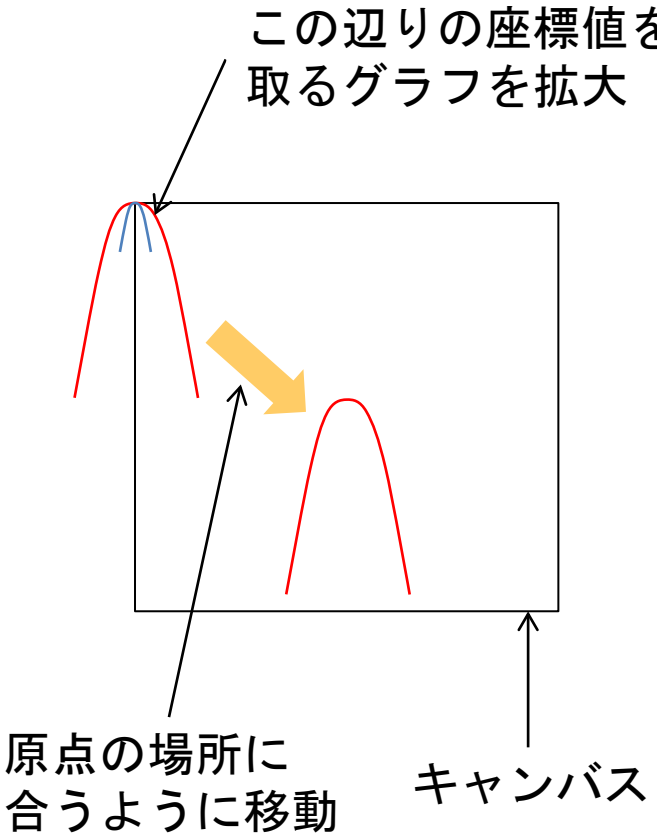
- 問題3-B-4 について、グラフの原点をキャンバスの中央(x=250, y=250)のままで、xの値が-3から3の範囲について、y軸方向がキャンバス一杯になるように、
縦・横を25倍に
表示せよ。



- パッケージ
  - jp1.lesson03
- クラス名
  - Ex35DrawGraph

問題3-B-5：値のグラフ化(3)

- ヒント




```
class Ex35DrawGraph {
   …
   int convertX(int x, int baseX, double scaleX) {
      …
   }
   int convertY(int y, int baseY, double scaleY) {
      …
   }
}
```

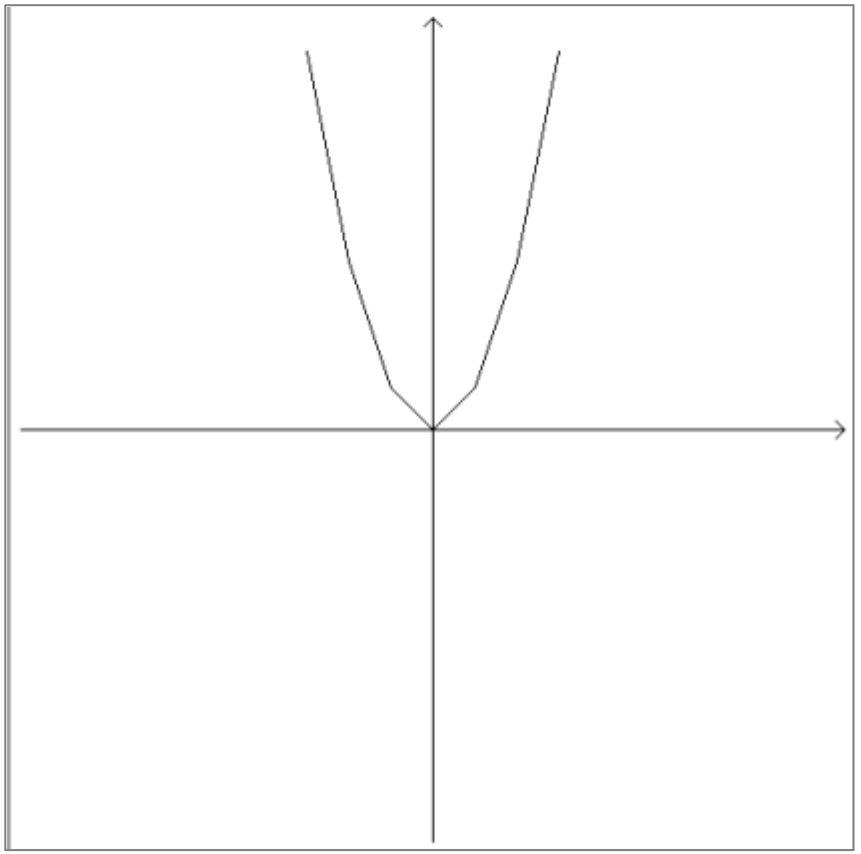
原点や拡大率を引数に取るメソッドを用意しても良いが…

```
class Ex35DrawGraph {
   int baseX;
   int baseY;
   double scaleX;
   double scaleY;
      ….
   int convertX(int x) {
      …
   }
   int convertY(int y) {
      …
   }
}
```

ここに書くと、全てのメソッドに共通に使うことができる。

問題3-B-6：値のグラフ化(4)

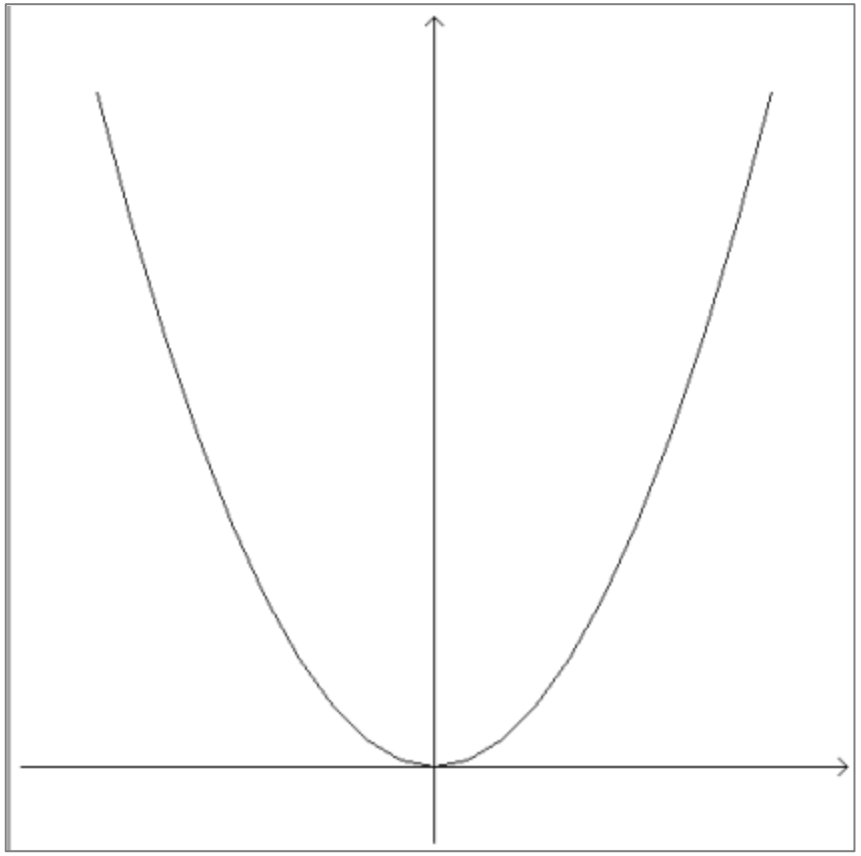
- 問題3-B-5 のグラフについて、右がx軸＋方向、上がy軸+方向となるように変更し、x軸、y軸も描け。



- パッケージ
  - jp1.lesson03
- クラス名
  - Ex36DrawGraph

問題3-B-7：値のグラフ化(5)

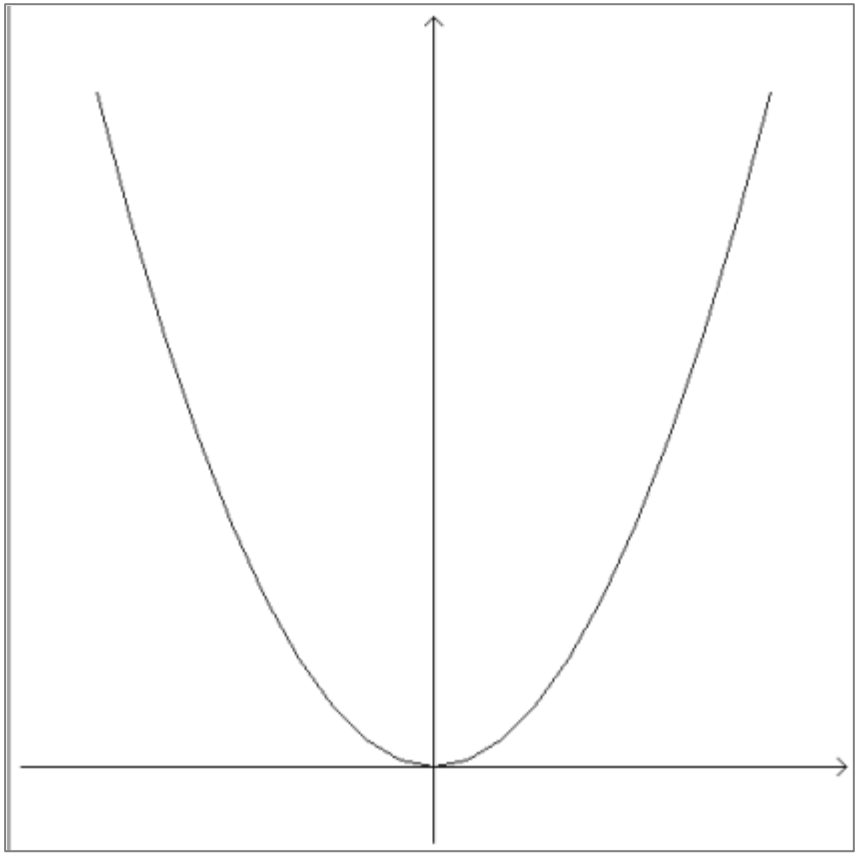
- 問題3-B-6 のグラフについて、原点を下の方にずらし、x軸方向は-10から10まで、y軸方向は0から100までが収まるようにせよ。
  - x軸方向に25倍、y軸方向の4倍程度が適当である。



- パッケージ
  - jp1.lesson03
- クラス名
  - Ex37DrawGraph

問題3-A-2：値のグラフ化

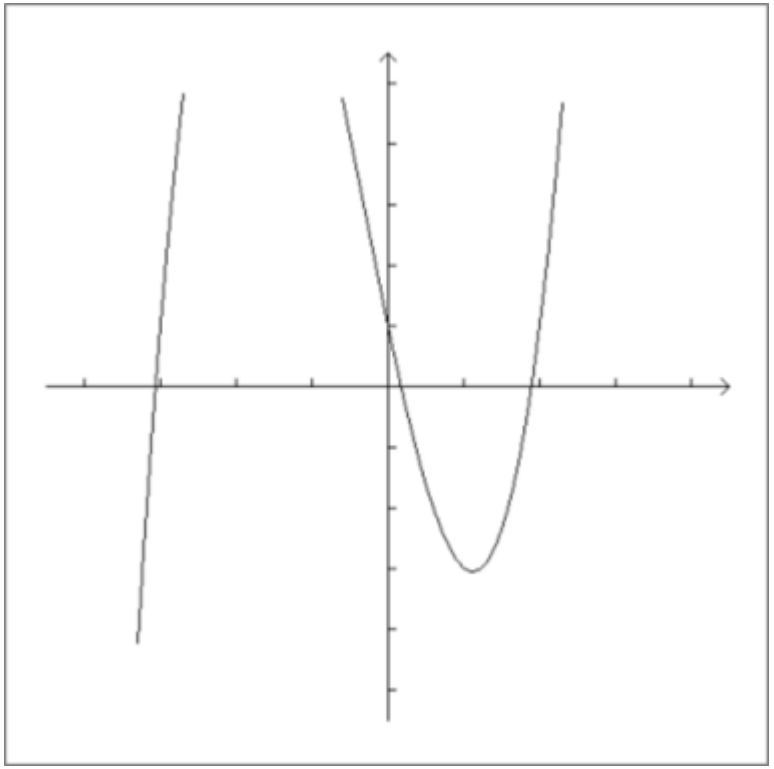
- $y=x^2$のグラフについて、x軸方向は-3から3まで、y軸方向は 0から10までが収まるようにせよ。描画の際には、xの値を0.1刻みで滑らかに表示せよ。
  - x軸方向に50倍、y軸方向の40倍程度が適当である。



- パッケージ
  - jp1.lesson03
- クラス名
  - Q32DrawGraph

問題3-A-3：値のグラフ化

- $y=x^3+x^2-6x+1$のグラフについて、x軸方向は-4から4まで、y軸方向は -5から5の範囲を描画せよ。描画の際には、xの値を0.1刻みで滑らかに表示せよ。
  - x軸方向に50倍、y軸方向の40倍程度が適当である。



- パッケージ
  - jp1.lesson03
- クラス名
  - Q33DrawGraph

宿題

- 基本問題を全て解く
  - 問題3-B-1, 問題3-B-2, 問題3-B-3, 問題3-B-4, 問題3-B-5, 問題3-B-6, 問題3-B-7
- 発展問題はできるだけ解く
  - 問題3-A-1, 問題3-A-2, 問題3-A-3
- GPフレームワークの機能でプログラムを提出
  - http://java2010.cis.k.hosei.ac.jp/appendix/submit-online/
- 〆切
  - 土曜日 23:59