```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;public class GUIPizza
{
// window frame
private JFrame frame;
private JPanel contentPane;    // radio buttons and button group
private JRadioButton regularCrustButton;
private JRadioButton thinCrustButton;
private JRadioButton handCrustButton;
private JRadioButton deepCrustButton;
private ButtonGroup crustButtonGroup;    // check boxes
private JCheckBox pepperoniBox;
private JCheckBox sausageBox;
private JCheckBox cheeseBox;
private JCheckBox pepperBox;
private JCheckBox onionBox;
private JCheckBox mushroomBox;
private JCheckBox oliveBox;
private JCheckBox anchovyBox;    // text fields
private JTextField breadSticksText;
private JTextField buffaloWingsText;
private JTextField nameText;
private JTextField addressText;
private JTextField cityText;public static void main (String[] args)
    {
        GUIPizza gui = new GUIPizza();
gui.start();
    }public void start()
    {
frame = new JFrame("GUI Pizza");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);makeMenus();
makeContent();frame.pack();
frame.setVisible(true);
    }private void makeMenus()
    {
        JMenuBar menuBar;menuBar = new JMenuBar();
frame.setJMenuBar(menuBar);        // set up menus
menuBar.add(makeFileMenu());
menuBar.add(makeHelpMenu());
}    private JMenu makeFileMenu()
    {
        JMenu menu;
        JMenuItem menuItem;// set up the File menu
menu = new JMenu("File");
menu.setMnemonic(KeyEvent.VK_F);        // add New menu item
menuItem = new JMenuItem("New Order");
menuItem.setMnemonic(KeyEvent.VK_N);
menuItem.addActionListener(new NewListener());
menuItem.setAccelerator(
KeyStroke.getKeyStroke(KeyEvent.VK_N,
                                    Event.CTRL_MASK));
menu.add(menuItem);        // add Save menu item
menuItem = new JMenuItem("Save Order");
menuItem.setMnemonic(KeyEvent.VK_S);
menuItem.addActionListener(new SaveListener());
menuItem.setAccelerator(
KeyStroke.getKeyStroke(KeyEvent.VK_S,
                                    Event.CTRL_MASK));
menu.add(menuItem);        // add Exit menu item
menu.addSeparator();
menuItem = new JMenuItem("Exit");
menuItem.setMnemonic(KeyEvent.VK_X);
menuItem.addActionListener(new ExitListener());
menuItem.setAccelerator(
KeyStroke.getKeyStroke(KeyEvent.VK_Q,
                                    Event.CTRL_MASK));
menu.add(menuItem);return menu;
    }private JMenu makeHelpMenu()
    {
        JMenu menu;
        JMenuItem menuItem;        // set up the Help menu
menu = new JMenu("Help");
menu.setMnemonic(KeyEvent.VK_H);        // add About menu item
menuItem = new JMenuItem("About GUI Pizza");
menuItem.setMnemonic(KeyEvent.VK_A);
menuItem.addActionListener(new AboutListener());
menu.add(menuItem);return menu;
    }private void makeContent()
    {
contentPane = (JPanel)frame.getContentPane();
contentPane.setLayout(new BorderLayout(6,6));
contentPane.setBorder(BorderFactory.createEmptyBorder(6,6,6,6));makeNorthRegion();
makeWestRegion();
makeCenterRegion();
makeEastRegion();
makeSouthRegion();
    }private void makeNorthRegion()
    {
        // get and display image in NORTH region
        JLabel imgLabel = new JLabel(new ImageIcon("L08-06.jpg"), JLabel.CENTER);
contentPane.add(imgLabel, BorderLayout.NORTH);
```

```
    }private void makeWestRegion()
    {
        // set up Crust options with radio buttons in WEST
        JPanel panel = new JPanel();
panel.setLayout(new BoxLayout(panel,BoxLayout.Y_AXIS));
panel.setBorder(BorderFactory.createTitledBorder("Choose a Crust"));crustButtonGroup = new ButtonGroup();regularCrustButton = new JRadioButto
crustButtonGroup.add(regularCrustButton);
panel.add(regularCrustButton);thinCrustButton = new JRadioButton("Thin Crust",false);
crustButtonGroup.add(thinCrustButton);
panel.add(thinCrustButton);handCrustButton = new JRadioButton("Hand-Tossed Crust",false);
crustButtonGroup.add(handCrustButton);
panel.add(handCrustButton);deepCrustButton = new JRadioButton("Deep-Dish Crust",false);
crustButtonGroup.add(deepCrustButton);
panel.add(deepCrustButton);contentPane.add(panel, BorderLayout.WEST);
    }private void makeCenterRegion()
    {
        // set up toppings with check boxes in CENTER
        JPanel panel = new JPanel();
panel.setLayout(new BoxLayout(panel,BoxLayout.Y_AXIS));
panel.setBorder(BorderFactory.createTitledBorder("Select Toppings"));pepperoniBox = new JCheckBox("Pepperoni", false);
panel.add(pepperoniBox);
sausageBox = new JCheckBox("Sausage", false);
panel.add(sausageBox);
cheeseBox = new JCheckBox("Extra Cheese", false);
panel.add(cheeseBox);
pepperBox = new JCheckBox("Bell Peppers", false);
panel.add(pepperBox);
onionBox = new JCheckBox("Onions", false);
panel.add(onionBox);
mushroomBox = new JCheckBox("Mushrooms", false);
panel.add(mushroomBox);
oliveBox = new JCheckBox("Olives", false);
panel.add(oliveBox);
anchovyBox = new JCheckBox("Anchovies", false);
panel.add(anchovyBox);
contentPane.add(panel, BorderLayout.CENTER);
    }private void makeEastRegion()
    {
        // set up side orders with quantities in EAST
        JPanel panel = new JPanel();
panel.setLayout(new BoxLayout(panel,BoxLayout.Y_AXIS));
panel.setBorder(BorderFactory.createTitledBorder("Sides (Enter Quantity)"));
panel.setPreferredSize(new Dimension(150,0));        JPanel smallPanel = new JPanel();
smallPanel.setLayout(new BoxLayout(smallPanel,BoxLayout.X_AXIS));
breadSticksText = new JTextField("");
breadSticksText.setMaximumSize(new Dimension(20,24));
breadSticksText.setHorizontalAlignment(JTextField.RIGHT);
smallPanel.add(breadSticksText);
smallPanel.add(new JLabel(" Bread Sticks"));
smallPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
panel.add(smallPanel);smallPanel = new JPanel();
smallPanel.setLayout(new BoxLayout(smallPanel,BoxLayout.X_AXIS));
buffaloWingsText = new JTextField("");
buffaloWingsText.setMaximumSize(new Dimension(20,24));
buffaloWingsText.setHorizontalAlignment(JTextField.RIGHT);
smallPanel.add(buffaloWingsText);
smallPanel.add(new JLabel(" Buffalo Wings"));
smallPanel.setAlignmentX(Component.LEFT_ALIGNMENT);
panel.add(smallPanel);
contentPane.add(panel, BorderLayout.EAST);
    }private void makeSouthRegion()
    {
        // set up delivery address in SOUTH
        JPanel panel = new JPanel();
panel.setLayout(new BoxLayout(panel,BoxLayout.X_AXIS));
panel.setBorder(BorderFactory.createTitledBorder("Deliver To:"));
        JPanel smallPanel = new JPanel();
smallPanel.setLayout(new BoxLayout(smallPanel,BoxLayout.Y_AXIS));
smallPanel.setBorder(BorderFactory.createEmptyBorder(3,3,3,3));
smallPanel.add(new JLabel("Name:"));
smallPanel.add(new JLabel("Address:"));
smallPanel.add(new JLabel("City, St, Zip:"));
panel.add(smallPanel);
smallPanel = new JPanel();
smallPanel.setLayout(new BoxLayout(smallPanel,BoxLayout.Y_AXIS));
smallPanel.setBorder(BorderFactory.createEmptyBorder(3,3,3,3));
nameText = new JTextField();
addressText = new JTextField();
cityText = new JTextField();
smallPanel.add(nameText);
smallPanel.add(addressText);
smallPanel.add(cityText);
panel.add(smallPanel);
contentPane.add(panel, BorderLayout.SOUTH);
    }private class AboutListener implements ActionListener
    {
public void actionPerformed(ActionEvent e)
        {
JOptionPane.showMessageDialog(frame,
                    "GUI Pizza\n\nVersion 1.0\nBuild B20080225-1336\n\n" +
                        "(c) Copyright Merrill Hall 2008\nAll rights reserved\n\n" +
                        "Visit /\nEducation To Go\n" +
                        "Intermediate Java Course",
                    "About GUI Pizza",
                    JOptionPane.INFORMATION_MESSAGE);
        }
    }private class ExitListener implements ActionListener
    {
```

```
public void actionPerformed(ActionEvent e)
        {
System.exit(0);
        }
    }private class NewListener implements ActionListener
    {
public void actionPerformed(ActionEvent e)
        {
// reset all options
regularCrustButton.setSelected(true);
pepperoniBox.setSelected(false);
sausageBox.setSelected(false);
cheeseBox.setSelected(false);
pepperBox.setSelected(false);
onionBox.setSelected(false);
mushroomBox.setSelected(false);
oliveBox.setSelected(false);
anchovyBox.setSelected(false);
breadSticksText.setText("");
buffaloWingsText.setText("");
nameText.setText("");
addressText.setText("");
cityText.setText("");
        }
    }private class SaveListener implements ActionListener
    {
public void actionPerformed(ActionEvent ae)
        {
// write pizza order to an output file
String order = "Pizza Order\n" +
   "===========\n" +
   "Crust:\n";
if (regularCrustButton.isSelected())
order += "     Regular\n";
else if (thinCrustButton.isSelected())
order += "     Thin\n";
else if (deepCrustButton.isSelected())
order += "     Deep-Dish\n";
else if (handCrustButton.isSelected())
order += "     Hand-Tossed\n";
else
JOptionPane.showMessageDialog(frame,
                        "You must select a crust type!",
                        "Crust Type Error",
                        JOptionPane.ERROR_MESSAGE);
order += "Toppings:\n";
if (pepperoniBox.isSelected())
order += "     Pepperoni\n";
if (sausageBox.isSelected())
order += "     Sausage\n";
if (cheeseBox.isSelected())
order += "     Extra Cheese\n";
if (pepperBox.isSelected())
order += "     Peppers\n";
if (onionBox.isSelected())
order += "     Onions\n";
if (mushroomBox.isSelected())
order += "     Mushrooms\n";
if (oliveBox.isSelected())
order += "     Olives\n";
if (anchovyBox.isSelected())
order += "     Anchovies\n";
int bs = 0;
int bw = 0;
try
{
if (!breadSticksText.getText().isEmpty())
bs = Integer.parseInt(breadSticksText.getText());
if (!buffaloWingsText.getText().isEmpty())
bw = Integer.parseInt(buffaloWingsText.getText());}
catch (NumberFormatException nfe)
{
JOptionPane.showMessageDialog(frame,
                        "Side order entries must be numeric,\n and must be whole numbers",
                        "Side Order Error",
                        JOptionPane.ERROR_MESSAGE);
}
if (bs > 0 || bw > 0)
            {
order += "Sides:\n";
if (bs > 0)
order += "     " + bs + " Bread Sticks\n";
if (bw > 0)
order += "     " + bw + " Buffalo Wings\n";
    }
if (nameText.getText().isEmpty() ||
addressText.getText().isEmpty() ||
cityText.getText().isEmpty())
JOptionPane.showMessageDialog(frame,
                        "Address fields may not be empty.",
                        "Address Error",
                        JOptionPane.ERROR_MESSAGE);
else
{
order += "Deliver To:\n";
order += "     " + nameText.getText() + "\n";
order += "     " + addressText.getText() + "\n";
order += "     " + cityText.getText() + "\n";
```

```
}
order += "\n***END OF ORDER ***\n";
try
            {
PrintStream oFile = new PrintStream("PizzaOrder.txt");
oFile.print(order);
oFile.close();
            }
catch(IOException ioe)
            {
System.out.println("\n*** I/O Error ***\n" + ioe);
}
        }
    }
}
```

Print Window Close Window