

# Python & Java {4} Teachers





# Drawing Polygons

Level 3 - Java



{4}



Python & Java 4 Teachers

# Introduction

## At Play

With the invention of domestic computers followed arts' progression into the digital age.

The ability to create digital art with ease has increased art's accessibility and flexibility and has revolutionized the graphical design and animation industries.

One way in which digital art is implemented is in the use of creating road signs – by creating them digitally, its easy to make many identical copies of the same design!



# Task

The town of 'JavaVille' has built some new roads, but, as there are no road signs, people keep crashing!

You have been tasked with designing some street signs to help control the flow of traffic!



# What it will look like...

{4}

Python & Java 4 Teachers



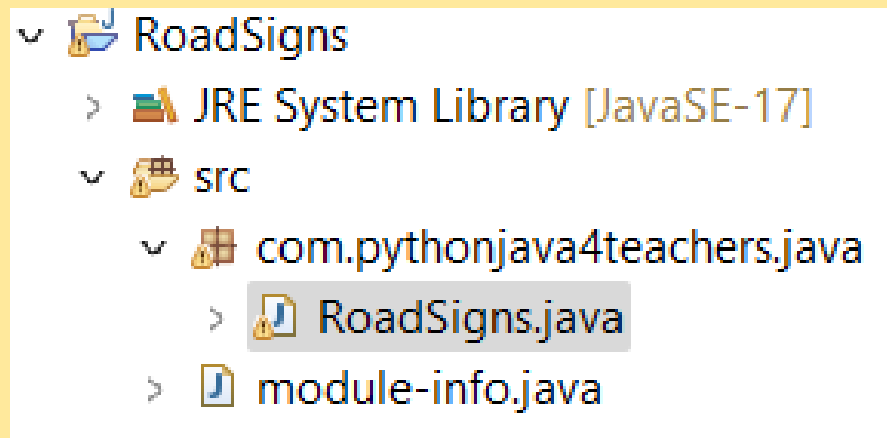
For this example, we are going to design a STOP sign.

Once you are used to drawing the basic shapes, feel free to get creative and design your own!

# Before You Start...

Create a new Java project. If you are not sure what steps to take, use the [Intro To Java](#) resource.

Have your IDE file system set up in the following way:



# Note on Errors

Java is slightly more complex than Python – with more complexity, comes more possibility! But also more potential errors.

Hovering over red underlined code often shows a list of possible solutions.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;

import javax.swing.JFrame;
import javax.swing.SwingUtilities;

public class RoadSi
    public void pai
        super.paint

    Polygon b
    for (int i
        b.addPoin
            (int)

    g.setColor(
    g.fillPolyg
    g.drawPolyg
```

The type javax.swing.SwingUtilities is not accessible

7 quick fixes available:

- [Create class 'SwingUtilities' in package 'javax.swing'](#)
- [Create record 'SwingUtilities' in package 'javax.swing'](#)
- [Create interface 'SwingUtilities' in package 'javax.swing'](#)
- [Create annotation 'SwingUtilities' in package 'javax.swing'](#)
- [Create enum 'SwingUtilities' in package 'javax.swing'](#)
- [Add 'requires java.desktop' to module-info.java](#)
- [Fix project setup...](#)

For example, if you get an error such as 'this package is not accessible', hover over your code and select the following.



# Process

The program should:

- ✓ Create a window and a canvas that we can draw shapes on
- ✓ Draw basic shapes
- ✓ Add colour and text to the sign





# Step 1

Import external libraries and create window

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;
import javax.swing.SwingUtilities;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class RoadSigns extends JPanel {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            var panel = new RoadSigns();
            panel.setBackground(Color.cyan);
            var frame = new JFrame("Stop Sign");
            frame.setSize(220, 230);
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.getContentPane().add(panel, BorderLayout.CENTER);
            frame.setVisible(true);
        });
    }
}
```

First, we import Java packages that help us create our window.

We then create a 'RoadSigns' class and add a main method which is an entry point to start the execution of a program.

Here we initialise our window, setting its size and giving it a title.



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# Step 2

Create a subroutine to draw shapes...

```
public class RoadSigns extends JPanel {  
  
    // Draw shapes  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(() -> {  
            var panel = new RoadSigns();  
            panel.setBackground(Color.cyan);  
            var frame = new JFrame("Stop Sign");  
            frame.setSize(220, 230);  
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
            frame.getContentPane().add(panel, BorderLayout.CENTER);  
            frame.setVisible(true);  
        });  
    }  
}
```

Inside of our RoadSigns class, we are going to create a subroutine where we will draw some shapes.



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



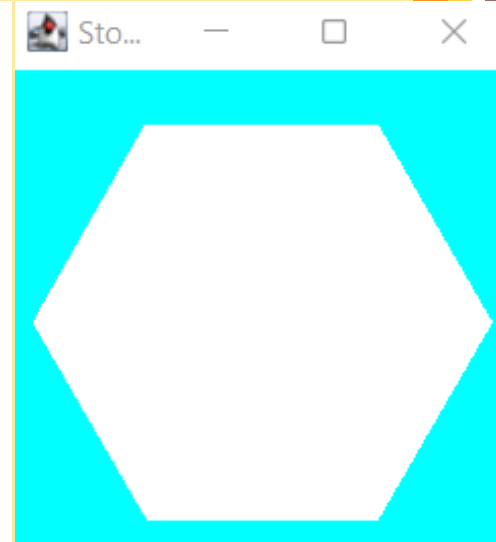
# Step 3

Drawing some shapes...

{4}

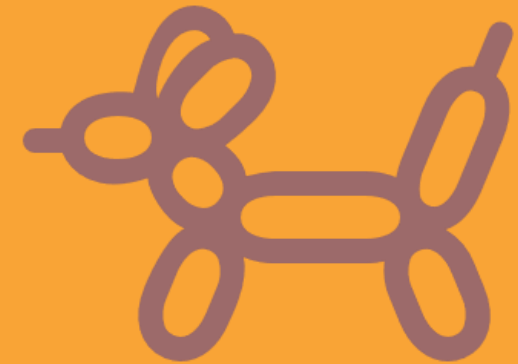
Python & Java 4 Teachers

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
  
    Polygon b = new Polygon();  
    for (int i = 0; i < 6; i++)  
        b.addPoint((int) (100 + 90 * Math.cos(i * 2 * Math.PI / 6)),  
                (int) (100 + 90 * Math.sin(i * 2 * Math.PI / 6)));  
  
    g.setColor(Color.WHITE);  
    g.fillPolygon(b);  
    g.drawPolygon(b);  
}
```



To create the hexagonal shape of the stop sign, create an instance of the Polygon() object. We can then use a FOR loop to add new points to the polygon using some maths.

Don't worry too much about how this works, but feel free to experiment with it and see what happens!



# Step 4

Adding some details

```
g.setColor(Color.WHITE);  
g.fillPolygon(b);  
g.drawPolygon(b);  
  
Polygon p = new Polygon();  
for (int i = 0; i < 6; i++)  
    p.addPoint((int) (100 + 80 * Math.cos(i * 2 * Math.PI / 6)),  
              (int) (100 + 80 * Math.sin(i * 2 * Math.PI / 6)));  
  
g.setColor(Color.RED);  
g.fillPolygon(p);  
g.drawPolygon(p);  
  
}
```

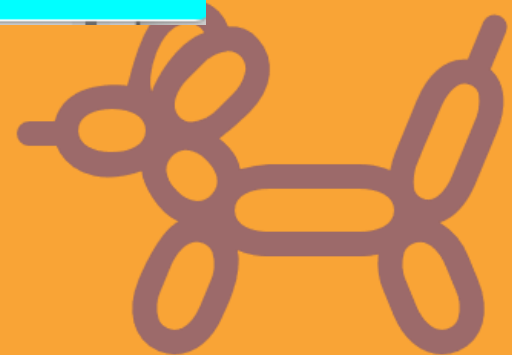
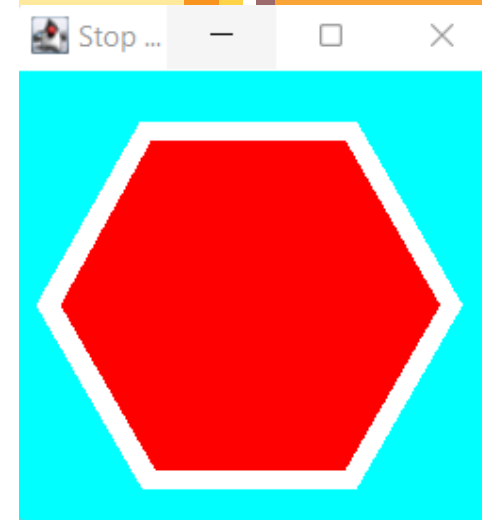
Make a second Polygon, but this time make it slightly smaller. This will simulate the sign having a white border.



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



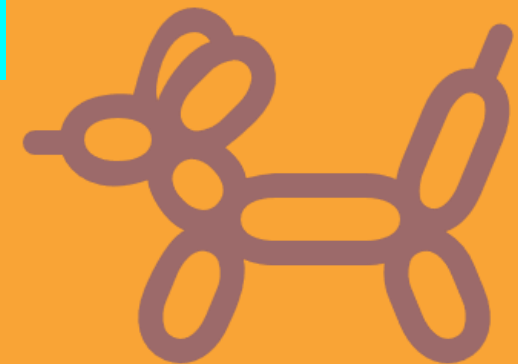
# Step 5

Adding even more details...

```
g.setColor(Color.RED);  
g.fillPolygon(p);  
g.drawPolygon(p);  
  
Graphics2D g2 = (Graphics2D)g;  
  
g2.setColor(Color.WHITE);  
Font font = new Font("Arial", Font.PLAIN, 40);  
g2.setFont(font);  
  
g2.drawString("STOP", 45, 110);  
  
}
```



Finally, we want to add some text to our sign. Create an instance of the Graphics2D class. We will use this to draw some white text to the screen.



# What it will look like when run...

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;
import javax.swing.SwingUtilities;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class RoadSigns extends JPanel {

    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        Polygon b = new Polygon();
        for (int i = 0; i < 6; i++)
            b.addPoint((int) (100 + 90 * Math.cos(i * 2 * Math.PI / 6)),
                (int) (100 + 90 * Math.sin(i * 2 * Math.PI / 6)));

        g.setColor(Color.WHITE);
        g.fillPolygon(b);
        g.drawPolygon(b);

        Polygon p = new Polygon();
        for (int i = 0; i < 6; i++)
            p.addPoint((int) (100 + 80 * Math.cos(i * 2 * Math.PI / 6)),
                (int) (100 + 80 * Math.sin(i * 2 * Math.PI / 6)));

        g.setColor(Color.RED);
        g.fillPolygon(p);
        g.drawPolygon(p);

        Graphics2D g2 = (Graphics2D)g;

        g2.setColor(Color.WHITE);
        Font font = new Font("Arial", Font.PLAIN, 40);
        g2.setFont(font);

        g2.drawString("STOP", 45, 110);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            var panel = new RoadSigns();
            panel.setBackground(Color.cyan);
            var frame = new JFrame("Stop Sign");
            frame.setSize(220, 230);
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.getContentPane().add(panel, BorderLayout.CENTER);
            frame.setVisible(true);
        });
    }
}
```

```
public class RoadSigns extends JPanel {

    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        Polygon b = new Polygon();
        for (int i = 0; i < 6; i++)
            b.addPoint((int) (100 + 90 * Math.cos(i * 2 * Math.PI / 6)),
                (int) (100 + 90 * Math.sin(i * 2 * Math.PI / 6)));

        g.setColor(Color.WHITE);
        g.fillPolygon(b);
        g.drawPolygon(b);

        Polygon p = new Polygon();
        for (int i = 0; i < 6; i++)
            p.addPoint((int) (100 + 80 * Math.cos(i * 2 * Math.PI / 6)),
                (int) (100 + 80 * Math.sin(i * 2 * Math.PI / 6)));

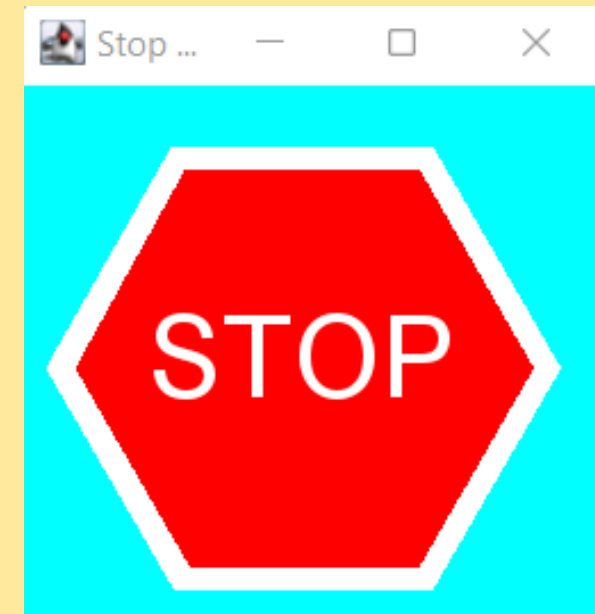
        g.setColor(Color.RED);
        g.fillPolygon(p);
        g.drawPolygon(p);

        Graphics2D g2 = (Graphics2D)g;

        g2.setColor(Color.WHITE);
        Font font = new Font("Arial", Font.PLAIN, 40);
        g2.setFont(font);

        g2.drawString("STOP", 45, 110);
    }
}
```

```
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        var panel = new RoadSigns();
        panel.setBackground(Color.cyan);
        var frame = new JFrame("Stop Sign");
        frame.setSize(220, 230);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(panel, BorderLayout.CENTER);
        frame.setVisible(true);
    });
}
```



# Conclusion

Well done! Now hopefully JavaVille won't have as many traffic jams. However, this is only one sign.

Could you experiment and research ways to draw different shapes? How could you draw a circular, triangular or rectangular sign?

- **Learning outcomes:**

- ✓ Create a GUI based application
- ✓ Use subroutines
- ✓ Use FOR loops to draw shapes
- ✓ Experiment with colours and designs to create your own road sign



# Links to everyday life...

## At Play

Understanding the importance of programming and its relationship to creating repeatable designs

Being able to create original designs and be creative

Experimenting with code and learning how to solve problems using code!



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers



# Congratulations!

You have created a polygon program in Java!



{4}

Python & Java 4 Teachers