

Προγραμματισμός II

20/5/2020

Εμφάνιση κειμένου και γραφικών με χρήση
των JFrame, JPanel

Σχεδίαση γραμμής

- Η μέθοδος

Graphics.drawLine(int x1,int y1, int x2, int y2)

σχεδιάζει μια γραμμή από το σημείο (x1,y1) στο σημείο (x2,y2).

Σχεδίαση γραμμής

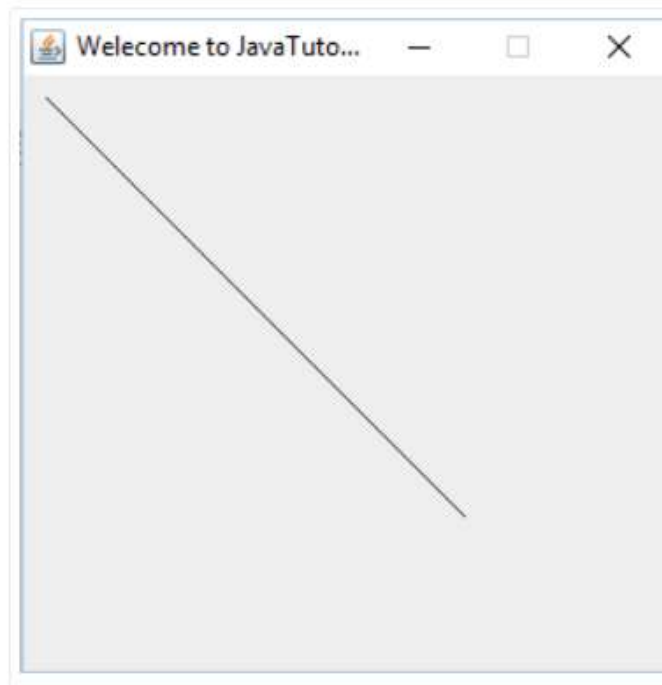
```
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
public class JFrameGraphics extends JPanel{  
    public void paint(Graphics g){  
  
        g.drawLine(10, 10, 200, 300);  
    }  
}
```

Σχεδίαση γραμμής

```
public static void main(String[] args){
    JFrame frame= new JFrame("Welcome");
    frame.getContentPane().add(new JFrameGraphics());
    frame.setSize(600, 400);
    frame.setVisible(true);

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setResizable(false);
}
}
```

Σχεδίαση γραμμής



Σχεδίαση με το JPanel

- Η σχεδίαση ορίζεται μέσω του JPanel, το οποίο είναι ένα συστατικό (component) που υπάρχει στο πακέτο javax.swing.
- Χρησιμοποιήσαμε τη main για να είναι εκτελέσιμος ο κώδικας. Στην πράξη τέτοια συστατικά ορίζονται ξεχωριστά από την κλάση της εφαρμογής που περιέχει τη main.
- Κάθε φορά που το σύστημα ανιχνεύει ένα συστατικό που πρέπει να σχεδιαστεί (πχ είναι ορατό για πρώτη φορά ή είναι ένα παράθυρο του οποίου αλλάζει το μέγεθος), καλεί τελικά τη μέθοδο paint, στην οποία μεταβιβάζει ένα αντικείμενο της κλάσης Graphics.

Σχεδίαση με το JPanel

- Τελικά τα γραφικά στη Java ανάγονται στον ορισμό συστατικών και της μεθόδου `paint`.
- Μπορούμε να διαβάσουμε το τρέχον μέγεθος ενός συστατικού με τις μεθόδους `getWidth()` και `getHeight()`.
- Η επάνω αριστερά γωνία ενός συστατικού είναι το σημείο $(0,0)$ με τις x συντεταγμένες να εκτείνονται προς τα δεξιά και τις y συντεταγμένες να εκτείνονται προς τα κάτω.

Σχεδίαση κύκλου

- Η μέθοδος

Graphics.drawOval(int x,int y, int width, int height)

σχεδιάζει κύκλους ή ελλείψεις.

- Το (x,y) είναι το κέντρο και τα $width,height$ είναι το πάχος, ύψος της έλλειψης (για κύκλο είναι ίσα).

Σχεδίαση κύκλου

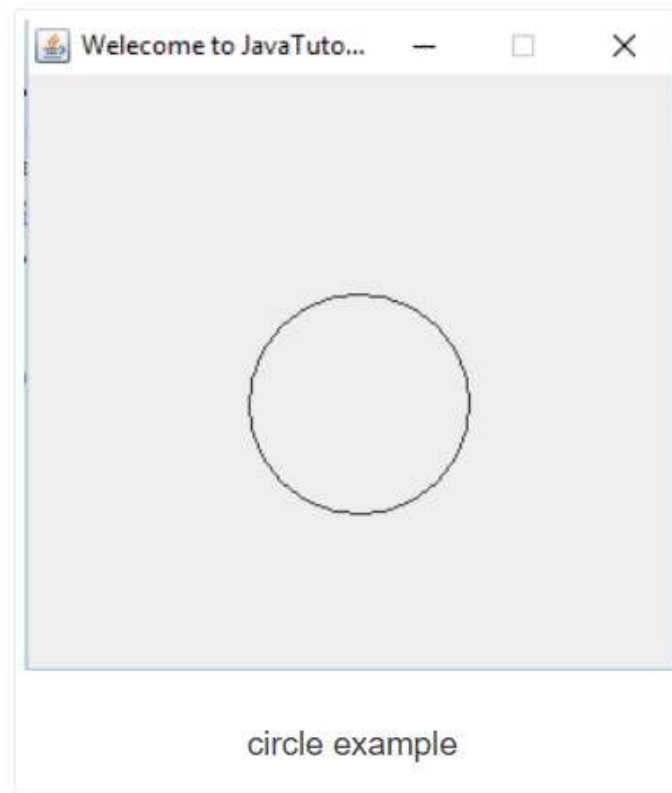
```
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
public class JFrameGraphics extends JPanel{  
    public void paint(Graphics g){  
        g.drawOval(100, 100, 100, 100);  
  
    }  
}
```

Σχεδίαση κύκλου

```
public static void main(String[] args){  
    JFrame frame= new JFrame("JavaTutorial.net");  
    frame.getContentPane().add(new JFrameGraphics());  
    frame.setSize(300, 300);  
    frame.setVisible(true);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.setResizable(false);  
}  
}
```

Σχεδίαση κύκλου

Έξοδος



Σχεδίαση παραλληλογράμμου

- Η μέθοδος

Graphics.drawRect(int x,int y, int width, int height)

σχεδιάζει παραλληλόγραμμα (ή τετράγωνο).

- Το (x,y) είναι η αρχή και τα width,height είναι το πάχος και το ύψος του παραλληλογράμμου (για τετράγωνο είναι ίσα).

Σχεδίαση παραλληλογράμμου

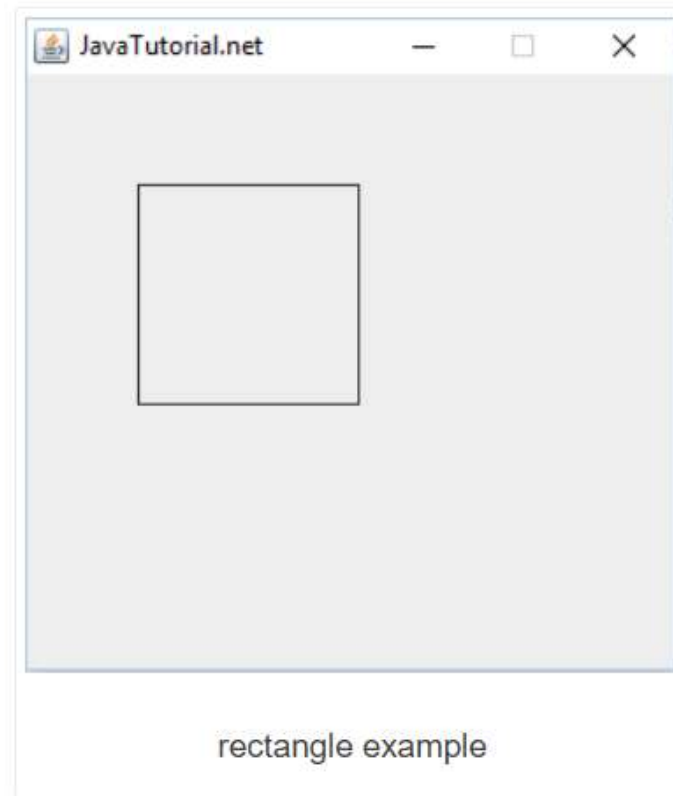
```
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
public class JFrameGraphics extends JPanel{  
    public void paint(Graphics g){  
        g.drawRect(10, 10, 100, 100);  
    }  
}
```

Σχεδίαση παραλληλογράμμου

```
public static void main(String[] args){  
    JFrame frame= new JFrame("JavaTutorial.net");  
    frame.getContentPane().add(new JFrameGraphics());  
    frame.setSize(300, 300);  
    frame.setVisible(true);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.setResizable(false);  
}  
}
```

Σχεδίαση κύκλου

Έξοδος



Σχεδίαση πολυγώνου

- Η μέθοδος

Graphics.drawPolygon(int [],int [], int points)

σχεδιάζει πολύγωνα.

- Ο πρώτος πίνακας περιέχει τις συντεταγμένες x των σημείων των πολυγώνων, ο δεύτερος πίνακας περιέχει τις συντεταγμένες y των σημείων των πολυγώνων και η τρίτη παράμετρος είναι ο αριθμός των σημείων.

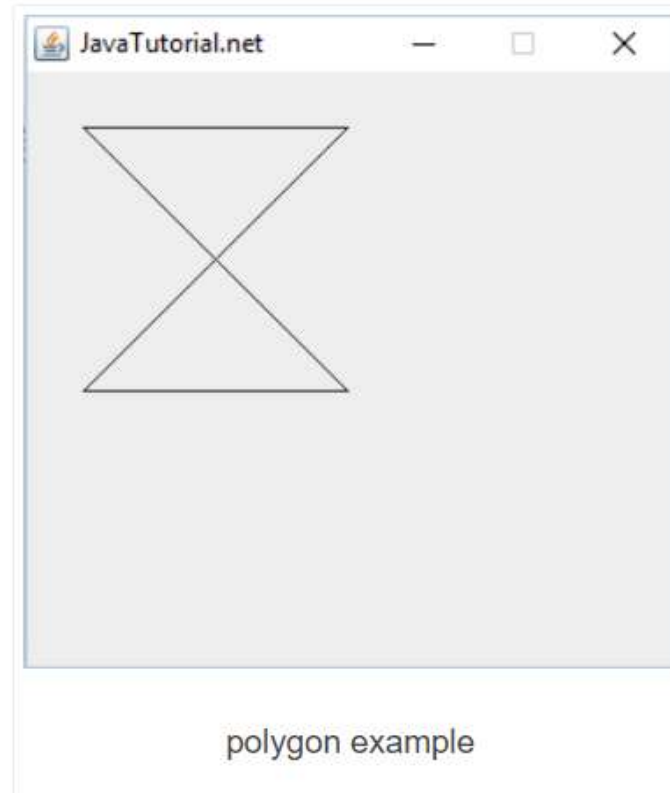
Σχεδίαση πολυγώνου

```
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
public class JFrameGraphics extends JPanel{  
    public void paint(Graphics g){  
        int xValues[] = {25, 145, 25, 145, 25};  
        int yValues[] = {25, 25, 145, 145, 25};  
        int points = 5;  
        g.drawPolygon(xValues, yValues, points);  
    }  
}
```

Σχεδίαση πολυγώνου

```
public static void main(String[] args){  
    JFrame frame= new JFrame("Java");  
    frame.getContentPane().add(new JFrameGraphics());  
    frame.setSize(300, 300);  
    frame.setVisible(true);  
  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.setResizable(false);  
}  
}
```

Σχεδίαση πολυγώνου



Εισαγωγή κειμένου

- Η μέθοδος

Graphics.drawText(string text, int x, int y)

εισάγει κείμενο.

- Ο πρώτος πίνακας περιέχει το κείμενο που θα εισαχθεί και οι δύο παράμετροι x,y είναι οι συντεταγμένες του σημείου από το οποίο θα αρχίσει το κείμενο.

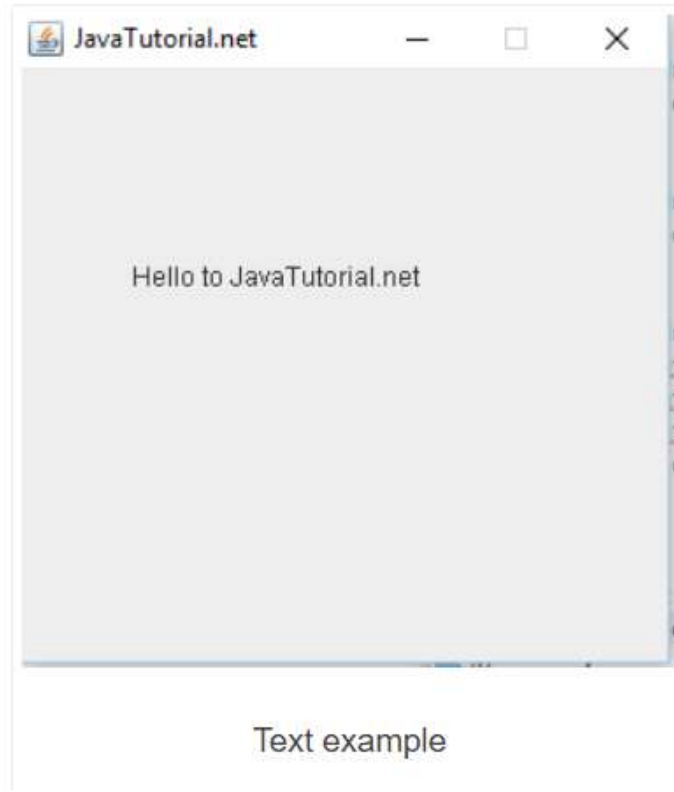
Εισαγωγή κειμένου

```
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
public class JFrameGraphics extends JPanel{  
    public void paint(Graphics g){  
        g.drawString("Hello to JavaTutorial.net", 10, 10);  
    }  
}
```

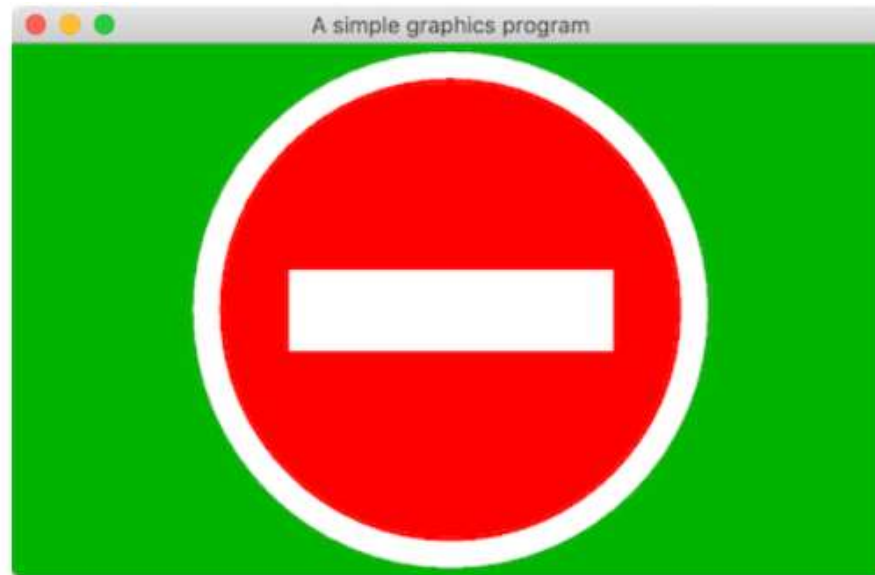
Εισαγωγή κειμένου

```
public static void main(String[] args){  
    JFrame frame= new JFrame("JavaTutorial.net");  
    frame.getContentPane().add(new JFrameGraphics());  
    frame.setSize(300, 300);  
    frame.setVisible(true);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.setResizable(false);  
}  
}
```

Εισαγωγή κειμένου



Σχεδιάζοντας το απαγορευτικό σήμα



Σχεδιάζοντας το απαγορευτικό σήμα

```
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.Point;  
import javax.swing.SwingUtilities;  
import javax.swing.JFrame;  
import javax.swing.JPanel;
```

Σχεδιάζοντας το απαγορευτικό σήμα

```
/**  
 * A panel maintaining a picture of a Do Not Enter sign.  
 */  
public class DoNotEnterSign extends JPanel {  
    private static final long serialVersionUID = 7148504528835036003L;
```

Σχεδιάζοντας το απαγορευτικό σήμα

```
/**  
 * Called by the runtime system whenever the panel needs painting.  
 */  
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
  
    var center = new Point(getWidth() / 2, getHeight() / 2);  
    var radius = Math.min(getWidth() / 2, getHeight() / 2) - 5;  
    var diameter = radius * 2;  
    var innerRadius = (int)(radius * 0.9);  
    var innerDiameter = innerRadius * 2;  
    var barWidth = (int)(innerRadius * 1.4);  
    var barHeight = (int)(innerRadius * 0.35);
```

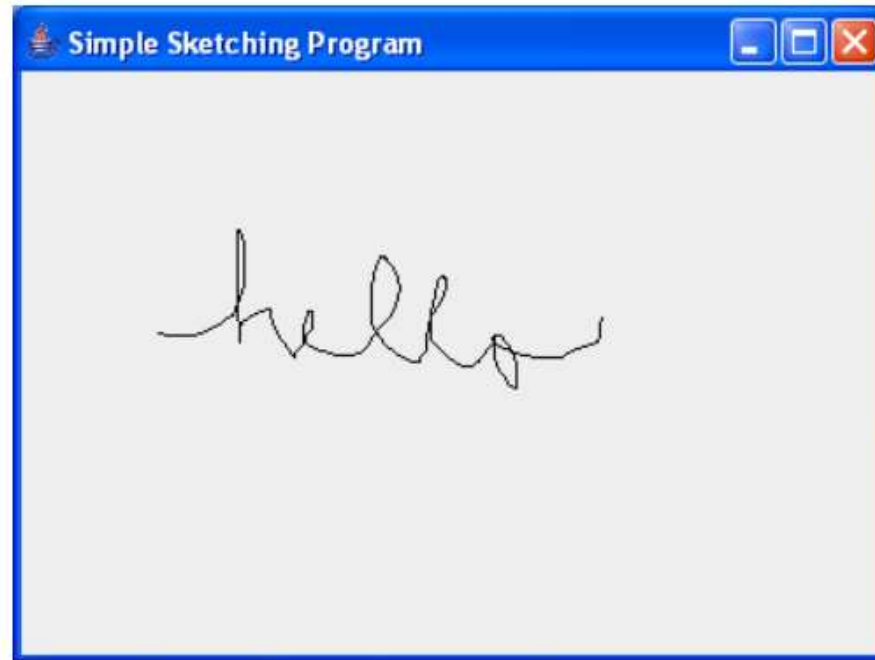
Σχεδιάζοντας το απαγορευτικό σήμα

```
g.setColor(Color.WHITE);  
g.fillOval(center.x - radius, center.y - radius, diameter, diameter);  
g.setColor(Color.RED);  
g.fillOval(center.x - innerRadius, center.y - innerRadius,  
innerDiameter, innerDiameter);  
g.setColor(Color.WHITE);  
g.fillRect(center.x - barWidth/2, center.y - barHeight/2, barWidth,  
barHeight);  
}
```

Σχεδιάζοντας το απαγορευτικό σήμα

```
/**  
 * A little driver in case you want a stand-alone application.  
 */  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> {  
        var panel = new DoNotEnterSign();  
        panel.setBackground(Color.GREEN.darker());  
        var frame = new JFrame("A simple graphics program");  
        frame.setSize(400, 300);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.getContentPane().add(panel, BorderLayout.CENTER);  
        frame.setVisible(true);  
    }); } }
```

Σκίτσα



Σκίτσα

- Το προηγούμενο παράδειγμα ήταν στατικό.
- Σε αυτό το παράδειγμα θα δούμε γραφικά με αλληλεπίδραση.
- Για αυτό θα χρησιμοποιήσουμε events.

Σκίτσα

```
import java.awt.BorderLayout;  
import java.awt.Point;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.awt.event.MouseMotionAdapter;  
import java.awt.Graphics;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.SwingUtilities;  
import javax.swing.JFrame;  
import javax.swing.JPanel;
```

Σκίτσα

```
public class SimpleSketchPanel extends JPanel {  
    private static final long serialVersionUID = -3630443364990545965L;  
  
    private List<List<Point>> curves = new ArrayList<>();  
  
    public SimpleSketchPanel() {  
        // Register event listeners on construction of the panel.  
        addMouseListener(new MouseAdapter() {  
            public void mousePressed(MouseEvent e) {  
                var newCurve = new ArrayList<Point>();  
                newCurve.add(new Point(e.getX(), e.getY()));  
                curves.add(newCurve);  
            }  
        });  
    }  
}
```

Σκίτσα

```
addMouseListener(new MouseMotionAdapter() {  
    public void mouseDragged(MouseEvent e) {  
        curves.get(curves.size() - 1).add(new Point(e.getX(), e.getY()));  
        repaint(0, 0, getWidth(), getHeight());  
    }  
});  
}
```

Σκίτσα

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    for (var curve: curves) {  
        var previousPoint = curve.get(0);  
        for (var point: curve) {  
            g.drawLine(previousPoint.x, previousPoint.y, point.x, point.y);  
            previousPoint = point;  
        }  
    }  
}
```

Σκίτσα

```
/**
 * A little driver in case you want to sketch as a stand-alone
 application.
 */
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        var frame = new JFrame("Simple Sketching Program");
        frame.getContentPane().add(new SimpleSketchPanel(),
BorderLayout.CENTER);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);
        frame.setVisible(true);
    });
}
```

Java 2D API

- Επεκτείνει το AWT, με περισσότερες δυνατότητες για γραφικά και κείμενο.
- Για λόγους συμβατότητας το Java 2D είναι υπερσύνολο του AWT.
- Το Java 2D είναι ιδιαίτερα ισχυρό και χρησιμοποιείται στην κατασκευή πλουσιότερων interfaces, παιχνιδιών, πολυμεσικών εφαρμογών και ειδικών εφέ.
- Υπάρχει και εδώ η `paintComponent`, η οποία εντάσσεται στην `paint`.

Java 2D API

```
import java.awt.EventQueue;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import javax.swing.JFrame;  
import javax.swing.JPanel;
```

Java 2D API

```
class Surface extends JPanel {  
  
    private void doDrawing(Graphics g) {  
  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.drawString("Java 2D", 50, 50);  
    }  
    @Override  
    public void paintComponent(Graphics g) {  
  
        super.paintComponent(g);  
        doDrawing(g);  
    }  
}
```


Java 2D API

```
public class BasicEx extends JFrame {  
    public BasicEx() {  
        initUI();  
    }  
    private void initUI() {  
        add(new Surface());  
  
        setTitle("Simple Java 2D example");  
        setSize(300, 200);  
        setLocationRelativeTo(null);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```

Java 2D API

```
public static void main(String[] args) {
```

```
    EventQueue.invokeLater(new Runnable() {
```

```
        @Override
```

```
        public void run() {
```

```
            BasicEx ex = new BasicEx();
```

```
            ex.setVisible(true);
```

```
        }
```

```
    });
```

```
}
```

```
}
```

Java 2D API



Java 2D API

Περισσότερες πληροφορίες για το Java 2D υπάρχουν στο:

<http://zetcode.com/gfx/java2d/>