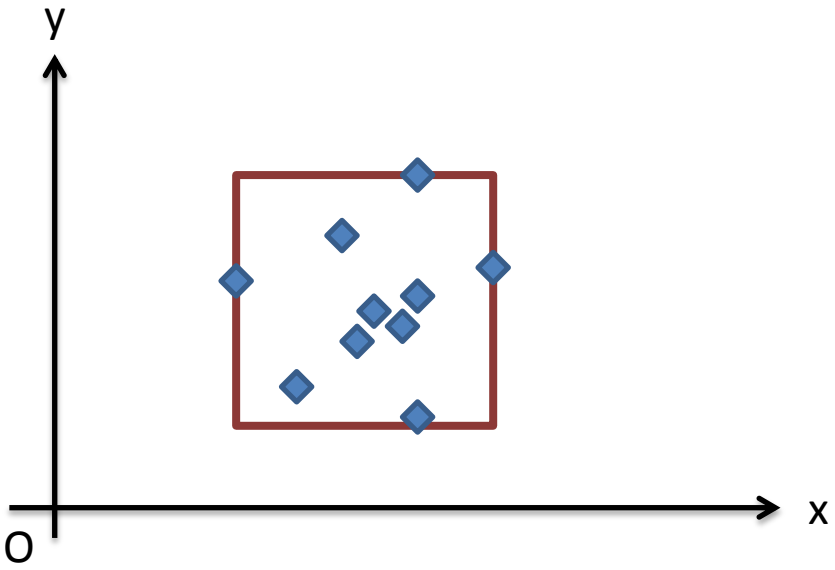
プログラミング基礎

第8.5回

課題

- 100個の点群（2次元）が与えられた際，それらを内包するXY軸に沿った矩形の大きさを求めよ



解き方

1：mainメソッド（RectangleFromPointsクラス）

```
public class RectangleFromPoints
{
   static public void main(String[] args)
   {
      int numps = 100;
      java.util.Random rand = new java.util.Random();

      Points points = new Points(numps);
      //numps個の点を生成
      for(int i = 0; i < numps; i++)
      {
         double x, y;
         Vector2D p;
         x = rand.nextGaussian() * 100.0;//ランダムな値を生成
         y = rand.nextGaussian() * 100.0;//ランダムな値を生成
         p = new Vector2D(x, y);
         System.out.println(x + ", " + y);
         points.setPoints(i, p);
      }
      //点群から最小，最大のx，yを得る
      double minx, maxx, miny, maxy;
      minx = points.getMinX();
      maxx = points.getMaxX();
      miny = points.getMinY();
      maxy = points.getMaxY();

      System.out.println();
      System.out.println(minx + ", " + miny);
      System.out.println(minx + ", " + maxy);
      System.out.println(maxx + ", " + maxy);
      System.out.println(maxx + ", " + miny);
      System.out.println(minx + ", " + miny);
   }
}
```

1: 点を表すクラス

```
Vector2D
{
   double x;
   double y;

   Vector2D(double xx, double yy)
   {
      x = xx;
      y = yy;
   }

   double getX()
   {
      return x;
   }

   double getY()
   {
      return y;
   }
}
```

2.点群を表すクラス

＊コメントが書かれている箇所を自分で作る

```
class Points
{
   int numPoints;
   Vector2D[] ps;

   Points(int numps)
   {
      numPoints = numps;
      ps = new Vector2D[numPoints];
   }

   void setPoints(int i, Vector2D p)
   {
       ps[i] = p;
   }

   double getMinX()
   {
      //配列psの中から最小のXを探し，戻り値とする
   }

   double getMaxX()
   {
      //配列psの中から最大のXを探し，戻り値とする
   }

   double getMinY()
   {
      //配列psの中から最小のYを探し，戻り値とする
   }

   double getMaxY()
   {
      //配列psの中から最大のYを探し，戻り値とする
   }
}
```

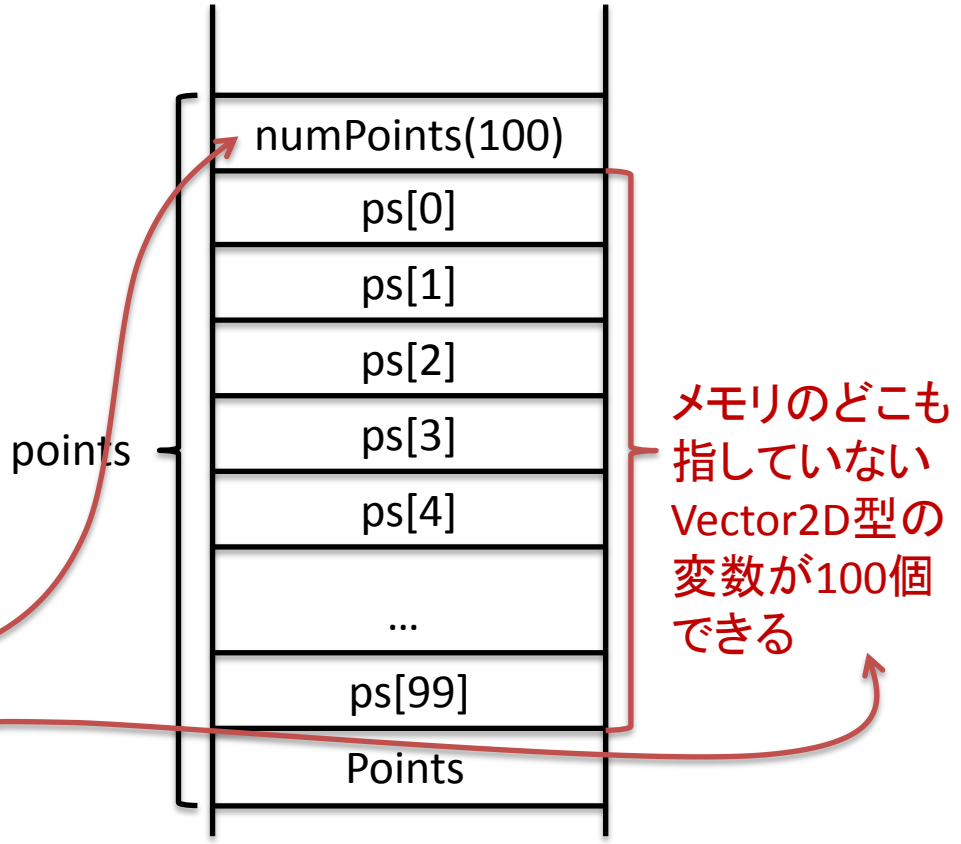
補足

- どう動くのか？

```
int numps = 100;
…
Points points = new Points(numps);

class Points
{
    int numPoints;
    Vector2D[] ps;

    Points(int numps)
    {
        numPoints = numps;
        ps = new Vector2D[numPoints];
    }
```



補足

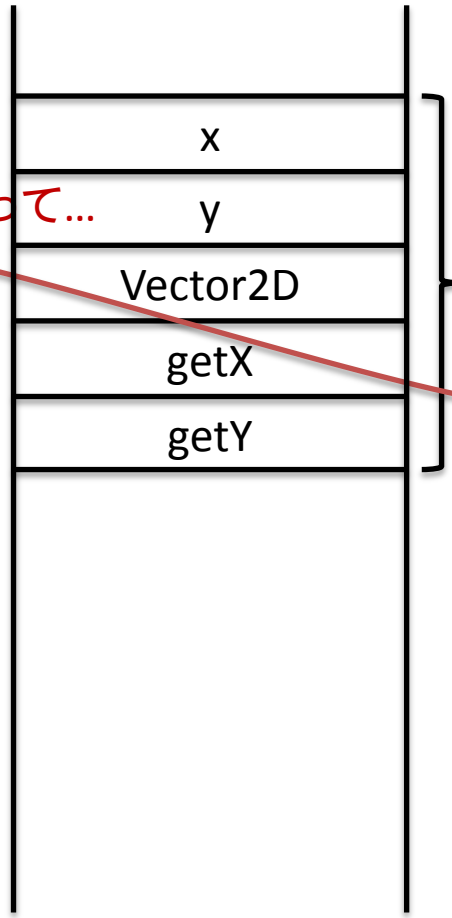
- どう動くのか？

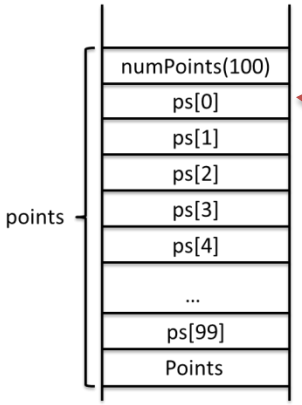
例えばiが0の時…

```
for(int i = 0; i < numps; i++)
{
   …
   p = new Vector2D(x, y);
   …
   points.setPoints(i, p);
}

void setPoints(int i, Vector2D p)
 {
    ps[i] = p;
 }
```

Vector2Dのインスタンスを作って…





インスタンスがメモリのどこにあるか（=p）を配列ps[0]へ代入

補足

- どう動くのか？

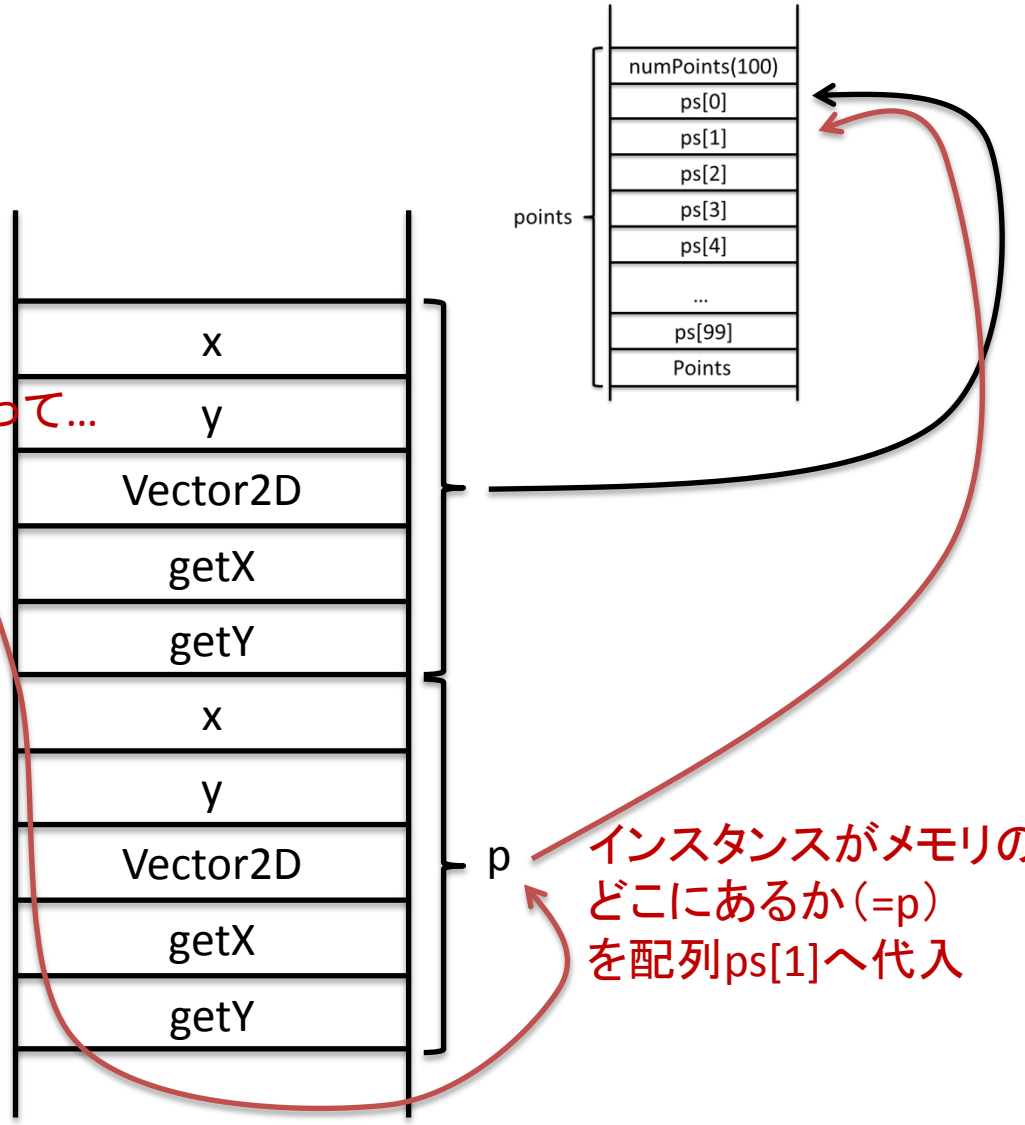
次にiが1の時は

```
for(int i = 0; i < numps; i++)
{
   …
   p = new Vector2D(x, y);
   …
   points.setPoints(i, p);
}

void setPoints(int i, Vector2D p)
 {
    ps[i] = p;
 }
```

さらにVector2Dの
インスタンスを作って…

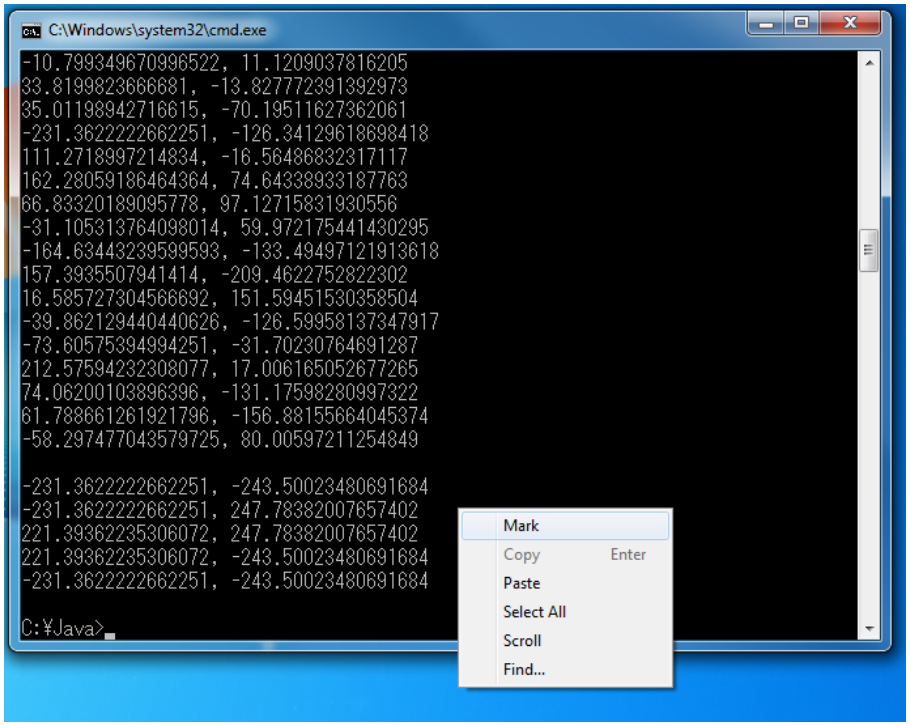
以上が100回繰り返される



補足

- 結果の確認方法（Windowsの場合）

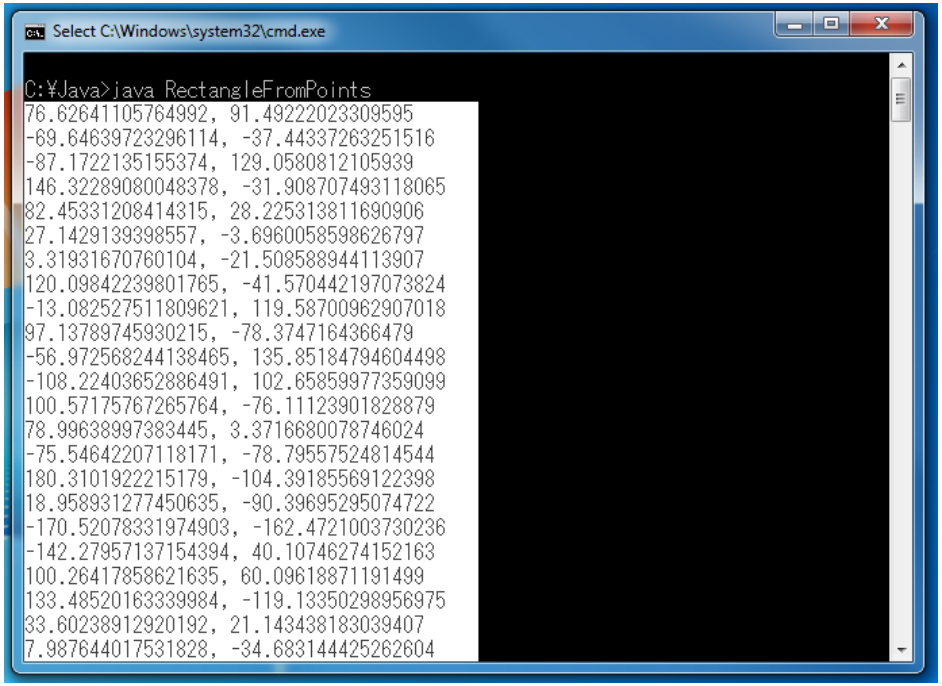
（UbuntuでもOpenOfficeで同様のことが可能？（未確認））



コマンドプロンプト上で右クリック，[選択]をクリック
（スクリーンショットは英語版Windowsのため英語表記）

補足

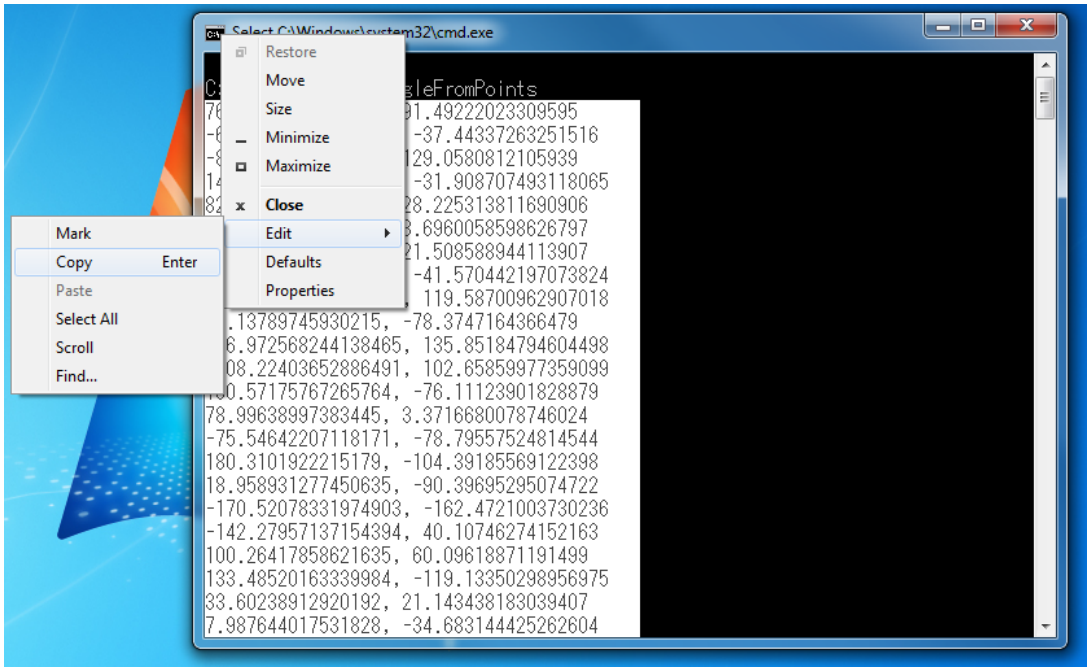
- 結果の確認方法（Windowsの場合）



出力されたデータをすべて選択

補足

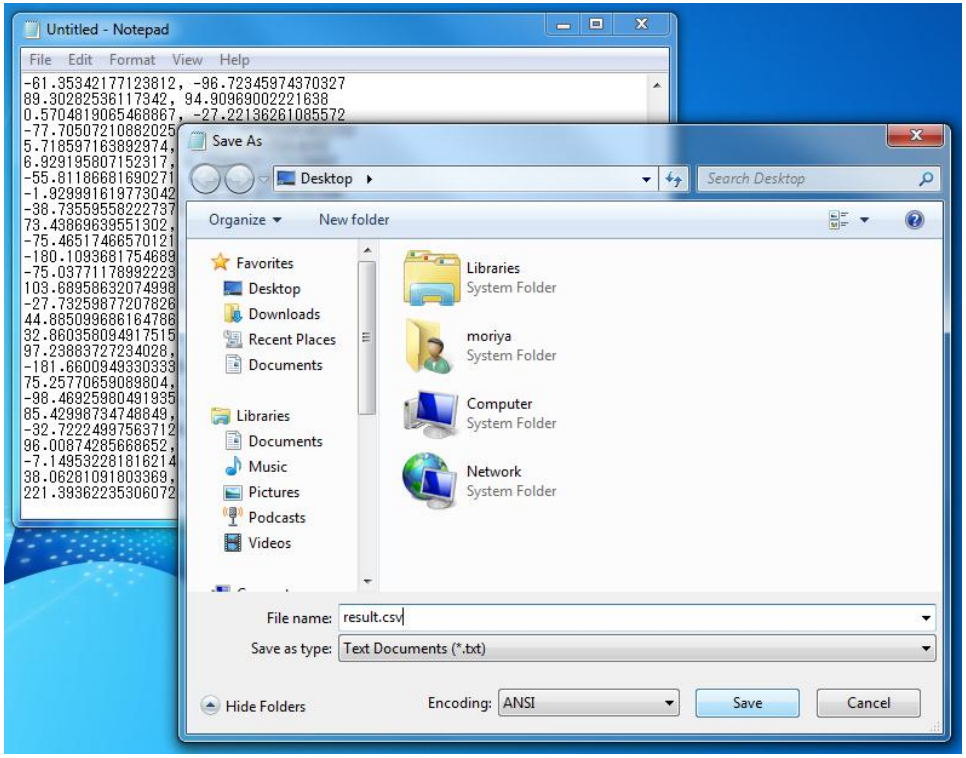
- 結果の確認方法（Windowsの場合）



コマンドプロンプト左上のアイコンを右クリック
[編集]→[コピー]

補足

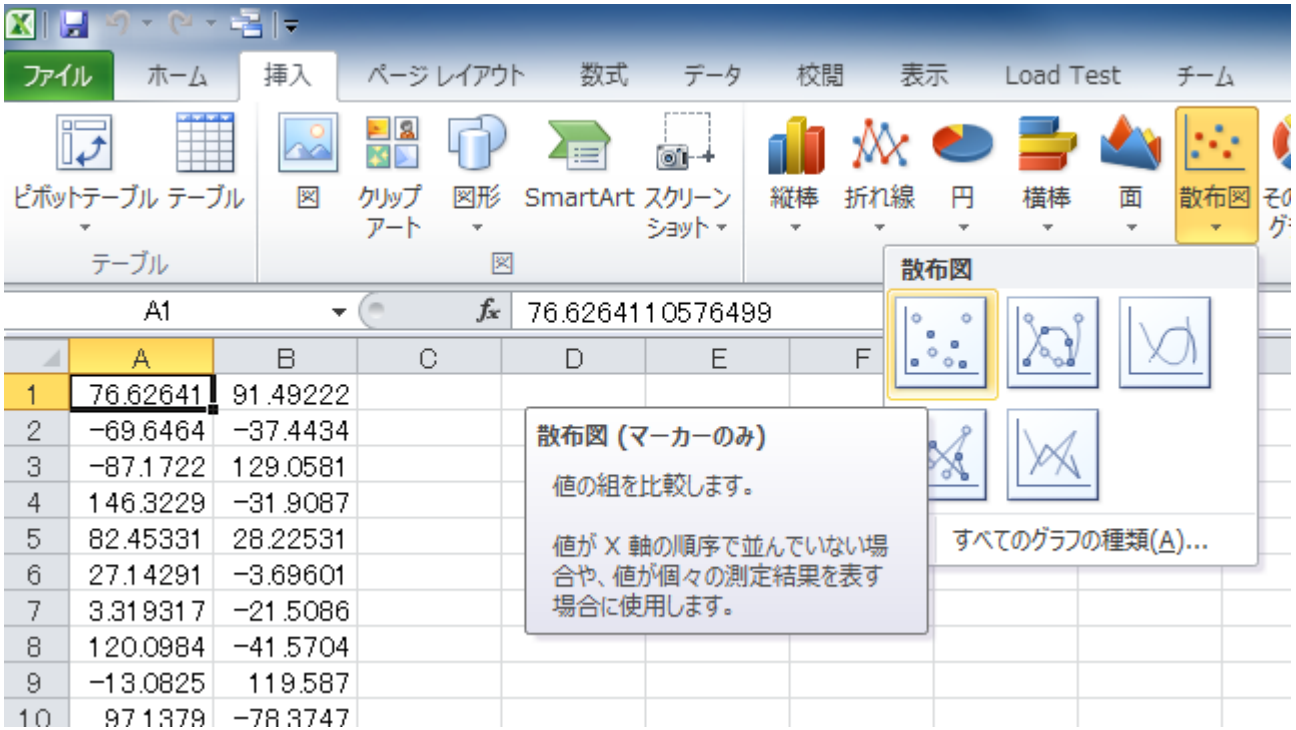
- 結果の確認方法（Windowsの場合）



適当なエディタ（メモ帳でも構わない）へデータを貼り付け
拡張子をcsv（ファイル名.csv）にして保存

補足

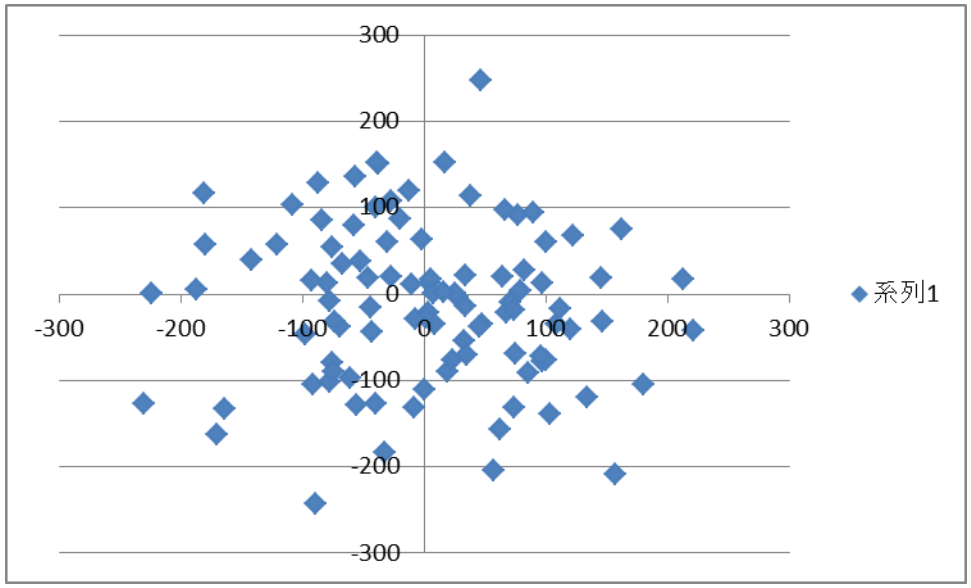
- 結果の確認方法（Windowsの場合）



保存したcsvファイルをExcelで開き，
[挿入]タブ→グラフ[散布図]→[散布図（マーカーのみ）]

補足

- 結果の確認方法（Windowsの場合）

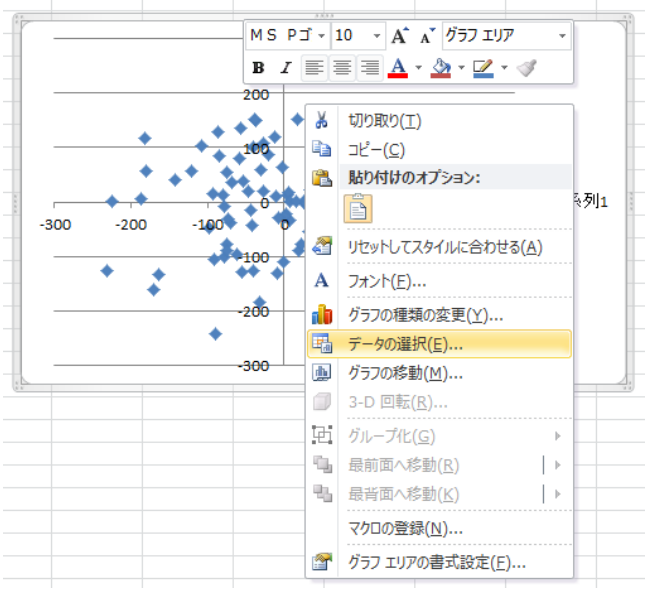


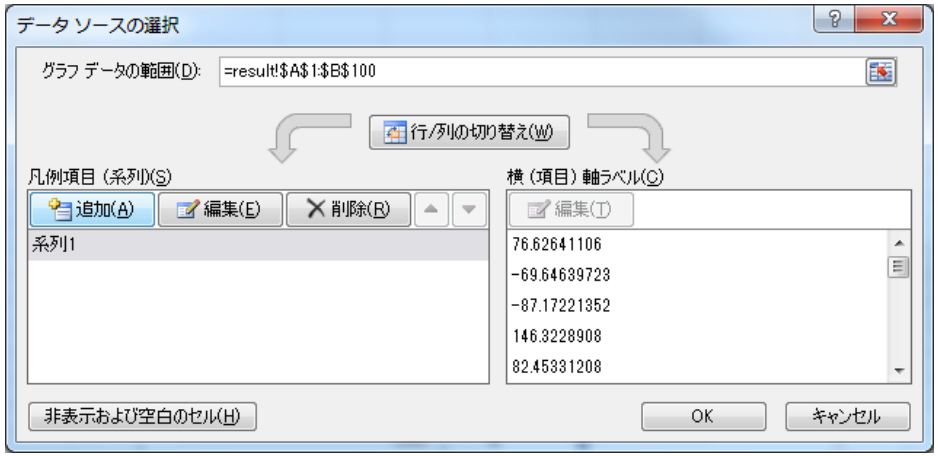
ここまででランダムに生成された100個の点がグラフ化される

補足

- 結果の確認方法（Windowsの場合）

次にプログラムで求めた矩形をグラフ化する

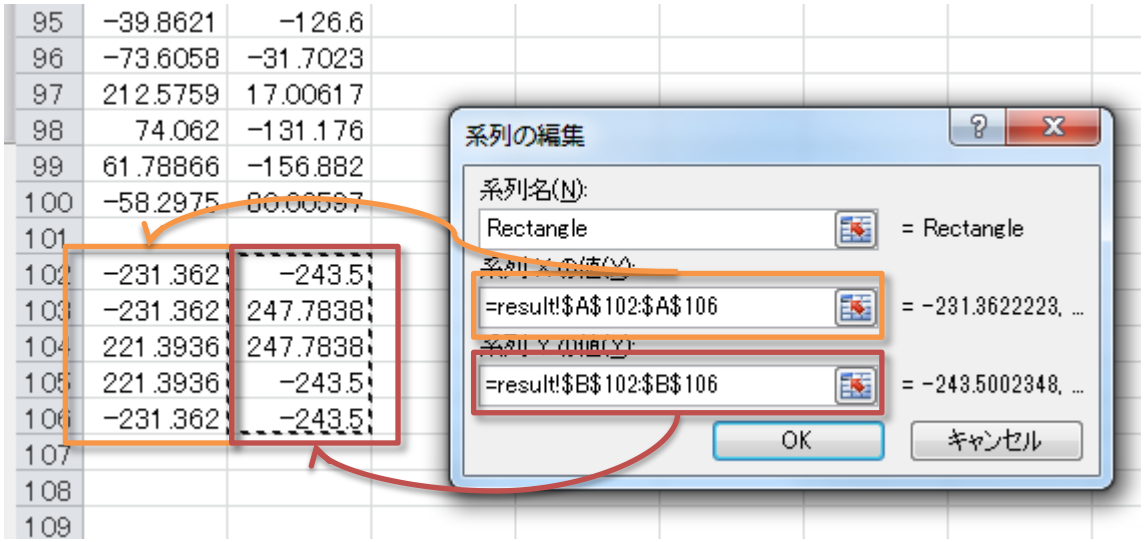




グラフ上で右クリック，[データの選択]→[追加]

補足

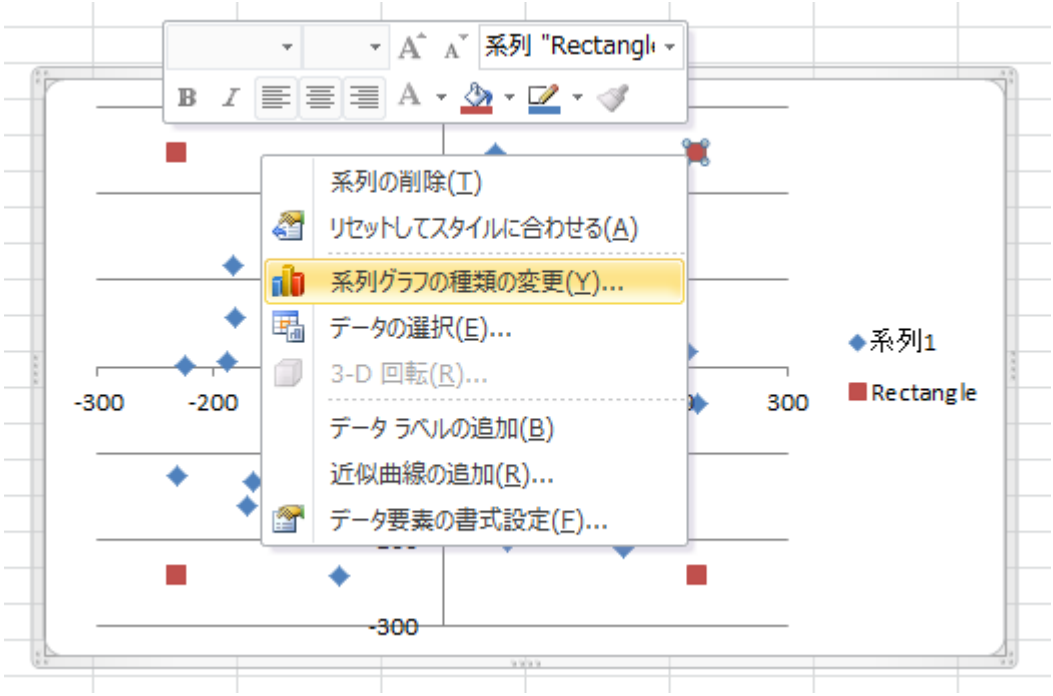
- 結果の確認方法（Windowsの場合）



系列名をRectangle,
矩形の座標は，空白行以下にあるので系列X，Yをそれぞれ選択してOK

補足

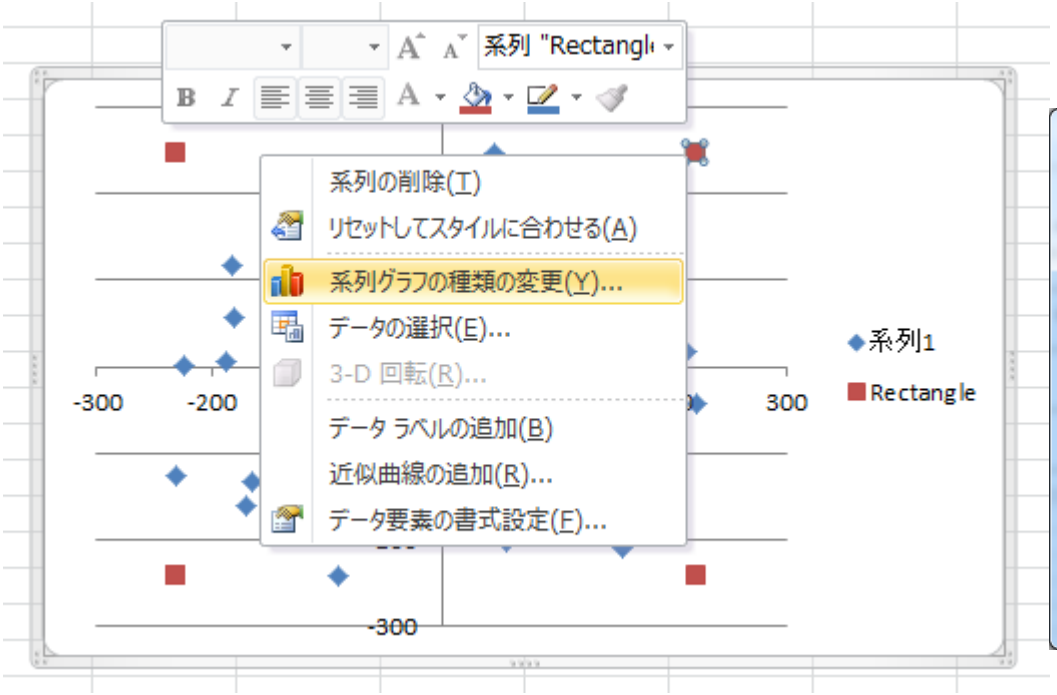
- 結果の確認方法（Windowsの場合）

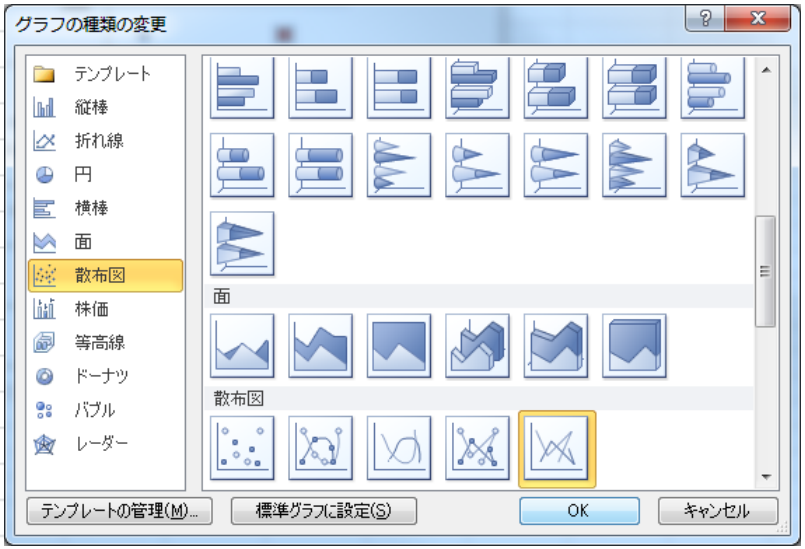


追加された点を右クリック→[系列グラフの種類の変更]

補足

- 結果の確認方法（Windowsの場合）

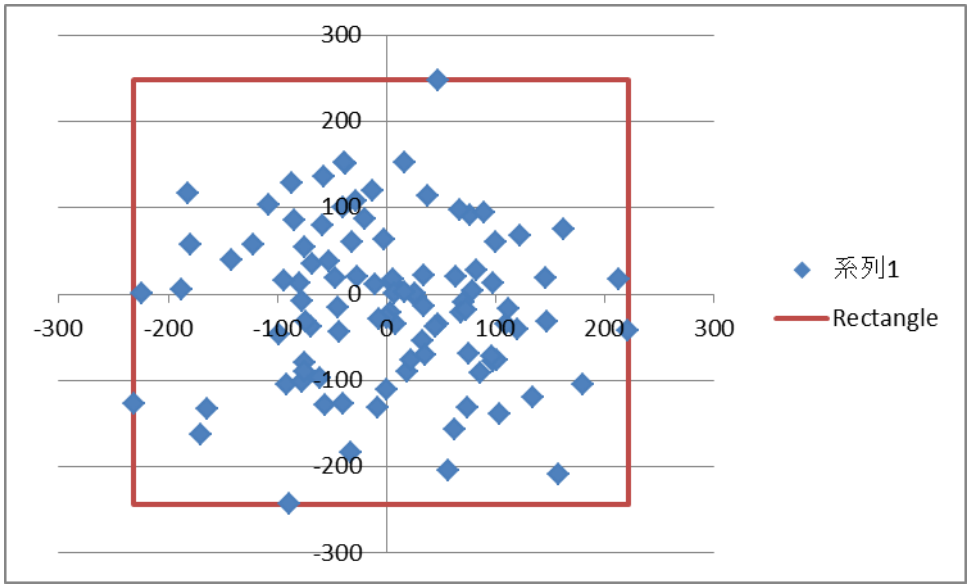




追加された点を右クリック→[系列グラフの種類の変更]
散布図（直線）を右クリックしてOK

補足

- 結果の確認方法（Windowsの場合）



正しいプログラムが書けていれば，点群を内包する矩形が現れる