

Graphics

CSBridge Game Show



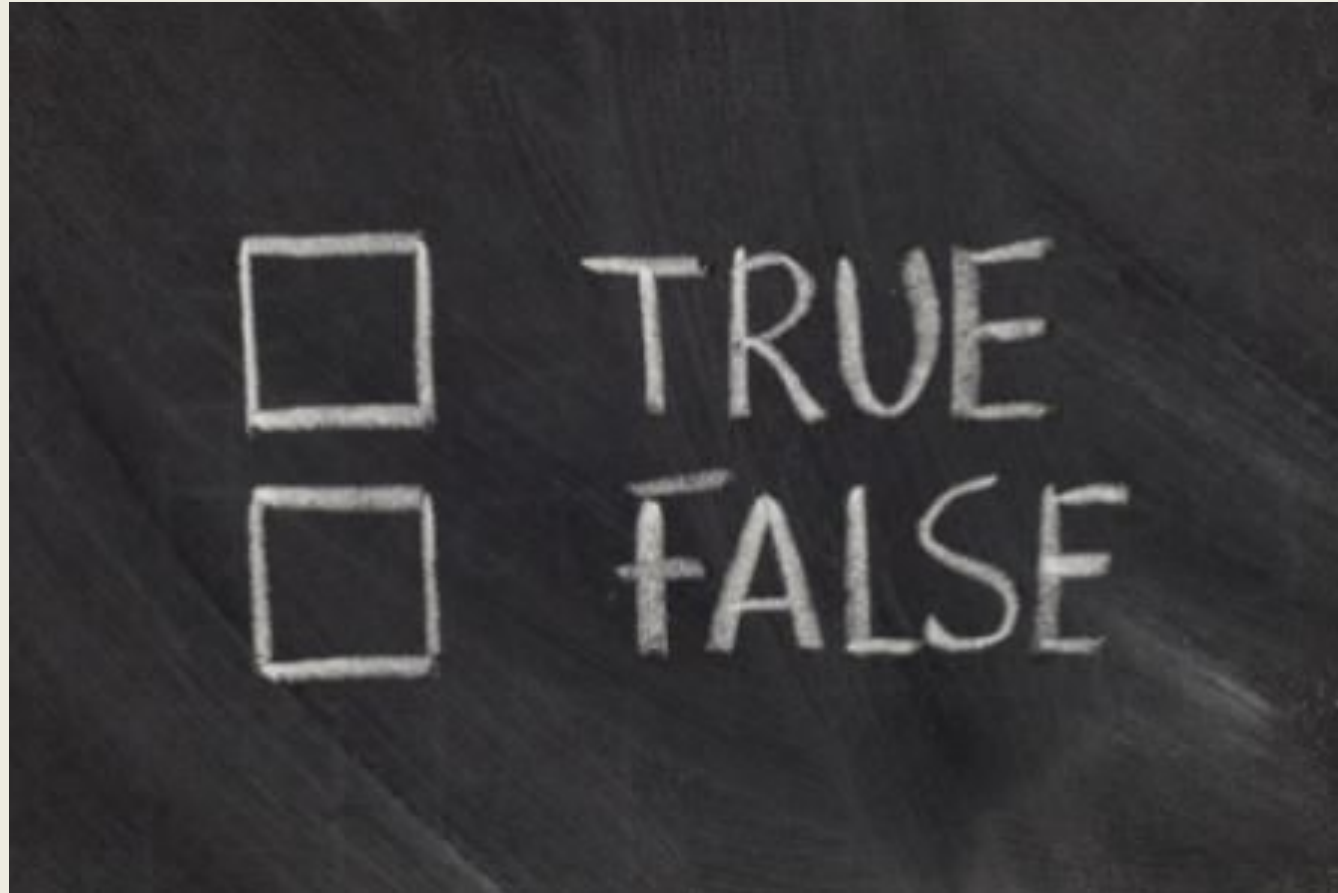
Or and And

```
// 1. Get a door number from the user
int doorChoice = readInt("Choose a door: ");
while (doorChoice < 1 || doorChoice > 3) {
    println("Choose a door number between 1 and 3");
    doorChoice = readInt("Choose a door: ");
}
```

or

	or
&&	and

Boolean



Boolean Variable

```
boolean karelIsAwesome = true;
```

```
boolean myBool = 1 < 2;
```

Boolean Operations

```
boolean a = true;
```

```
boolean b = false;
```

```
//This is false
```

```
boolean a_and_b = a && b;
```

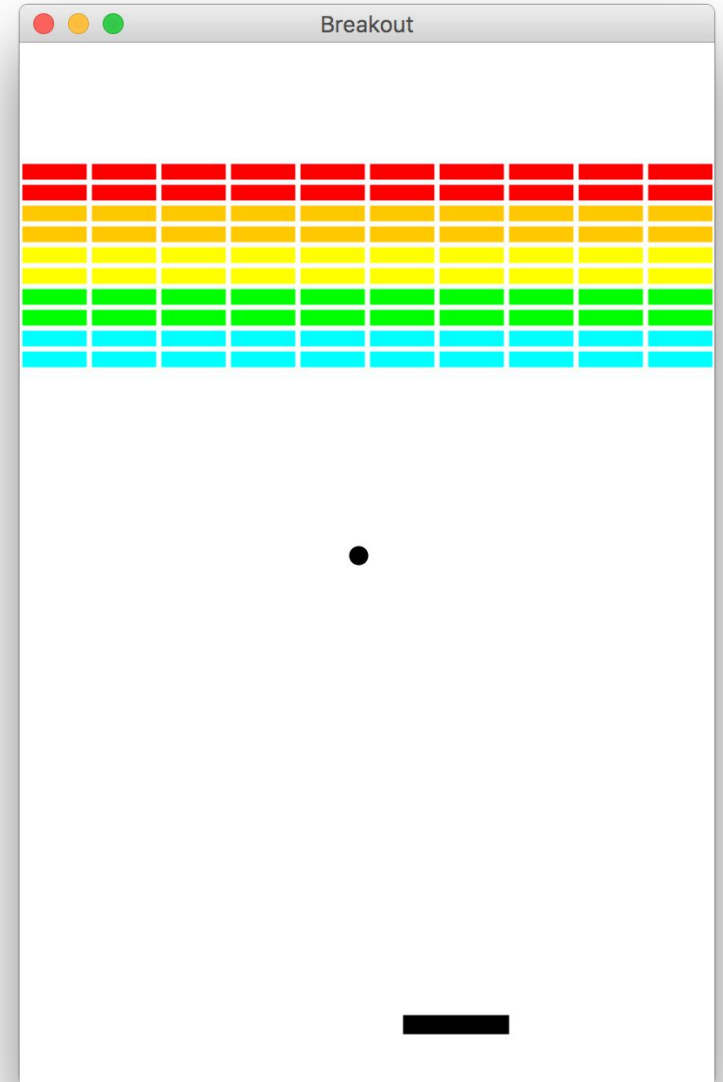
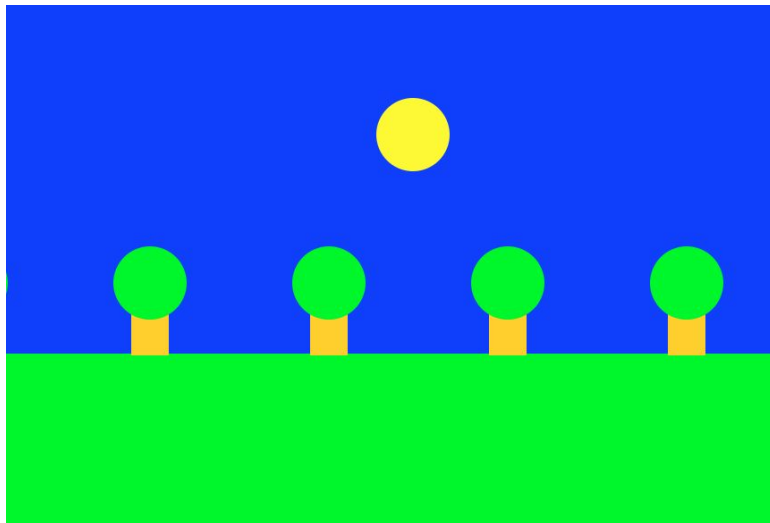
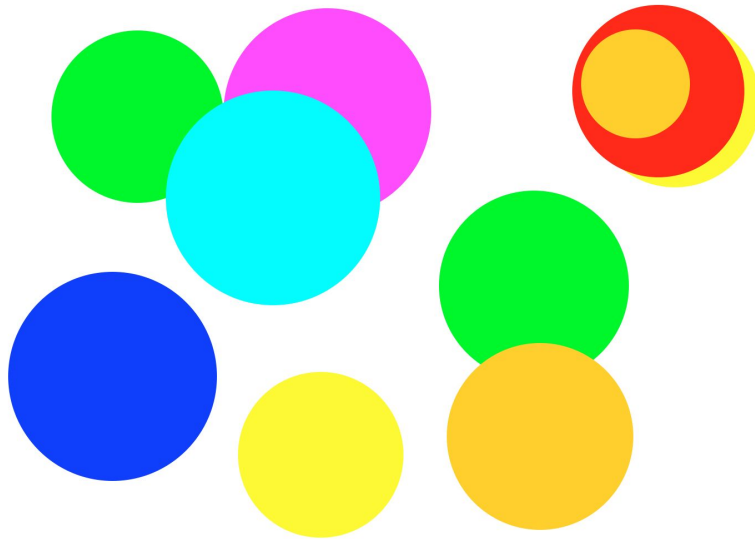
```
//This is true
```

```
boolean a_or_b = a || b;
```

```
//This is false
```

```
boolean not_a = !a;
```

Beyond Console Programs

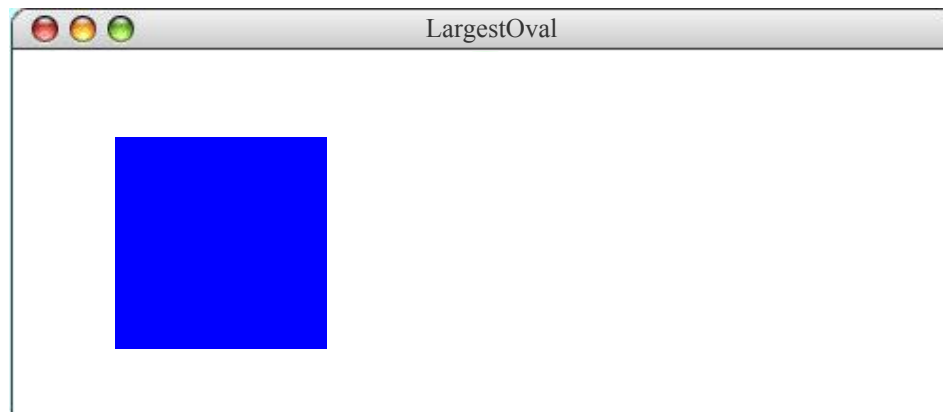


GRect

GRect is a variable that stores a rectangle.

As an example, the following `run` method displays a rectangle

```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```



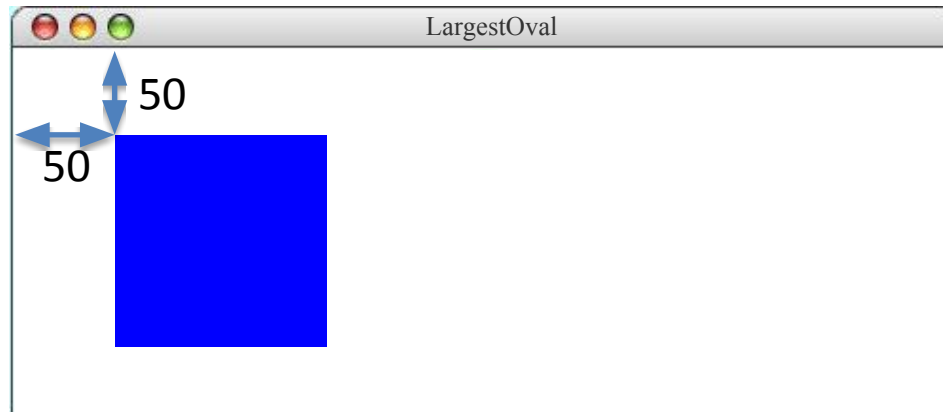
GRect

GRect is a variable that stores a rectangle.

As an example, the following `run` method displays a rectangle

```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```

Coordinates for a rectangle are the top left corner.



Graphics Coordinates

0,0

x 40,20

x 120,40

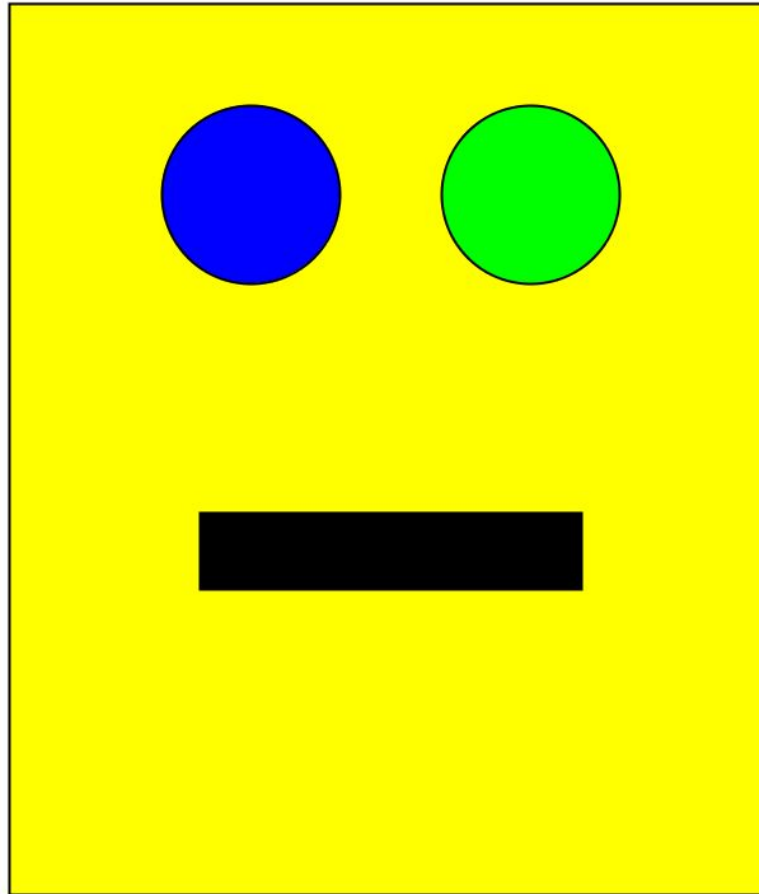
x 40,120

getWidth();

getHeight();

Karel's Grandpa

Robot Face



Karel's Grandpa

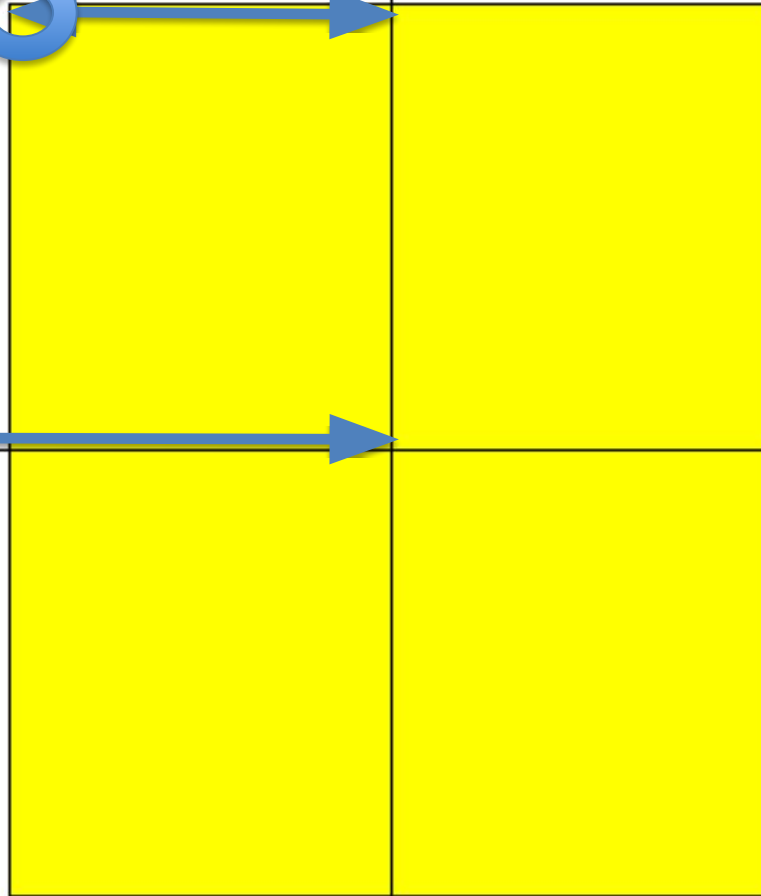


Karel's Grandpa

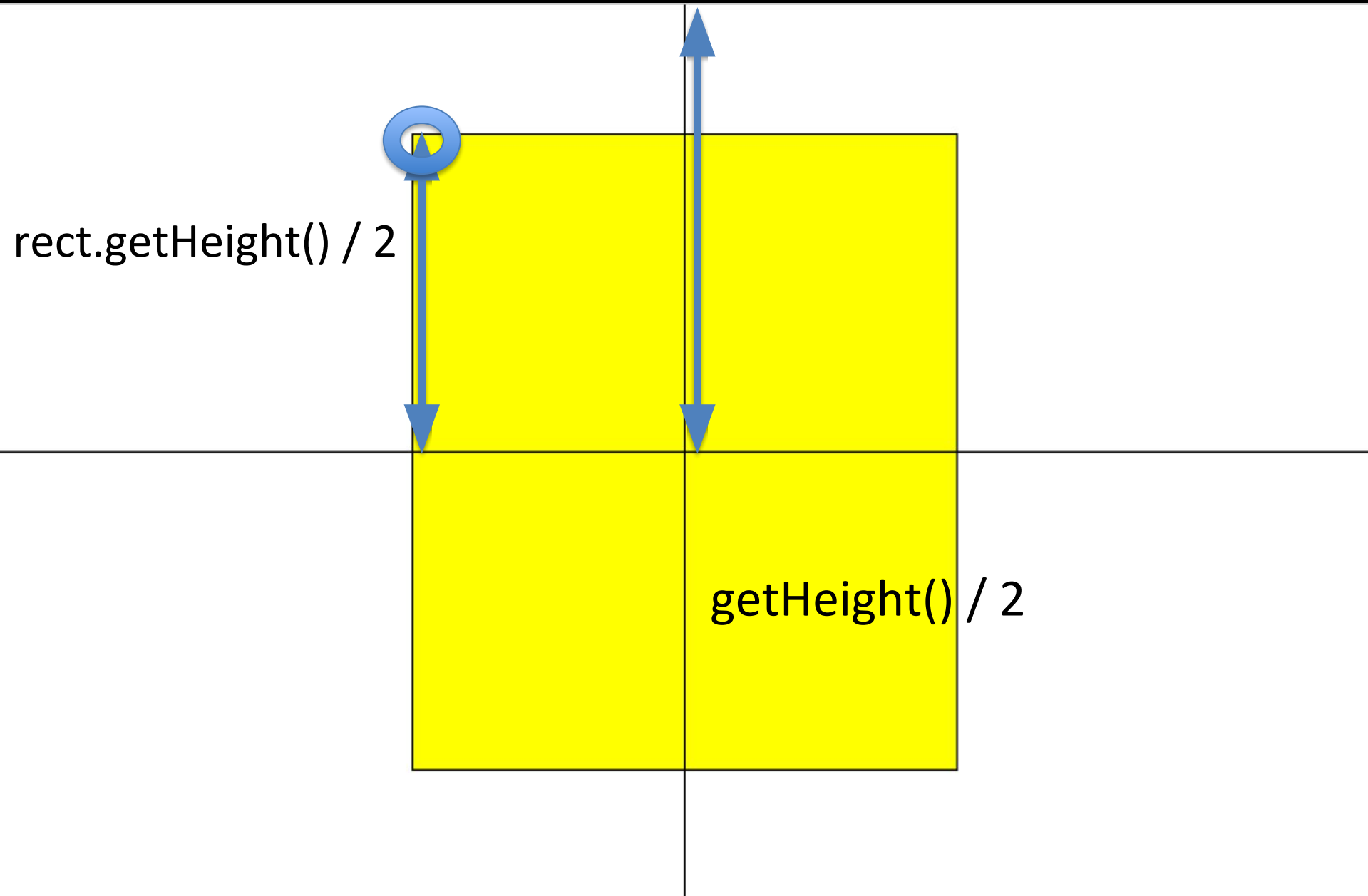
`rect.getWidth() / 2`



`getWidth() / 2`



Karel's Grandpa

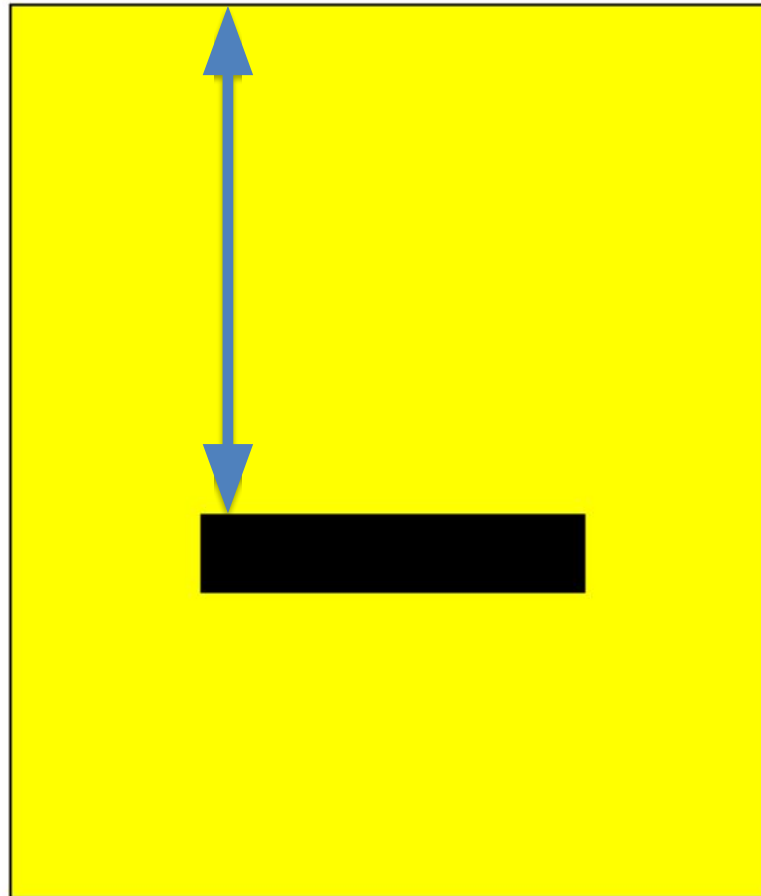


Constants

```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```

Karel's Grandpa

```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```



GLabel

A variable that represents text.

```
public class HelloProgram extends GraphicsProgram {  
    public void run() {  
        GLabel label = new GLabel("hello, world", 100,  
75);  
        label.setFont("SansSerif-36");  
        label.setColor(Color.RED);  
        add(label);  
    }  
}
```



GLabel

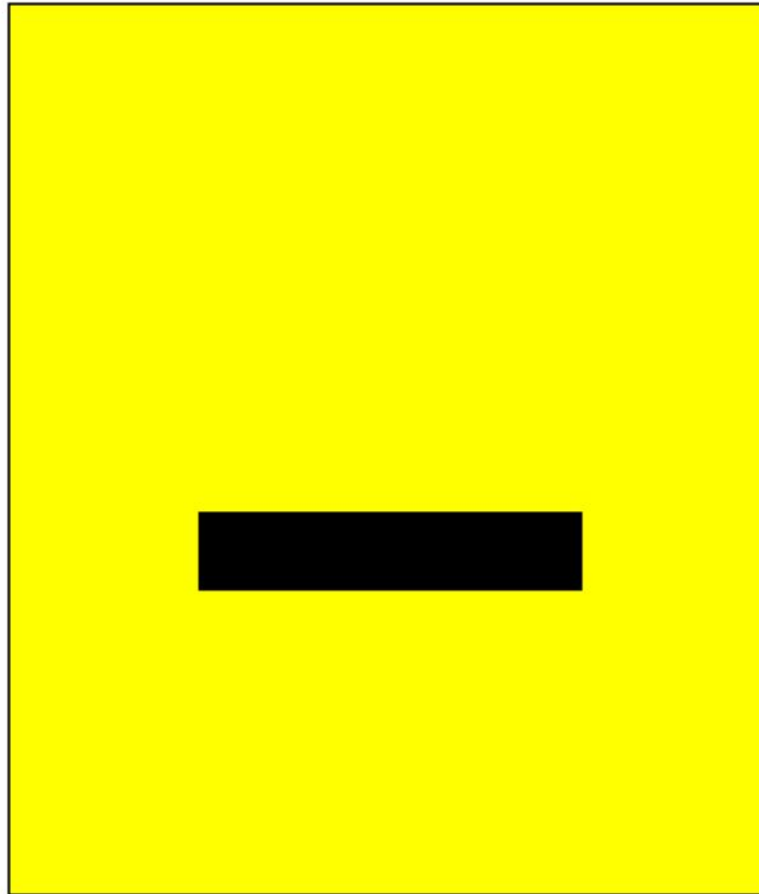
A variable that represents text.

```
public class HelloProgram extends GraphicsProgram {  
    public void run() {  
        GLabel label = new GLabel("hello, world", 100,  
75);  
        label.setFont("SansSerif-36");  
        label.setColor(Color.RED);  
        add(label);  
    }  
}
```



Karel's Grandpa

Robot Face



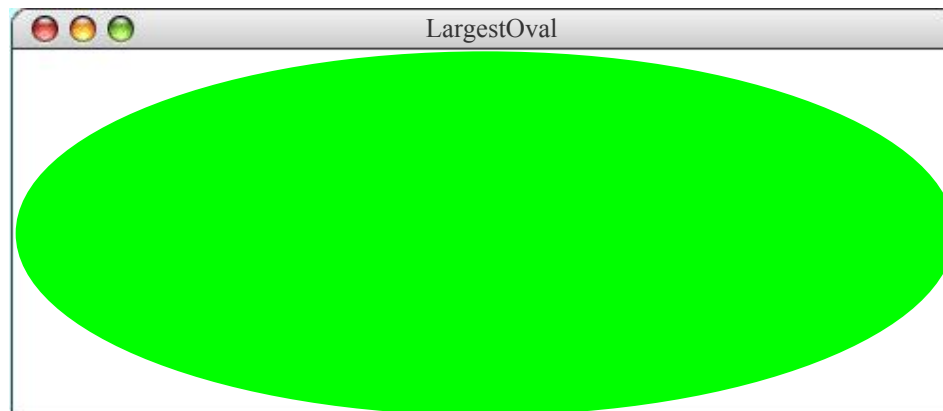
```
/*The distance from the top of the screen to the base of the label:*/  
private static final int LABEL_Y = 50;
```

GOval

The `GOval` class represents an elliptical shape defined by the boundaries of its enclosing rectangle.

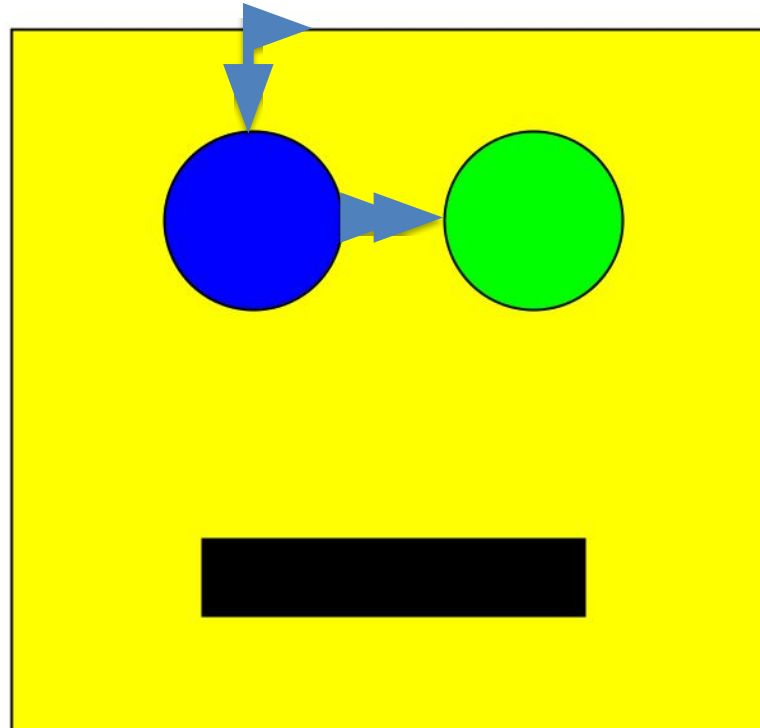
As an example, the following `run` method creates the largest oval that fits within the canvas:

```
public void run() {  
    GOval oval = new GOval(getWidth(), getHeight());  
    oval.setFilled(true);  
    oval.setColor(Color.GREEN);  
    add(oval, 0, 0);  
}
```



Karel's Grandpa

Robot Face

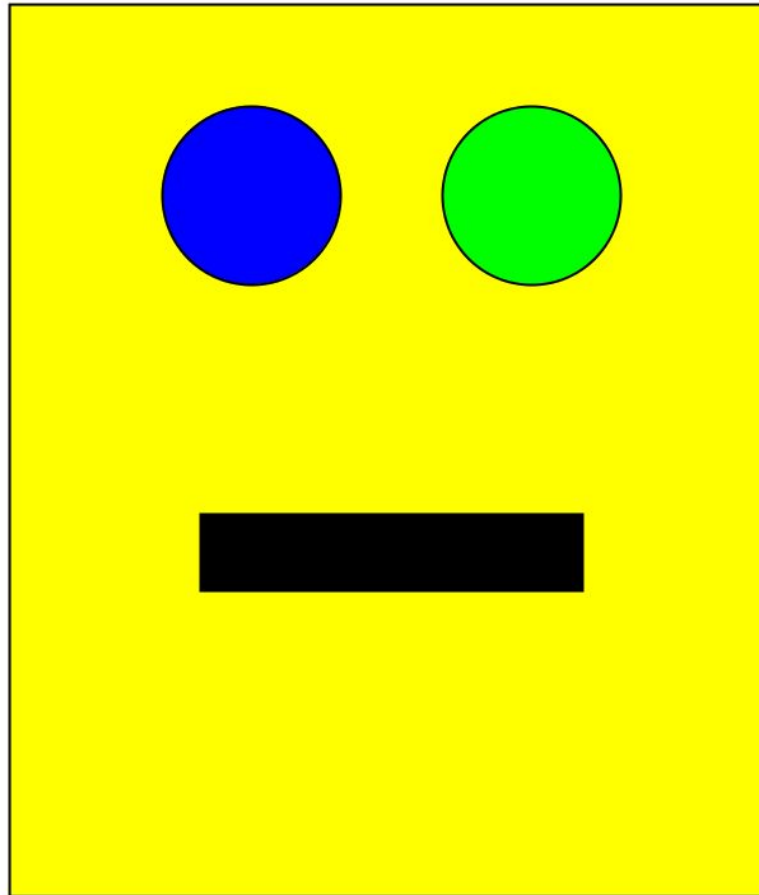


```
/*The distance from the top of the head to the top of the eyes:*/  
private static final int EYE_Y_OFFSET = 40;
```

```
/*The distance in between the two eyes:*/  
private static final int EYE_X_SEPARATION = 40;
```

Karel's Grandpa

Robot Face



Graphics Methods

<code>add (object)</code>	Adds the object to the canvas at the front of the stack
<code>add (object, x, y)</code>	Moves the object to (x, y) and then adds it to the canvas
<code>remove (object)</code>	Removes the object from the canvas
<code>removeAll ()</code>	Removes all objects from the canvas
<code>getElementAt (x, y)</code>	Returns the frontmost object at (x, y), or <code>null</code> if none
<code>getWidth ()</code>	Returns the width in pixels of the entire canvas
<code>getHeight ()</code>	Returns the height in pixels of the entire canvas
<code>setBackground (c)</code>	Sets the background color of the canvas to <i>c</i> .
<code>pause (milliseconds)</code>	Pauses the program for the specified time in milliseconds
<code>waitForClick ()</code>	Suspends the program until the user clicks the mouse

Graphics Methods

<code>add (object)</code>	Adds the object to the canvas at the front of the stack
<code>add (object, x, y)</code>	Moves the object to (x, y) and then adds it to the canvas
<code>remove (object)</code>	Removes the object from the canvas
<code>removeAll ()</code>	Removes all objects from the canvas
<code>getElementAt (x, y)</code>	Returns the frontmost object at (x, y), or <code>null</code> if none
<code>getWidth ()</code>	Returns the width in pixels of the entire canvas
<code>getHeight ()</code>	Returns the height in pixels of the entire canvas
<code>setBackground (c)</code>	Sets the background color of the canvas to <i>c</i> .
<code>pause (milliseconds)</code>	Pauses the program for the specified time in milliseconds
<code>waitForClick ()</code>	Suspends the program until the user clicks the mouse

Reference Sheet

Constructors

new GLabel(String text) or new GLabel(String text, double x, double y)

Creates a new **GLabel** object; the second form sets its location as well.

new GRect(double x, double y, double width, double height)

Creates a new **GRect** object; the **x** and **y** parameters can be omitted and default to 0.

new GOval(double x, double y, double width, double height)

Creates a new **GOval** object; the **x** and **y** parameters can be omitted and default to 0.

new GLine(double x1, double y1, double x2, double y2)

Creates a new **GLine** object connecting (**x1**, **y1**) and (**x2**, **y2**).

Methods common to all graphical objects

void setLocation(double x, double y)

Sets the location of this object to the specified coordinates.

void move(double dx, double dy)

Moves the object using the displacements **dx** and **dy**.

double getWidth()

Returns the width of the object.

double getHeight()

Returns the height of the object.

void setColor(Color c)

Sets the color of the object.

Methods available for GRect and GOval only

void setFilled(boolean fill)

Sets whether this object is filled (**true** means filled, **false** means outlined).

boolean isFilled()

Returns **true** if the object is filled.

void setFillColor(Color c)

Sets the color used to fill this object. If the color is **null**, filling uses the color of the object.

Methods available for GLabel only

void setFont(String fontName)

Sets the font, as described in Chapter 5.

double getAscent()

Returns the height above the text baseline.

double getDescent()

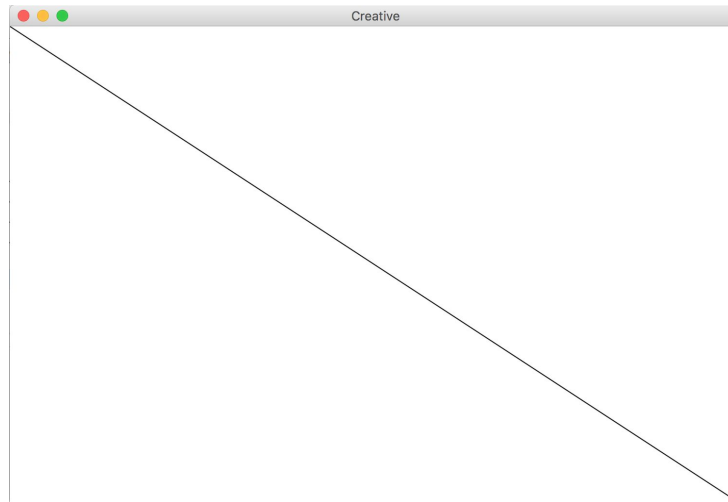
Returns the height below the text baseline.

GLine

The `GLine` class represents a line defined by a start point and an end point.

As an example, the following `run` method creates a diagonal line across the canvas:

```
public void run() {  
    GLine line = new GLine(0,0, getWidth(), getHeight());  
    add(line);  
}
```



Karel's Grandpa

