

# Designing Graphical Objects

CSC 2014 – Java Bootcamp

Dr. Mary-Angela Papalaskari  
Department of Computing Sciences  
Villanova University

Course website:

[www.csc.villanova.edu/~map/2014/](http://www.csc.villanova.edu/~map/2014/)

# Graphical Objects

- Some objects contain information that determines how the object should be represented visually
- Graphical objects
  - data about position, size, and other attributes
  - methods to draw the object
- Let's look at some other examples of graphical objects:

Example 1: SmilingFacePanel

[SmilingFace.java](#)

[SmilingFacePanel.java](#)



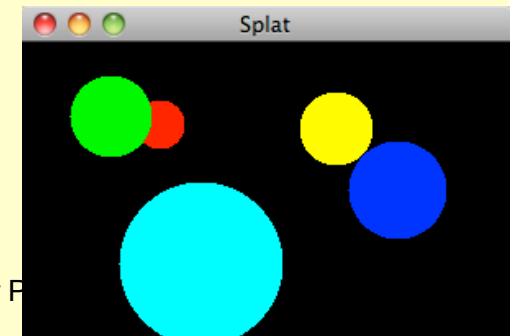
Example 2: SplatPanel

[Splat.java](#)

[SplatPanel.java](#)

[Circle.java](#)

CSC 2014 - Java Bootcamp - Dr F

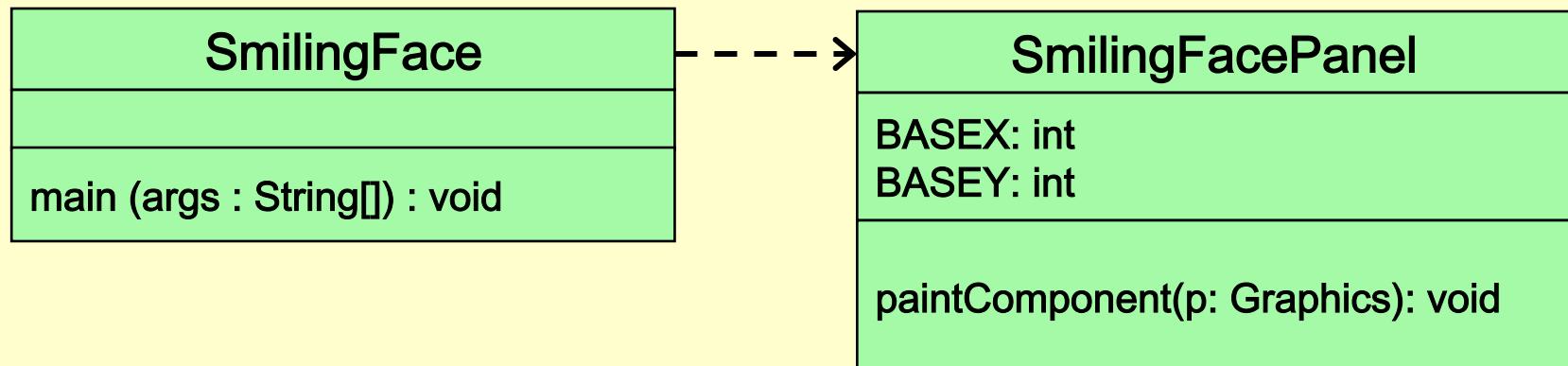


```
//*****  
// SmilingFace.java      Author: Lewis/Loftus  
//  
// Demonstrates the use of a separate panel class.  
//*****  
  
import javax.swing.JFrame;  
  
public class SmilingFace  
{  
    //-----  
    // Creates the main frame of the program.  
    //-----  
    public static void main (String[] args)  
    {  
        JFrame frame = new JFrame ("Smiling Face");  
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
  
        SmilingFacePanel panel = new SmilingFacePanel();  
  
        frame.getContentPane().add(panel);  
  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

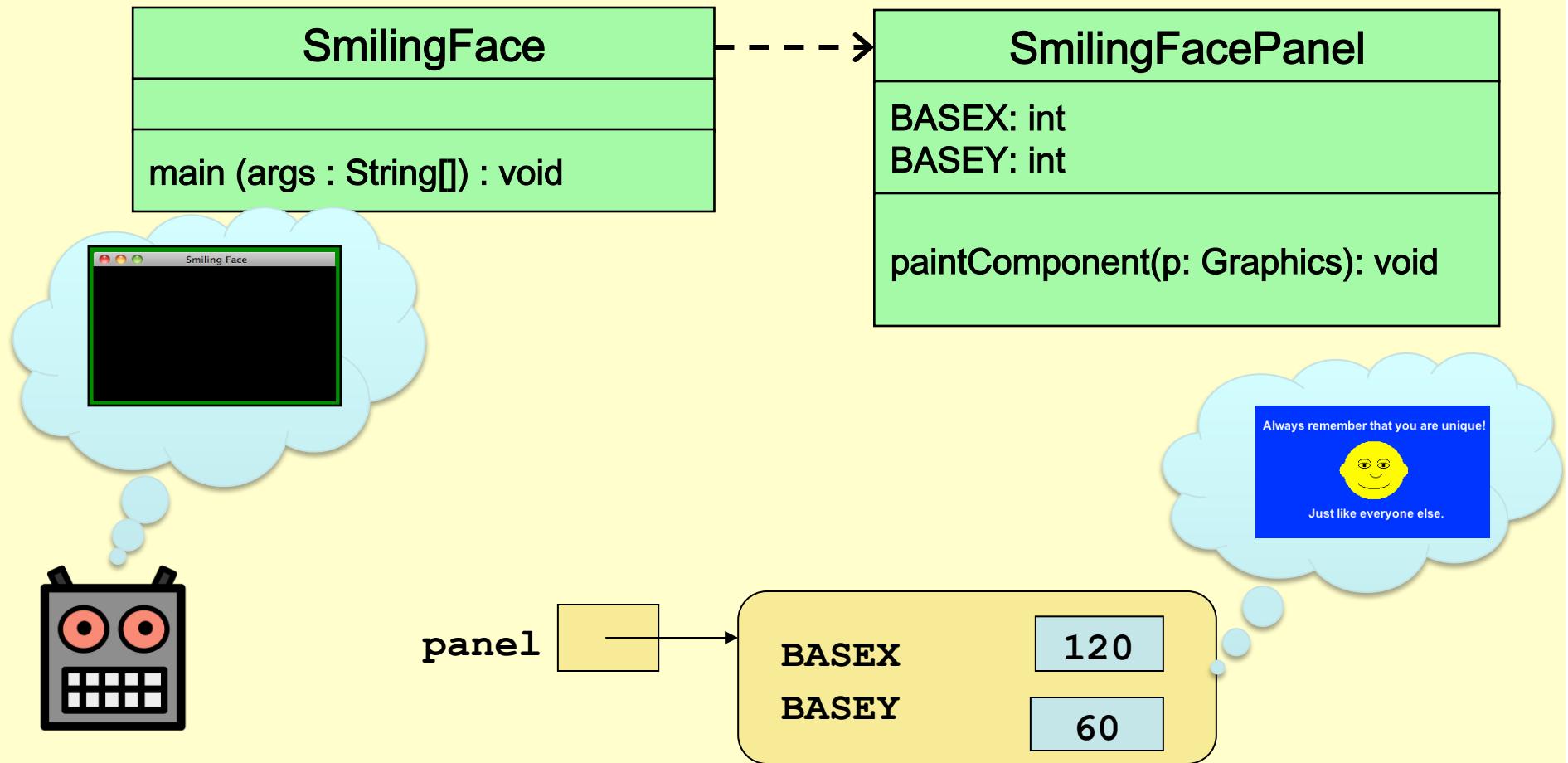
```
//*****  
// SmilingFace  
//  
// Demonstrat  
//*****  
  
import javax.s  
  
public class S  
{  
    //-----  
    // Creates  
    //-----  
    public stat  
{  
        JFrame frame = new JFrame ("Smiling Face");  
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
  
        SmilingFacePanel panel = new SmilingFacePanel();  
  
        frame.getContentPane().add(panel);  
  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```



- The SmilingFace program draws a face by defining the paintComponent method of a panel
- A UML class diagram:



- The SmilingFace program draws a face by defining the paintComponent method of a panel
- A UML class diagram:



```
//*****  
//  SmilingFacePanel.java      Author: Lewis/Loftus  
//  
//  Demonstrates the use of a separate panel class.  
//*****  
  
import javax.swing.JPanel;  
import java.awt.*;  
  
public class SmilingFacePanel extends JPanel  
{  
    private final int BASEX = 120, BASEY = 60; // base point for head  
  
    //-----  
    // Constructor: Sets up the main characteristics of this panel.  
    //-----  
    public SmilingFacePanel ()  
    {  
        setBackground (Color.blue);  
        setPreferredSize (new Dimension(320, 200));  
       setFont (new Font("Arial", Font.BOLD, 16));  
    }  
}
```

continue

```
continue
//-----
//  Draws a face.
//-----
public void paintComponent (Graphics page)
{
    super.paintComponent (page);

    page.setColor (Color.yellow);
    page.fillOval (BASEX, BASEY, 80, 80); // head
    page.fillOval (BASEX-5, BASEY+20, 90, 40); // ears

    page.setColor (Color.black);
    page.drawOval (BASEX+20, BASEY+30, 15, 7); // eyes
    page.drawOval (BASEX+45, BASEY+30, 15, 7);

    page.fillOval (BASEX+25, BASEY+31, 5, 5); // pupils
    page.fillOval (BASEX+50, BASEY+31, 5, 5);

    page.drawArc (BASEX+20, BASEY+25, 15, 7, 0, 180); // eyebrows
    page.drawArc (BASEX+45, BASEY+25, 15, 7, 0, 180);

    page.drawArc (BASEX+35, BASEY+40, 15, 10, 180, 180); // nose
    page.drawArc (BASEX+20, BASEY+50, 40, 15, 180, 180); // mouth
    page.setColor (Color.white);
    page.drawString ("Always remember that you are unique!",
                    BASEX-105, BASEY-15);
    page.drawString ("Just like everyone else.", BASEX-45, BASEY+105);
}
}
```

# Jpanel Class – let's look at Java API

The screenshot shows a Java API documentation page for the `JPanel` class. The left sidebar lists various Java classes, and the main content area displays the `JPanel` class details.

**Java™ Platform Standard Ed. 6**

[All Classes](#)

Packages

[java.applet](#)  
[java.awt](#)  
[java.awt.color](#)

[JobKOctets](#)  
[JobKOctetsProcessed](#)  
[JobKOctetsSupported](#)  
[JobMediaSheets](#)  
[JobMediaSheetsCompleted](#)  
[JobMediaSheetsSupported](#)  
[JobMessageFromOperator](#)  
[JobName](#)  
[JobOriginatingUserName](#)  
[JobPriority](#)  
[JobPrioritySupported](#)  
[JobSheets](#)  
[JobState](#)  
[JobStateReason](#)  
[JobStateReasons](#)  
[Joinable](#)  
[JoinRowSet](#)  
[JOptionPane](#)  
[JPanel](#)  
[JPasswordField](#)

**Overview Package Class Use Tree Deprecated Index Help**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [ME](#)

**javax.swing**

**Class JPanel**

`java.lang.Object`  
└ `java.awt.Component`  
  └ `java.awt.Container`  
    └ `javax.swing.JComponent`  
      └ `javax.swing.JPanel`

**All Implemented Interfaces:**

[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#)

**Direct Known Subclasses:**

[AbstractColorChooserPanel](#), [JSpinner.DefaultEditor](#)

---

```
public class JPanel
extends JComponent
implements Accessible
```

`JPanel` is a generic lightweight container. For examples and task-oriented documentation for `JPanel`, see [Tasks](#).

# Jpanel Class – let's look at Java API

Java™ Platform Standard Ed. 6

All Classes

Packages

[java.applet](#)

[java.awt](#)

[java.awt.color](#)

[JobKOctets](#)

[JobKOctetsProcessed](#)

[JobKOctetsSupported](#)

[JobMediaSheets](#)

[JobMediaSheetsCompleted](#)

[JobMediaSheetsSupported](#)

[JobMessageFromOperator](#)

[JobName](#)

[JobOriginatingUserName](#)

[JobPriority](#)

[JobPrioritySupported](#)

[JobSheets](#)

[JobState](#)

[JobStateReason](#)

[JobStateReasons](#)

[Joinable](#)

[JoinRowSet](#)

[JOptionPane](#)

[JPanel](#)

[JPasswordField](#)

[Overview](#)

[Package](#)

**Class**

[Use](#)

[Tree](#)

[Deprecated](#)

[Index](#)

[Help](#)

[PREV CLASS](#)

[NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#)

[NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [ME](#)

javax.swing

**Class JPanel**

[java.lang.Object](#)

  └ [java.awt.Component](#)

    └ [java.awt.Container](#)

      └ [javax.swing.JComponent](#)

        └ [javax.swing.JPanel](#)

**All Implemented Interfaces:**

[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#)

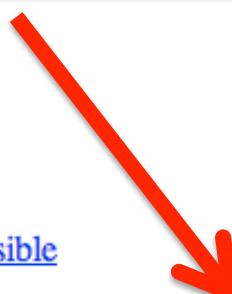
**Direct Known Subclasses:**

[AbstractColorChooserPanel](#), [JSpinner.DefaultEditor](#)

public class JPanel  
extends [JComponent](#)  
implements [Accessible](#)

JPanel is a generic lightweight container. For examples and task-oriented documentation for JPanel, see [Tasks](#).

The SmilingFacePanel class is derived from the JPanel class using inheritance

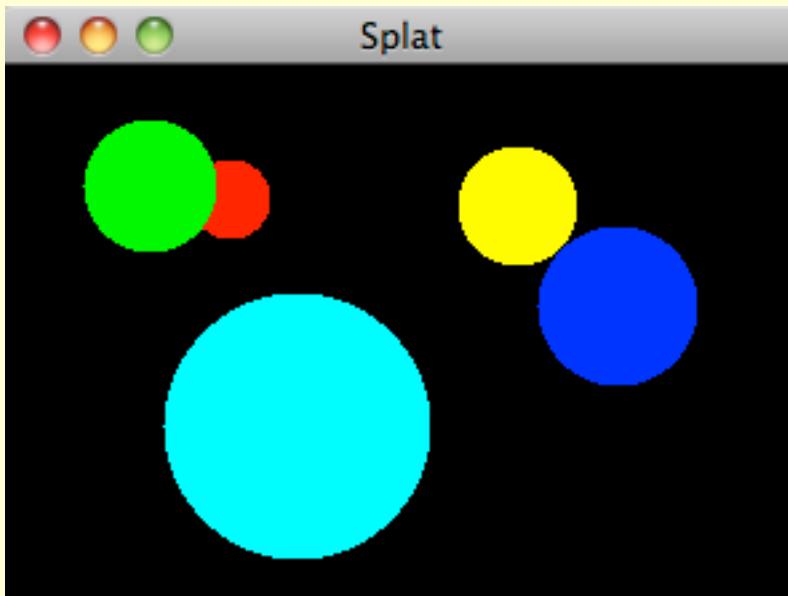


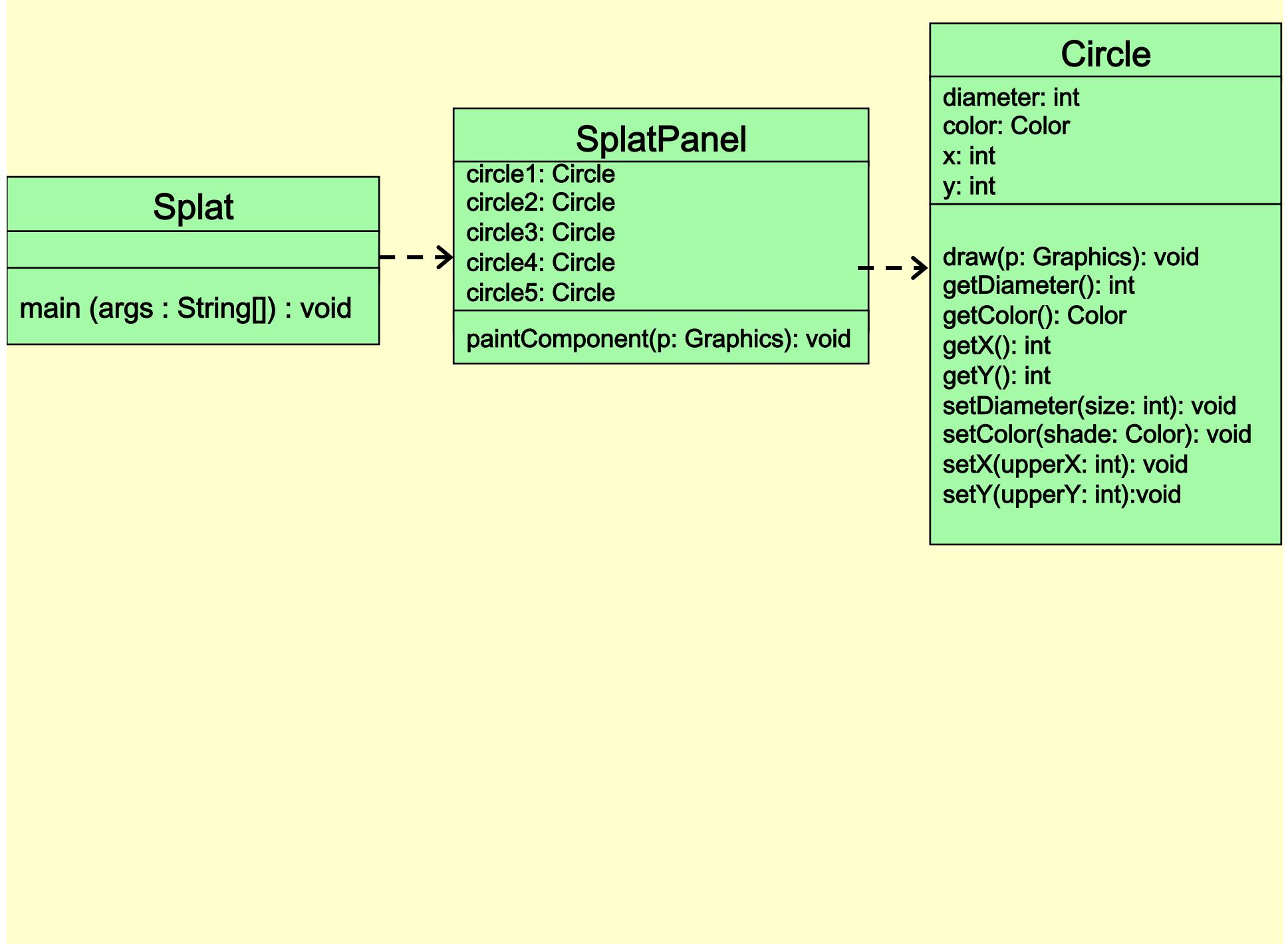
# Objects with a draw() method

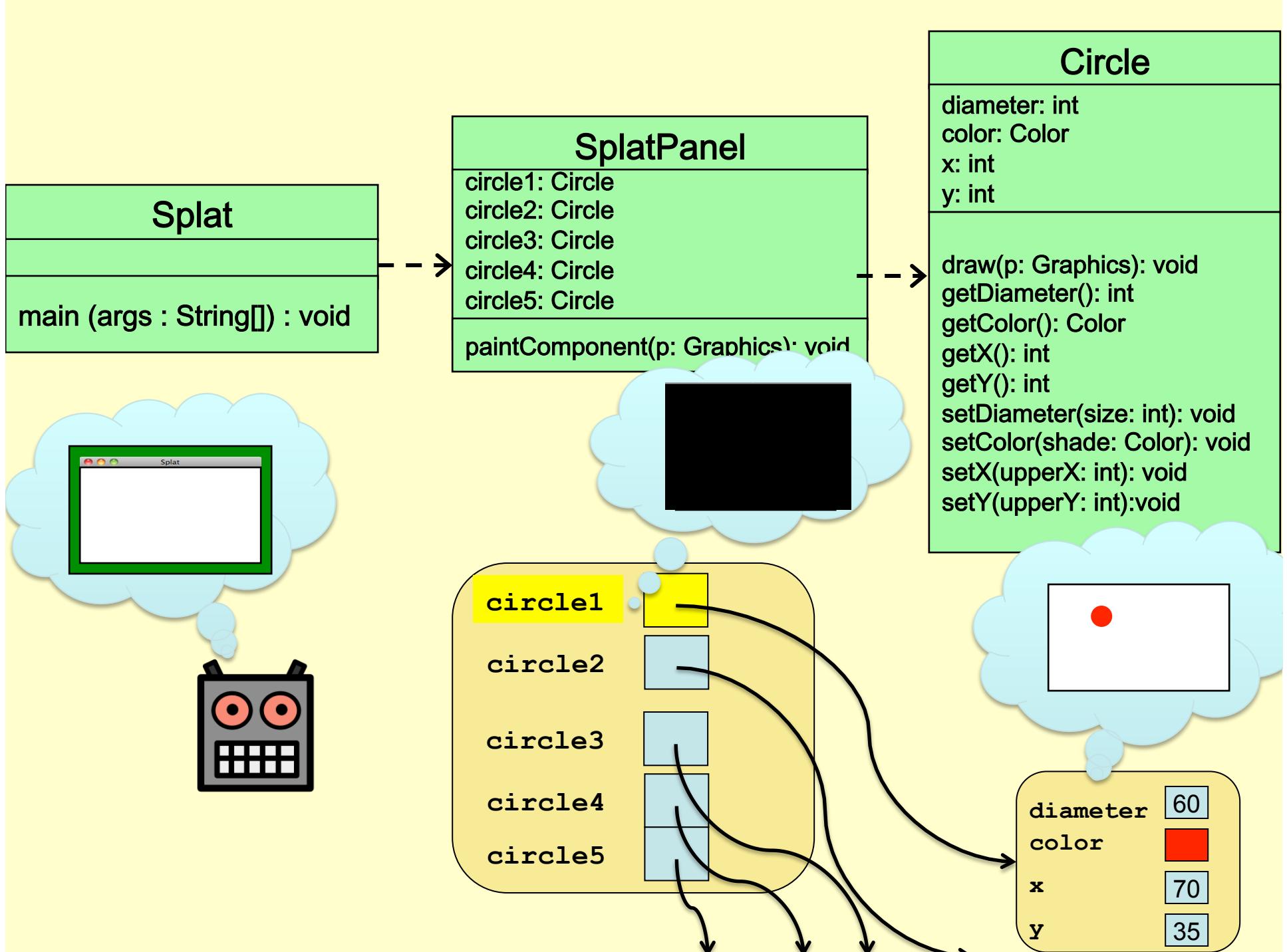
- The next example - Splat - is structured differently
- It draws a set of colored circles on a panel, but each circle is represented as a separate object that maintains its own graphical information
- The paintComponent method of the panel "asks" each circle to draw itself
- See [Splat.java](#)
- See [SplatPanel.java](#)
- See [Circle.java](#)

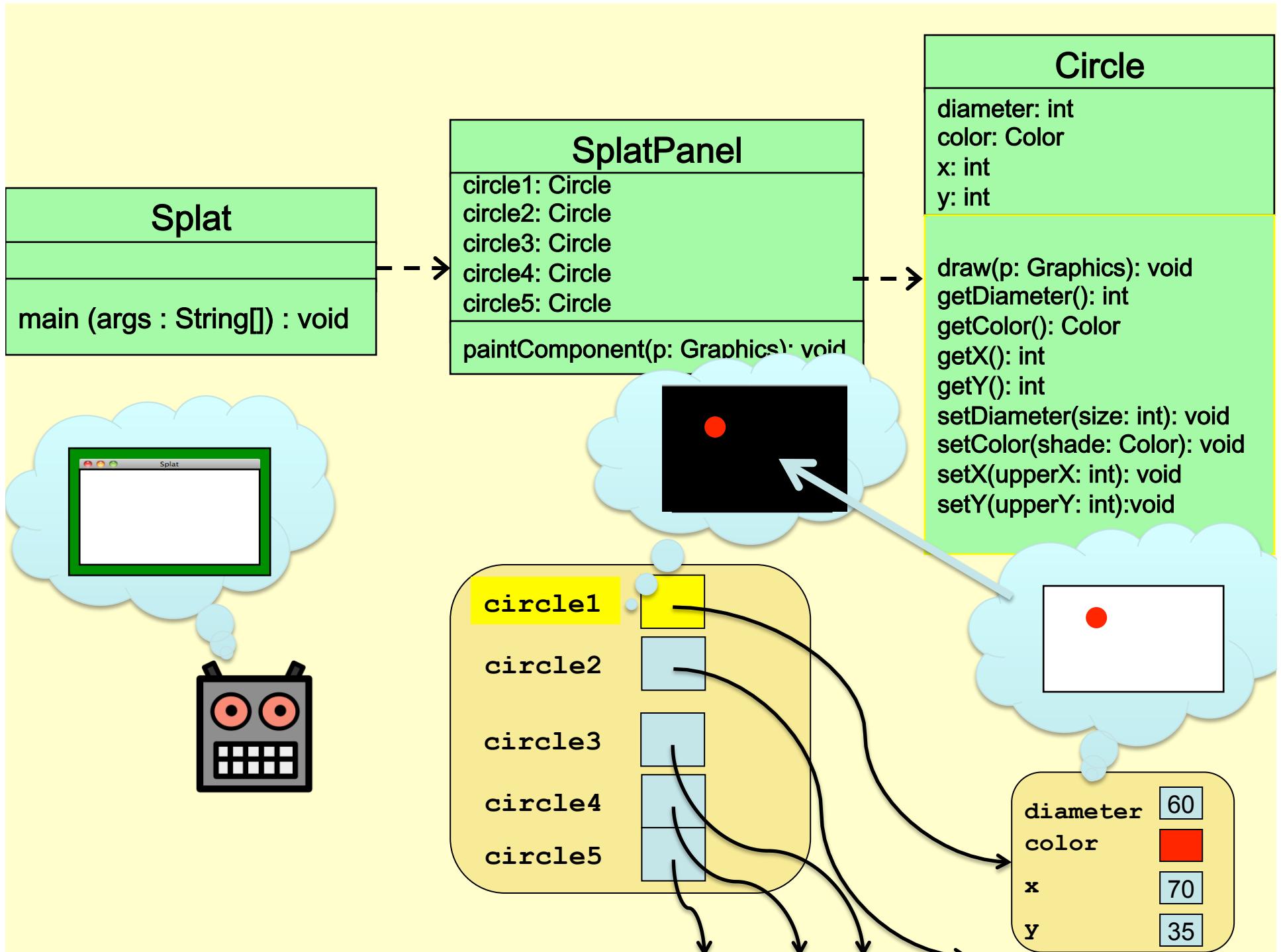
```
//*****  
// Splat.java      Author: Lewis/Loftus  
//  
// Demonstrates the use of graphical objects.  
//*****  
  
import javax.swing.*;  
import java.awt.*;  
  
public class Splat  
{  
    //-----  
    // Presents a collection of circles.  
    //-----  
    public static void main (String[] args)  
    {  
        JFrame frame = new JFrame ("Splat");  
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
  
        frame.getContentPane().add(new SplatPanel());  
  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

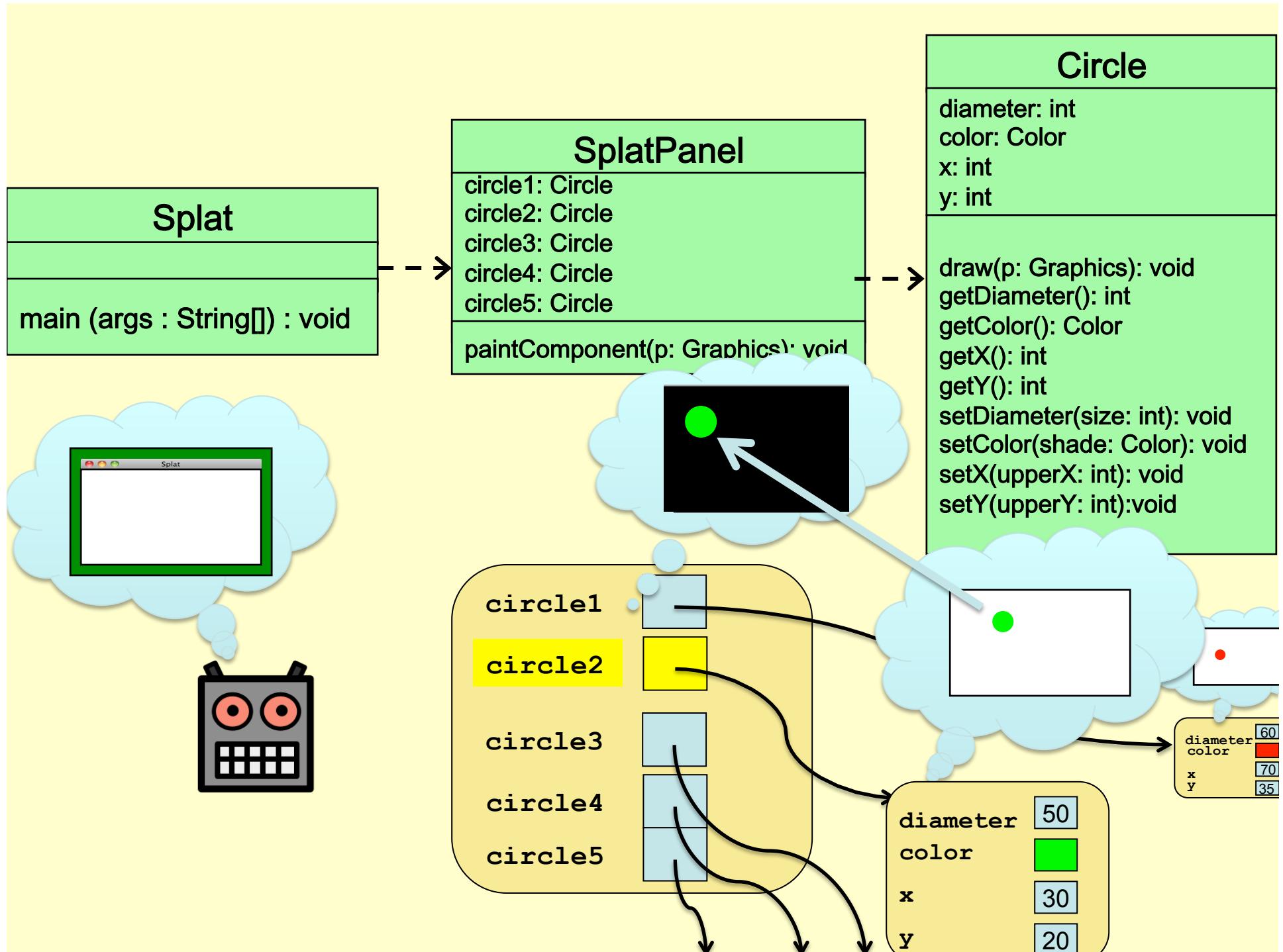
```
//*****  
// Splat.java  
//  
// Demonstrates Java's AWT graphics API.  
//*****  
  
import javax.swing.*;  
import java.awt.*;  
  
public class Splat extends JPanel  
{  
    //-----  
    // Presents  
    //-----  
  
    public static void main (String[] args)  
    {  
        JFrame frame = new JFrame ("Splat");  
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
  
        frame.getContentPane () .add (new SplatPanel());  
  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

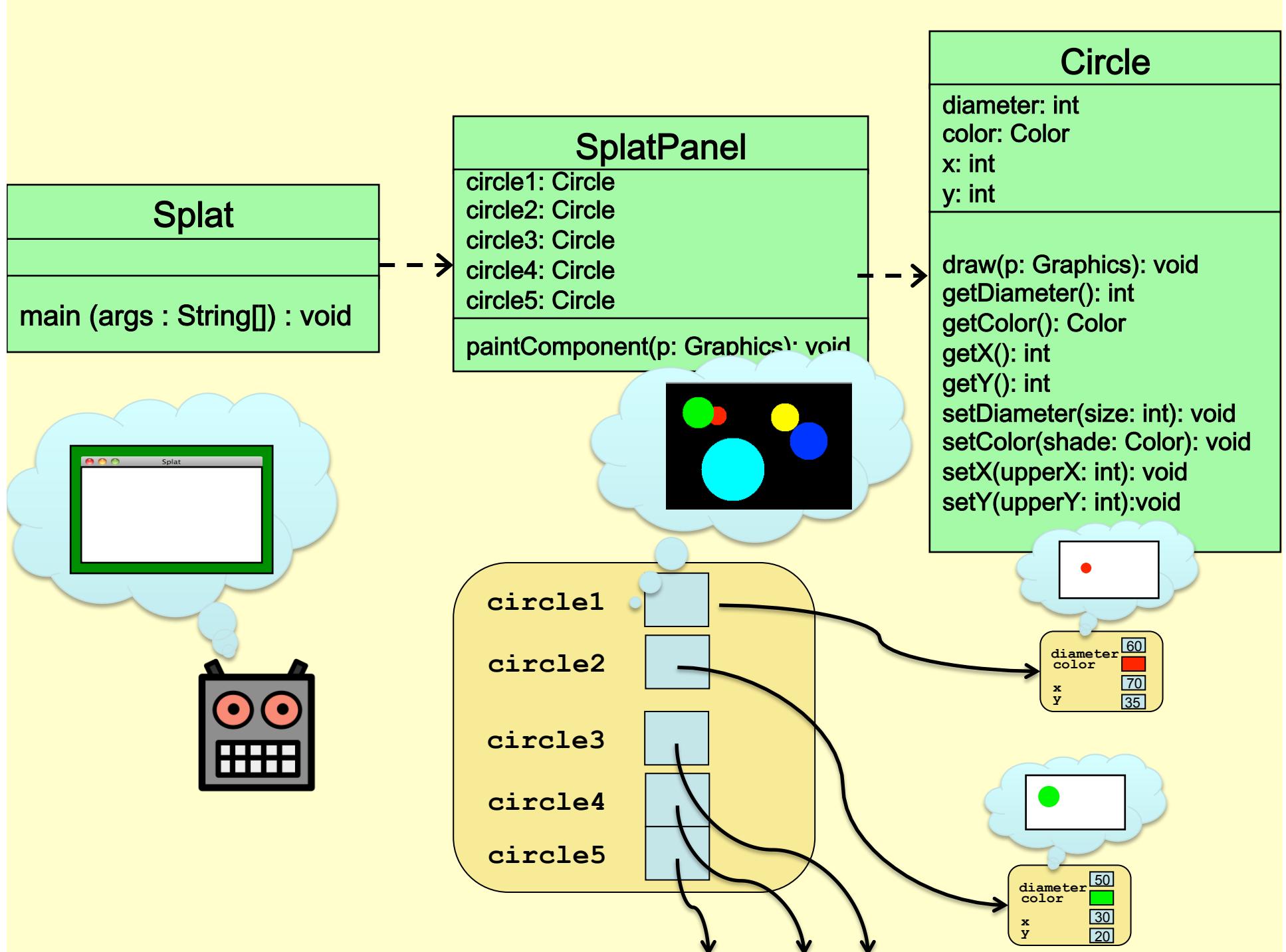












```
//*****  
// SplatPanel.java      Author: Lewis/Loftus  
//  
// Demonstrates the use of graphical objects.  
//*****  
  
import javax.swing.*;  
import java.awt.*;  
  
public class SplatPanel extends JPanel  
{  
    private Circle circle1, circle2, circle3, circle4, circle5;  
  
    //-----  
    // Constructor: Creates five Circle objects.  
    //-----  
    public SplatPanel()  
{  
        circle1 = new Circle (30, Color.red, 70, 35);  
        circle2 = new Circle (50, Color.green, 30, 20);  
        circle3 = new Circle (100, Color.cyan, 60, 85);  
        circle4 = new Circle (45, Color.yellow, 170, 30);  
        circle5 = new Circle (60, Color.blue, 200, 60);  
  
        setPreferredSize (new Dimension(300, 200));  
        setBackground (Color.black);  
    }  
}
```

continue

## continue

```
//-----  
//  Draws this panel by requesting that each circle draw itself.  
//-----  
public void paintComponent (Graphics page)  
{  
    super.paintComponent(page);  
  
    circle1.draw(page);  
    circle2.draw(page);  
    circle3.draw(page);  
    circle4.draw(page);  
    circle5.draw(page);  
}  
}
```

```
*****  
// Circle.java      Author: Lewis/Loftus  
//  
// Represents a circle with a particular position, size, and color.  
*****  
  
import java.awt.*;  
  
public class Circle  
{  
    private int diameter, x, y;  
    private Color color;  
  
    //-----  
    // Constructor: Sets up this circle with the specified values.  
    //-----  
    public Circle (int size, Color shade, int upperX, int upperY)  
    {  
        diameter = size;  
        color = shade;  
        x = upperX;  
        y = upperY;  
    }  
}
```

continue

**continue**

```
//-----  
//  Draws this circle in the specified graphics context.  
//-----  
public void draw (Graphics page)  
{  
    page.setColor (color);  
    page.fillOval (x, y, diameter, diameter);  
}  
  
//-----  
//  Diameter mutator.  
//-----  
public void setDiameter (int size)  
{  
    diameter = size;  
}  
  
//-----  
//  Color mutator.  
//-----  
public void setColor (Color shade)  
{  
    color = shade;  
}
```

**continue**

**continue**

```
//-----  
//  X mutator.  
//-----  
public void setX (int upperX)  
{  
    x = upperX;  
}  
  
//-----  
//  Y mutator.  
//-----  
public void setY (int upperY)  
{  
    y = upperY;  
}  
  
//-----  
//  Diameter accessor.  
//-----  
public int getDiameter ()  
{  
    return diameter;  
}
```

**continue**

## continue

```
//-----  
// Color accessor.  
//-----  
public Color getColor ()  
{  
    return color;  
}  
  
//-----  
// X accessor.  
//-----  
public int getX ()  
{  
    return x;  
}  
  
//-----  
// Y accessor.  
//-----  
public int getY ()  
{  
    return y;  
}  
}
```