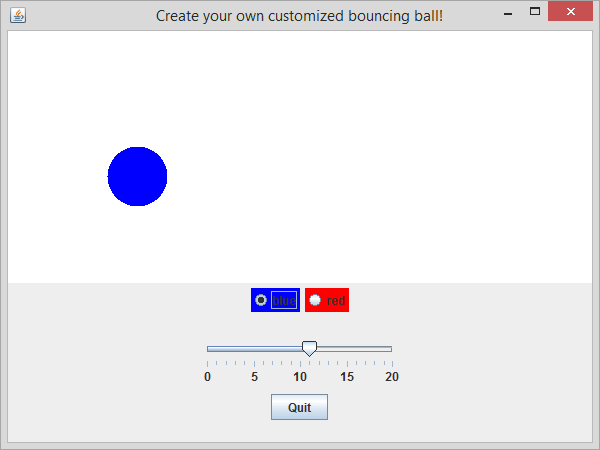
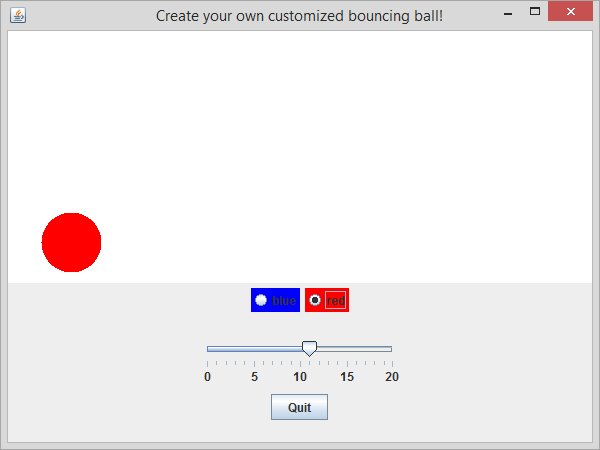
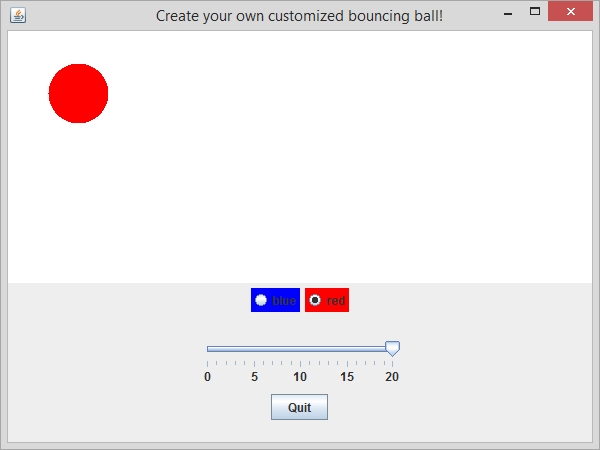
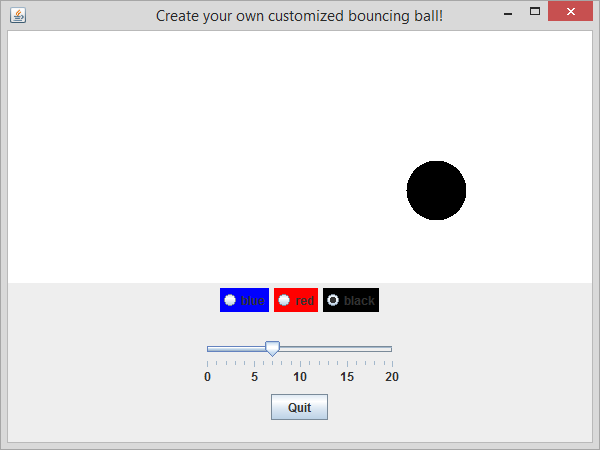
1. Enter BlueJ. Create a project with the **three** interfaces and **nine** classes listed below: Compile, run and test the project. Write the heading **Part 1** onto Word and then Paste at least three different screenshots [with different images] into Microsoft word.







1. Modify the code to add a color button for a predefined color (other than blue or red). [Hint: look at the getColorName method in the class ColorButton.] Write the heading “**Part 2”** in Word, and paste all the class(es) that you modified code, along with at least one screenshot that shows the ball in the color you added.



/\*\*

\* ControlPanel.java

\* Contain buttons that control the bouncing ball's color and speed.

\*/

public class ControlPanel extends javax.swing.JPanel {

private final int MAX\_SPEED = 20; // attribute

public ControlPanel (BallPanel aPanel) {

super(new java.awt.GridLayout(0, 1));

// create radio buttons

javax.swing.ButtonGroup group = new

javax.swing.ButtonGroup(); //2a

Colorable colorable = aPanel.getBall();

ColorButton redButton = new

ColorButton(java.awt.Color.red, colorable,

group, true);

ColorButton blueButton = new

ColorButton(java.awt.Color.blue, colorable,

group, false);

ColorButton blackButton = new ColorButton(java.awt.Color.black, colorable, group, false);

javax.swing.JPanel radioRow =

new javax.swing.JPanel(); //2b

radioRow.add(blueButton); //2c

radioRow.add(redButton);

radioRow.add(blackButton);

// create slider

Accelerator accelerator = aPanel.getBall();

javax.swing.JPanel sliderRow = new

javax.swing.JPanel();

sliderRow.add(new

SpeedSlider(javax.swing.JSlider.HORIZONTAL,

accelerator, 0, MAX\_SPEED, 0));

// create quit button

javax.swing.JPanel quitRow =

new javax.swing.JPanel();

quitRow.add(new QuitButton());

// put it all together

this.add(radioRow);

this.add(sliderRow);

this.add(quitRow);

}

}

1. Modify the code to add a color button for a color that is NOT predefined in Java. Write the heading “**Part 3”** in Word, and paste all the class(es) that you modified code, along with at least one screenshot that shows the ball in the color you added.

/\*\*

\* ColorButton.java

\* Controls the color of some Colorable object.

\* Illustrates the event-handler recipe.

\*/

public class ColorButton extends javax.swing.JRadioButton {

private java.awt.Color \_color; // attribute

private Colorable \_colorable; // peer object

public ColorButton (java.awt.Color aColor,

Colorable aColorable,

javax.swing.ButtonGroup aGroup,

boolean isSelected) {

super("", isSelected);

\_color = aColor;

\_colorable = aColorable;

this.setBackground(aColor); // make button display its color

this.setText(this.getColorName(\_color));

this.addActionListener(new ColorListener());

aGroup.add(this);

}

private class ColorListener implements java.awt.event.ActionListener {

public void actionPerformed (java.awt.event.ActionEvent e) {

\_colorable.setColor(\_color);

}

}

public String getColorName (java.awt.Color aColor) {

if (aColor == java.awt.Color.black) return "black";

else if (aColor == java.awt.Color.blue) return "blue";

else if (aColor == java.awt.Color.darkGray) return "dark gray";

else if (aColor == java.awt.Color.gray) return "gray";

else if (aColor == java.awt.Color.green) return "green";

else if (aColor == java.awt.Color.lightGray) return "light gray";

else if (aColor == java.awt.Color.magenta) return "magenta";

else if (aColor == java.awt.Color.orange) return "orange";

else if (aColor == java.awt.Color.pink) return "pink";

else if (aColor == java.awt.Color.red) return "red";

else if (aColor == java.awt.Color.white) return "white";

else if (aColor == java.awt.Color.yellow) return "yellow";

else {

return "random";

}// not one of the built-in colors

}

}

/\*\*

\* ControlPanel.java

\* Contain buttons that control the bouncing ball's color and speed.

\*/

import java.awt.Color;

public class ControlPanel extends javax.swing.JPanel {

private final int MAX\_SPEED = 20; // attribute

public ControlPanel (BallPanel aPanel) {

super(new java.awt.GridLayout(0, 1));

// create radio buttons

javax.swing.ButtonGroup group = new

javax.swing.ButtonGroup(); //2a

Colorable colorable = aPanel.getBall();

ColorButton redButton = new

ColorButton(java.awt.Color.red, colorable,

group, true);

ColorButton blueButton = new

ColorButton(java.awt.Color.blue, colorable,

group, false);

ColorButton blackButton = new ColorButton(java.awt.Color.black, colorable, group, false);

ColorButton randomButton = new ColorButton(new Color(155,0,255), colorable, group, false);

javax.swing.JPanel radioRow =

new javax.swing.JPanel(); //2b

radioRow.add(blueButton); //2c

radioRow.add(redButton);

radioRow.add(blackButton);

radioRow.add(randomButton);

// create slider

Accelerator accelerator = aPanel.getBall();

javax.swing.JPanel sliderRow = new

javax.swing.JPanel();

sliderRow.add(new

SpeedSlider(javax.swing.JSlider.HORIZONTAL,

accelerator, 0, MAX\_SPEED, 0));

// create quit button

javax.swing.JPanel quitRow =

new javax.swing.JPanel();

quitRow.add(new QuitButton());

// put it all together

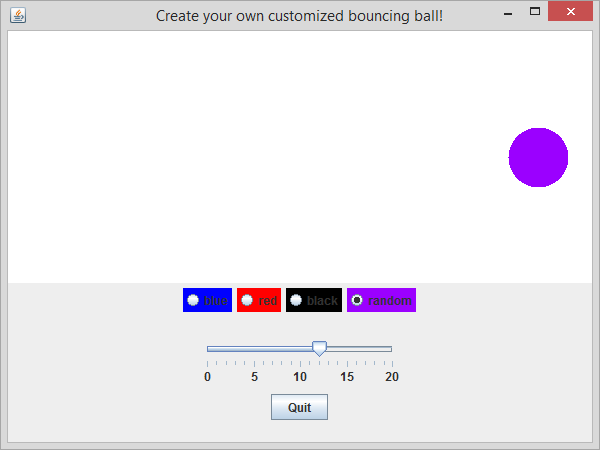
this.add(radioRow);

this.add(sliderRow);

this.add(quitRow);

}

}



1. Modify the code to put the control panel on the right side (instead of the bottom), and to change the maximum speed so that the ball can move faster. Write the heading “**Part 4”** in Word, and paste all the class(es) that you modified code, along with at least one screenshot that shows the ball in the color you added.

/\*\*

\* ControlPanel.java

\* Contain buttons that control the bouncing ball's color and speed.

\*/

import java.awt.Color;

public class ControlPanel extends javax.swing.JPanel {

private final int MAX\_SPEED = 40; // attribute

public ControlPanel (BallPanel aPanel) {

super(new java.awt.GridLayout(0, 1));

// create radio buttons

javax.swing.ButtonGroup group = new

javax.swing.ButtonGroup(); //2a

Colorable colorable = aPanel.getBall();

ColorButton redButton = new

ColorButton(java.awt.Color.red, colorable,

group, true);

ColorButton blueButton = new

ColorButton(java.awt.Color.blue, colorable,

group, false);

ColorButton blackButton = new ColorButton(java.awt.Color.black, colorable, group, false);

ColorButton randomButton = new ColorButton(new Color(155,0,255), colorable, group, false);

javax.swing.JPanel radioRow =

new javax.swing.JPanel(); //2b

radioRow.add(blueButton); //2c

radioRow.add(redButton);

radioRow.add(blackButton);

radioRow.add(randomButton);

// create slider

Accelerator accelerator = aPanel.getBall();

javax.swing.JPanel sliderRow = new

javax.swing.JPanel();

sliderRow.add(new

SpeedSlider(javax.swing.JSlider.HORIZONTAL,

accelerator, 0, MAX\_SPEED, 0));

// create quit button

javax.swing.JPanel quitRow =

new javax.swing.JPanel();

quitRow.add(new QuitButton());

// put it all together

this.add(radioRow);

this.add(sliderRow);

this.add(quitRow);

}

}

/\*\*

\* GUIBounceApp.java

\* Window that contains two panels, one that displays a bouncing ball

\* and one that holds control buttons.

\*/

public class GUIBounceApp extends javax.swing.JFrame {

public GUIBounceApp (String title) {

super(title);

this.setSize(600, 450);

BallPanel ballPanel = new BallPanel();

ControlPanel controlPanel = new

ControlPanel(ballPanel);

this.add(ballPanel,

java.awt.BorderLayout.CENTER);

this.add(controlPanel,

java.awt.BorderLayout.EAST);

this.setDefaultCloseOperation(

javax.swing.JFrame.EXIT\_ON\_CLOSE);

this.setVisible(true);

}

public static void main (String [ ] args) {

GUIBounceApp app = new GUIBounceApp(

"Create your own customized bouncing ball!");

}

}

