18-okt-2012/FK

ID1004 Laboration 3, 5-6 November 2012

Labben bör göras individuellt. Beräknad tid ca 2 timmar.

Instruktionen antar att labben utförs i datasal, med hjälp av den integrerade utvecklingsmiljön Eclipse. Alternativt kan du använda egen dator, någon lämplig texteditor, Java JDK och kompilera och köra själv. Om du väljer editorn TextPad, tänk att ha installerat Java JDK innan du installerar TextPad, eftersom TextPad vid sin installation då konfigureras för att stödja källkod i Java.

Labben innehåller fyra uppgifter som bygger på varandra. Det är först i den fjärde uppgiften som det blir ett körbart program.

Resurser online

Java version 6 API dokumentation

http://download.oracle.com/javase/6/docs/api/

Java JDK download (SE version 6)

http://www.oracle.com/technetwork/java/javase/downloads/index.html

TextPad editor for Windows

http://www.textpad.com/

Eclipse IDE (välj Classic)

http://www.eclipse.org/downloads/

Uppgift L3.1 Shape

För att dessa klasser ska kompilera så måste källkodsfilerna importera `java.awt.*` . Placera klasserna i ett nytt, gemensamt projekt i Eclipse. Projektet kan heta DrawShapes.

Skapa en klass Shape med följande fält och metoder (ingen main):

- En publik metod draw som inte returnerar något värde, har en tom kropp, och tar dessa parametrar:
  - Graphics g
  - int x
  - int y
  - int width
  - int height

Uppgift L3.2 Rectangle

Skapa en klass Rectangle som ärver från Shape (extends) och som åsidosätter (override) metoden draw. Låt Rectangle.draw rita en fyrkant med bredden width och höjden height. För att rita, använd Graphics-objektet g:

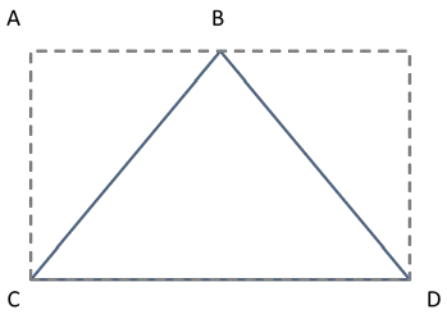
- Sätt färgen: `g.setColor(Color.BLACK)`
- Rita en rektangel: `g.drawRect(x, y, width, height)`

Uppgift L3.3 Triangle

Skapa en klass Triangle som ärver från Shape (extends) och som åsidosätter (override) metoden draw. Låt Triangle.draw rita en pyramid med basen width och höjden height. För att rita, använd Graphics-objektet g (ökande x går åt höger, ökande y nedåt):

- Sätt färgen: `g.setColor(Color.BLUE)`
- Rita en rak linje: `g.drawLine(x1, y1, x2, y2)`

Tips: triangeln omsluts av en tänkt rektangel med bredden width och höjden height.



- A = x, y
- B = x + width/2, y
- C = x, y + height
- D = x + width, y + height

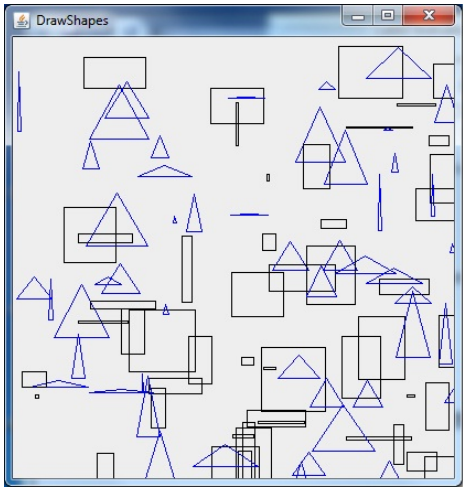
Tips: Använd gärna klassen java.awt.Point för att hålla reda på triangelns hörn.

Uppgift L3.4 DrawShapes

Hämta filerna DrawShapes.java och ShapesPanel.java från kurssidan på kth/social, eller kopiera från källkoderna sist i detta dokument.

DrawShapes innehåller huvudprogrammet (metoden main). ShapesPanel förväntar sig att hitta klasserna Shape, Rectangle och Triangle.

Kör DrawShapes. Resultatet bör set ut ungefär så här:



Studera källkoden till ShapesPanel. Titta på metoden drawShapeSomewhere. Den sista parametern har typen Shape. Men en Shape ritar ju ingenting.

Hur kommer det sig att det går att rita både rektanglar och trianglar med samma metod?

Hur svårt vore det att lägga till ytterligare en Shape, t ex en Ellipse och få den ritad med?

```
// ShapesPanel.java

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

public class ShapesPanel extends JPanel {

  // Skapa en rektangel och en triangel.
  Rectangle rect = new Rectangle();
  Triangle tri = new Triangle();

  public ShapesPanel() {
    // Tala om vilken första storlek vi önskar ha.
    setPreferredSize (new Dimension (400,400));
  }

  // Denna metod anropas från paintComponent och dess uppgift är att
  // rita den Shape som bifogats på ett slumpmässigt valt ställe på
  // skärmen, och med en storlek mellan 1 och 60.
  protected void drawShapeSomewhere(Graphics g,
                                    Dimension panelSize,
                                    Shape sh)
  {
    // Slumpa fram var på ritytan
    int shapeX = (int) (Math.random() * panelSize.getWidth());
    int shapeY = (int) (Math.random() * panelSize.getHeight());

    // Slumpa fram bredd och höjd.
    int shapeWidth  = 1 + (int) (Math.random() * 60);
    int shapeHeight = 1 + (int) (Math.random() * 60);

    // Rita denna Shape.
    sh.draw (g, shapeX, shapeY, shapeWidth, shapeHeight);
  }

  // Metoden paintComponent anropas varje gång denna JPanel behöver
  // ritas om. Det sker automatiskt, t ex om storleken har ändrats.
  public void paintComponent (Graphics g) {
    super.paintComponent(g);

    Dimension panelSize = getSize(); // Hur stor ritytan är just nu.

    // Femtio gånger, rita en rektangel och en triangel.
    for (int i = 0; i < 50; i++) {
      drawShapeSomewhere(g, panelSize, rect);
      drawShapeSomewhere(g, panelSize, tri);
    }

  }

} // slut på ShapesPanel.
```

```
// DrawShapes.java

import javax.swing.*;

public class DrawShapes {

  // Huvudprogram för DrawShapes.
  public static void main (String [] args) {

    // Skapa ett fönster med namnet på programmet.
    JFrame frame = new JFrame ("DrawShapes");

    // Om man stänger fönstret så avslutas programmet.
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Skapa en JPanel som innehåller det som visas i fönstret.
    JPanel panel = new ShapesPanel();

    // Stoppa in panelen i fönstret.
    frame.getContentPane().add(panel);

    // Se till att det som är i fönstret får plats
    frame.pack();

    // Gör fönster och innehåll synligt för användaren.
    frame.setVisible(true);
  }
} // slut på DrawShapes.
```