Programmieren in Java

1. Erläutern Sie die Wirkungsweise der folgenden Layoutmanager des JDK:
BorderLayout
GridLayout
FlowLayout
Null-Layout

2. Nennen Sie vier Steuerelemente die ActionEvents erzeugen.

3. Erläutern Sie an einem Beispiel welche drei Schritte nötig sind, um die korrekte Funktion eines JSlider-Steuerlements zu realisieren.

4. Zeigen Sie den Unterschied zwischen einem MouseListener und einem MouseMotionListener auf.

5. Über folgenden Code wurde eine Benutzereingabe gefordert:
```
String input = JOptionPane.showInputDialog("Bitte Betrag in EUR eingeben:");
```
Der Betrag soll in Dollar umgerechnet (Kurs: 1EUR = 1,31399 Dollar) und auf der Console ausgegeben werden. Mögliche Exceptions sollen angedeutet werden. Berücksichtigen Sie auch den Fall, dass der Benutzer den Dialog mit „Abbrechen“ schließt.

6. Welche Aufgabe erfüllt die paintComponent-Methode im Gegensatz zur repaint-Methode?

7. Warum braucht man eine ButtonGroup für JRadioButtons und nicht für JCheckBox?

8. Was passiert wenn man ein Objekt vom Typ JButton einem JFrame ohne Positionsangabe hinzufügt?

9. Zeigen Sie mit Hilfe eines Code Fragments den sinnvollen Einsatz der Klassen JMenu, JMenutItem und JMenuBar auf.

10. Der Benutzer soll in einem Dialog eine Farbe/Datei auswählen können. Welche Klassen bietet die Bibliothek an?

11. Welche Ausgabe erzeugt folgender Code:
```
class Schokolade {
	public final static int WEISS = 0, BRAUN = 1;
	private int farbe;

	Schokolade(int f) {
		switch (f) {
		case WEISS:
		case BRAUN:
			farbe = f;
			break;
		default:
		throw new IllegalArgumentException("Falsche Schoko-Farbe: " + f);
		}
	}
```

```
        public void test() {
                System.out.println("Aha, du magst also "
                                + ((farbe == WEISS) ? "weisse " : "braune ")
                                + "Schokolade gerne!");
        }

        public static void main(String[] args) {
                // Schokolade ws = new Schokolade( Schokolade.BRAUN );
                Schokolade ws = new Schokolade(4);
                ws.test();
        }
}
```

12. Analysieren Sie folgenden Code:

```
 1  public class BlackPanel extends JPanel {
 2          private static final int RECTANGLE_WIDTH = 20;
 3          private static final int RECTANGLE_HEIGHT = 30;
 4          private int xLeft = 5;
 5          private int yTop = 20;
 6
 7          public BlackPanel() {
 8          }
 9
10          public void paintComponent(Graphics g) {
11                  g.fillRect(xLeft, yTop, RECTANGLE_WIDTH, RECTANGLE_HEIGHT);
12          }
13
14          public void moveRectangleBy(int dx, int dy) {
15                  xLeft = xLeft + dx;
16                  yTop = yTop + dy;
17                  repaint();
18          }
19
20          public static void main(String[] args) {
21                  JFrame frame = new JFrame();
22                  BlackPanel bp = new BlackPanel();
23                  frame.add(bp);
24                  bp.moveRectangleBy(100, 200);
25                  frame.setBounds(0, 0, 300, 300);
26                  frame.setVisible(true);
27                  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28          }
```

a) Was passiert wenn man Zeile 27 weglassen würde?
b) Was passiert, wenn man Zeile 7 und 8 weglassen würde?
c) Zeichnen Sie möglichst genau, wie die Anzeige des Fensters aussieht. Ergänzen Sie, falls nötig, Koordinaten.
d) Wie breit und wie hoch ist bp?
e) Welche Aufgabe erfüllt der Befehl auf Zeile 17?
f) Wie ändert sich die Anzeige wenn man Zeile 24 weglässt?
g) Welche Aufgabe erfüllt das Wort final in Zeile 2?

13. Zeigen sie mit Hilfe eines UML-Diagramms die Vererbungsstruktur der Klassen Manager und Angestellter auf. Zeichnen Sie jeweils ein sinnvolles Attribut und eine sinnvolle Methode ein.