18

Aufgabe 3 (18 Punkte):

a) [10] Ergänzen Sie den folgenden Codeausschnitt aus einer EreignisEmpfänger-Klasse so, dass die Eingabe eines Dateinamens in einer JFextField-Komponente ausgelesen und dann die Methode `public void processFile(String filename)` des `Models` aufgerufen wird.

10

```
class EreignisEmpfänger ..........{
  ... Model model;
  .....
  public void callback-Methode(SomeEvent e){
    .....
    if (...){
      // Verarbeitung der TEXTEINGABE_PER_JTEXTFIELD
      // ab hier ergänzen:
      JTextField text = (JTextField) e.getSource();
      model.processFile(text.getText());
      // Ende der Ergänzungen
    }
    .....
  }
}
```

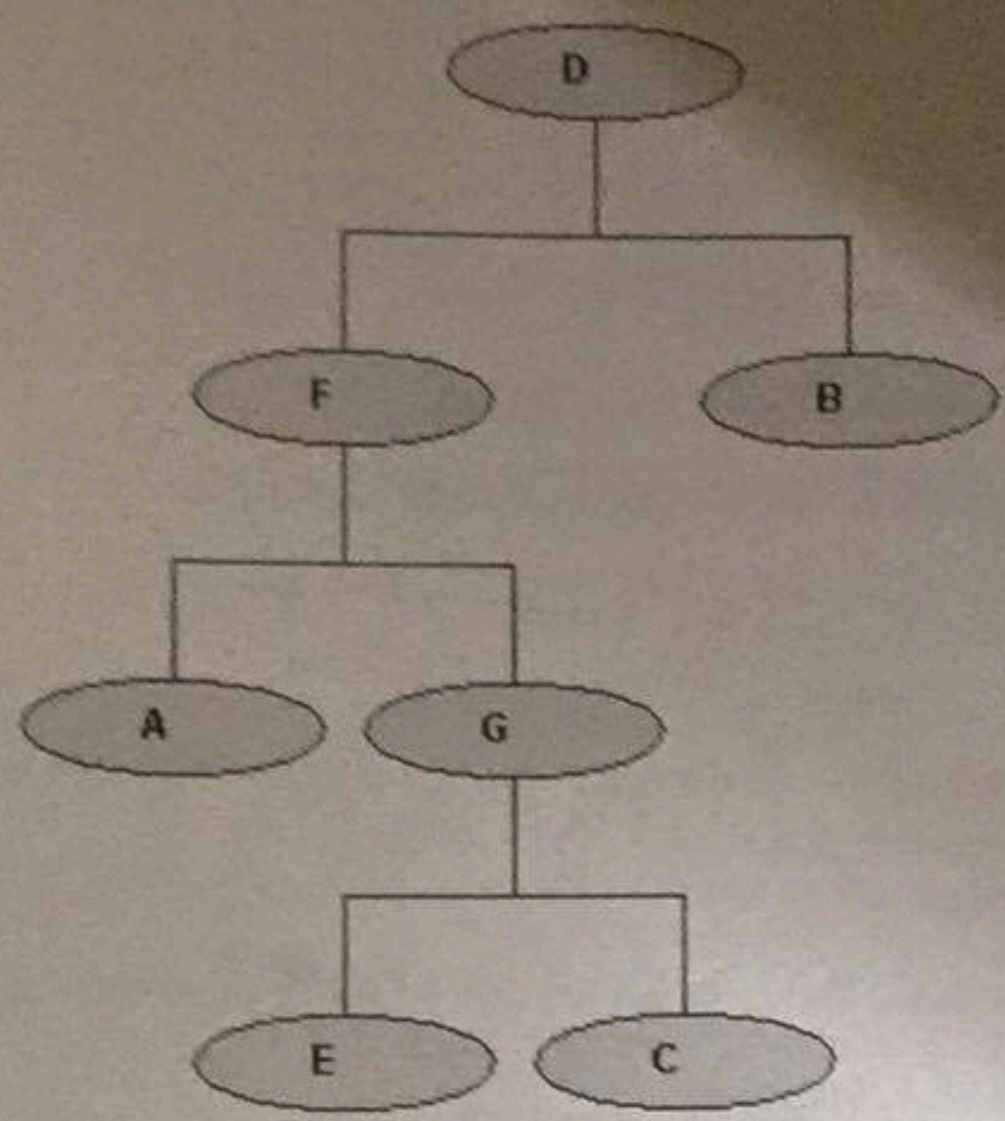
b) [8] Geben Sie für die Klasse Model (nur) die Instanzvariablen und die Implementierung derjenigen Instanzmethoden (aber nicht den Rest der Klasse) an, die für Kopplung von Model und View durch java.beans.PropertyChangeEvents notwendig sind.

8

Hinweis 1: Um Schreibaufwand zu sparen, dürfen Sie in dieser Aufgabe überall die Silbenfolge PropertyChange durch PC abkürzen (damit wird z.B. aus PropertyChangeEvent einfach PCEvent).
Hinweis 2: Es sind hier nur die für die Kopplung von Model und View notwendigen Instanzvariablen und -methoden anzugeben. Der Ablauf des Feuerns von PropertyChangeEvent muss nicht angegeben werden.

```
class Model ..........{
    protected PCSupport pcs;
    public Model(){
        pcs = new PCSupport(this);
    }
    public void addPCListener(PCListener pcl){
        pcs.addPCListener(pcl);
    }
    public void removePCListener(PCListener pcl){
        pcs.removePCListener(pcl);
....}
}
```

b) [6] Gegeben sei der folgende Baum:



b.1) Geben Sie die **inorder**-Reihenfolge der Knoten an:

A, F, E, G, C, D, B

b.2) Geben Sie die **preorder**-Reihenfolge der Knoten an:

D, F, A, G, E, C, B

b.3) Geben Sie die **postorder**-Reihenfolge der Knoten an:

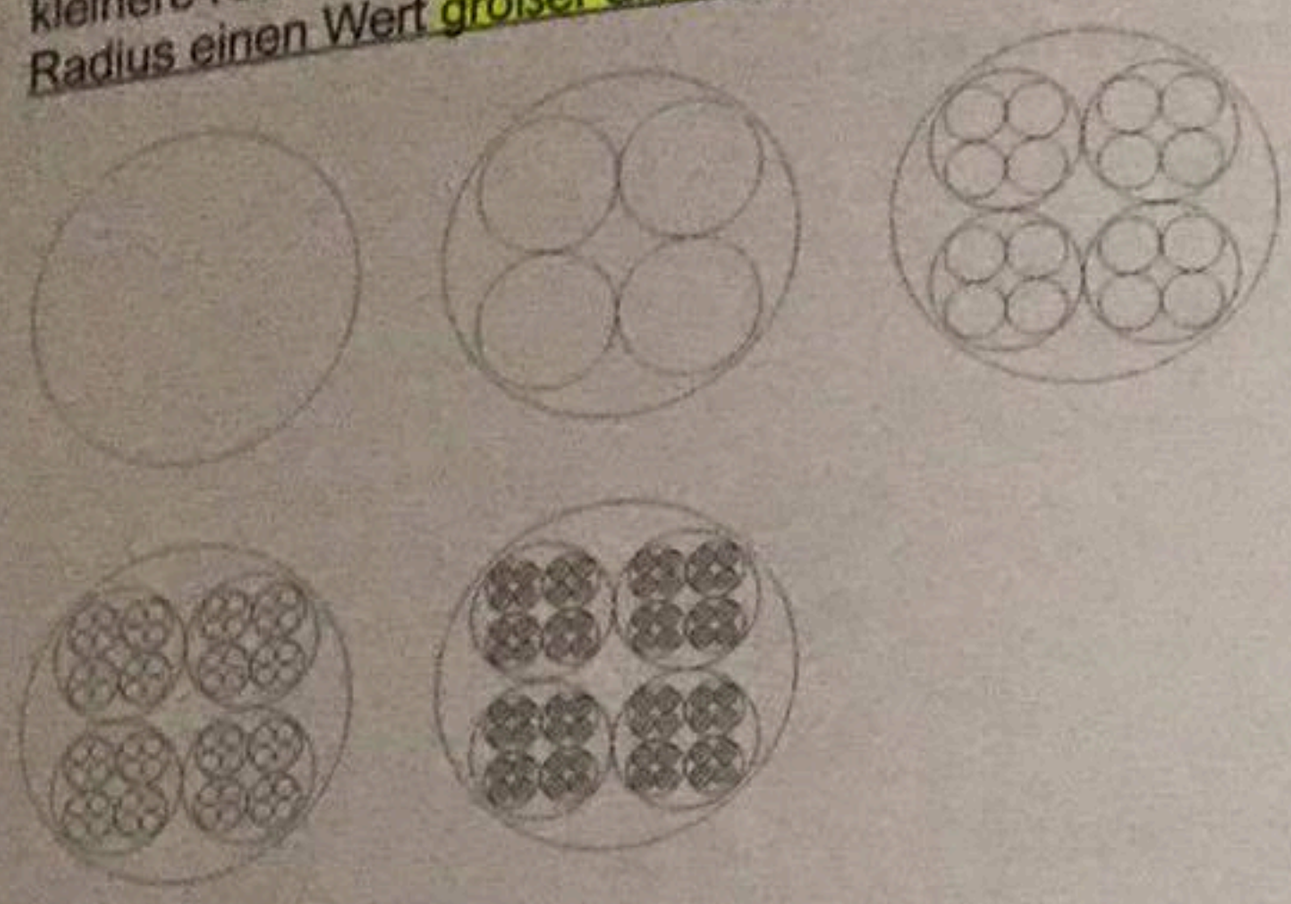
E, C, A, G, F, B, D

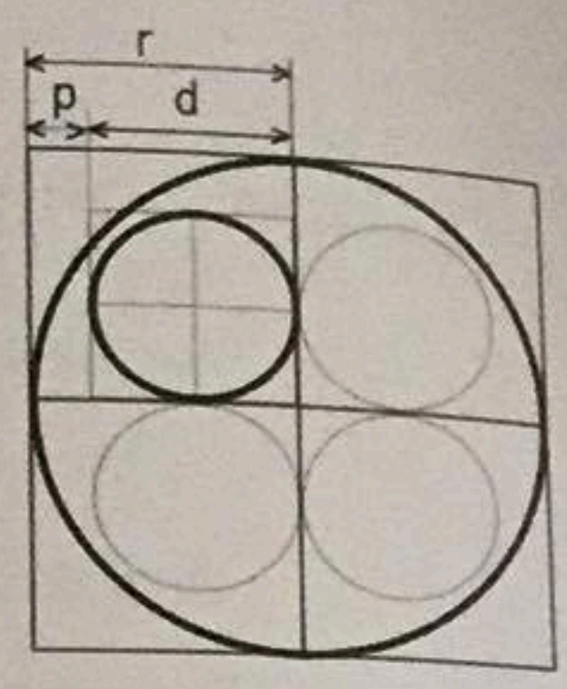
12

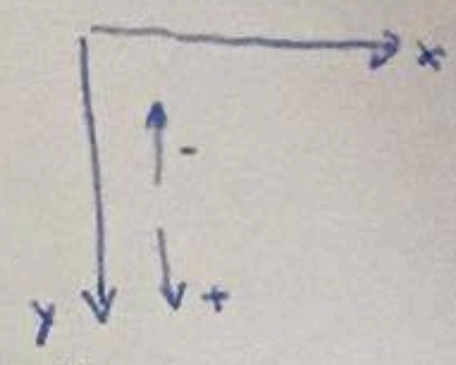
Aufgabe 1 (12 Punkte):

Die folgende Bildreihe veranschaulicht den Aufbau eines **rekursiv** gezeichneten Musters aus Kreisen:

Das Muster besteht aus einem Kreis mit Radius r (1. Bild). Dieser Kreis enthält vier innere Kreise mit den Durchmessern d (2. Bild). Jeder dieser Kreise enthält wiederum 4 kleinere Kreise (3.Bild). Diese Kreise enthalten noch kleinere Kreise usw., solange der Radius einen Wert größer als 4 Pixel hat.







Hinweise:

1) vgl. Bild: Für den Randabstand p gilt: $p = 0{,}18 \cdot r$
Für den Durchmesser d gilt: $d = 0{,}82 \cdot r$

2) Die Reihenfolge, in der Sie die Kreise zeichen, darf von der obigen Beschreibung beliebig abweichen.

Aufgabe (12 Punkte): Ergänzen Sie die folgende Methode. Verwenden Sie **Rekursion**:

*/** Zeichnet das oben beschriebene Muster durch Rekursion. **x** und **y** sind die Koordinaten der linken oberen Ecke des Rechtecks, das das Muster umfasst. **r** ist der Radius des äußersten Kreises des Musters. */*

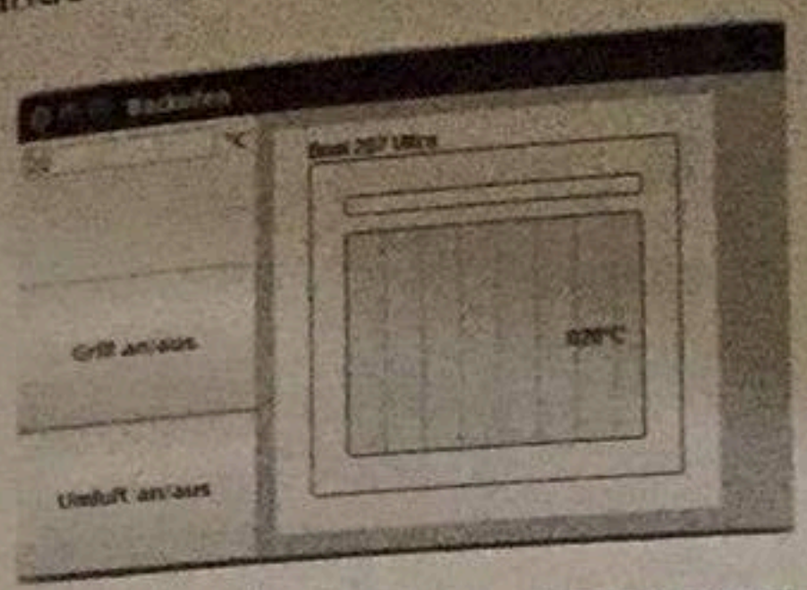
```
public static void drawPattern(Graphics g, int x, int y, int r){
    if(r <= 4){
        return;
    }
    int p = (int)(0.18 * r);
    int d = (int)(0.82 * r);
    g.drawOval(x, y, 2·r, 2·r);
    drawPattern(g, x+p, y+p, d/2);
    drawPattern(g, x+r, y+p, d/2);
    drawPattern(g, x+p, y+r, d/2);
    drawPattern(g, x+r, y+r, d/2);
}
```

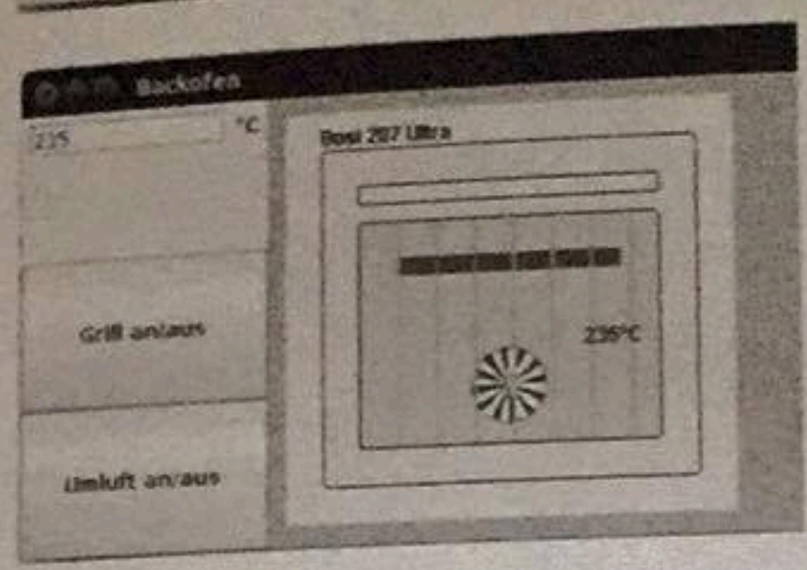
128

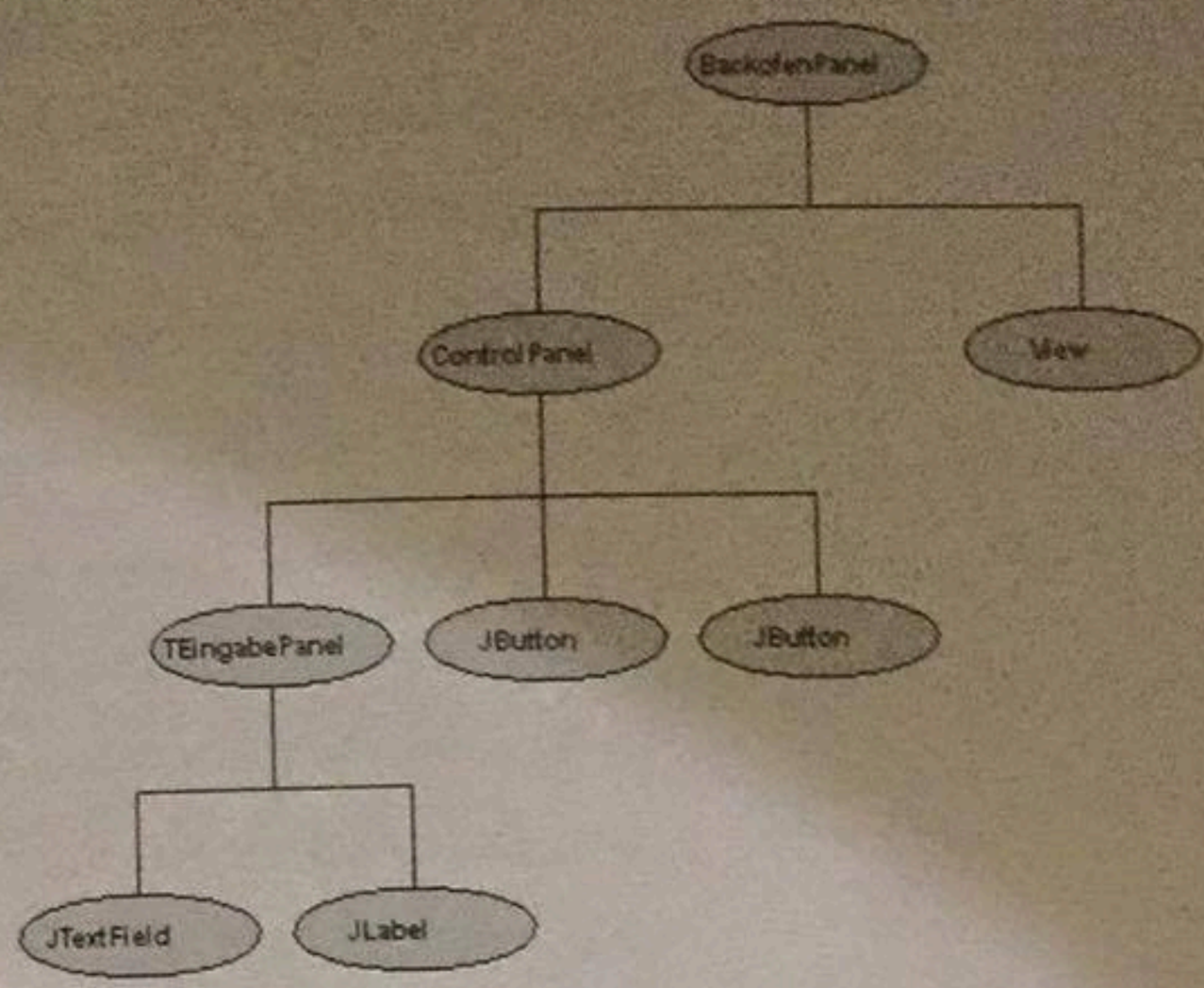
Aufgabe 4 (30Punkte):

Die folgende Abbildung zeigt die grafische Benutzeroberfläche einer Applikation, die entsprechend den im Unterricht besprochen Mustern Model-View-Controller-Architektur und SwingStandardForm programmiert wurde.

Die Applikation zeigt einen Backofen, bei dem die Temperatur eingestellt werden kann. Zusätzlich können unabhängig voneinander ein Grill und ein Umluftventilator an- oder abgeschaltet werden. In den beiden Bildern links sind zwei verschiedene Betriebszustände dargestellt. Im Bild rechts ist ein Teil der Containmenthierarchie dargestellt.







Von der Applikation sei bekannt, dass Model, View, Controller und das BackofenPanel in main() erzeugt werden.
Die GUI ist folgendermassen aufgebaut: Das **BackofenPanel** enthält ein **ControlPanel** und den **View** (die Ansicht des Backofens). Das ControlPanel wiederum enthält ein **TEingabePanel** und die beiden Schaltflächen. Das **TEingabePanel** enthält ein Eingabefeld und eine Beschriftungsfläche für die Temperatureinheit (vgl. Abb. rechts).

Gegeben sind die folgenden Klassen, die entsprechend ihrer Rolle in der MVC-Architektur benannt sind. Alle diese Klassen haben sinnvolle Konstruktoren.

```
class Model        // enthält u.a. die Methoden:
      public void umluftUmschalten()
      public void grillUmschalten()
      public void setT(int temperature)

class View         // die Ansicht des  Backofens

class Controller  // der zentrale Controller

class TEingabePanel // Komponente zur Temperatureingabe

class Befehl {
      public static final String GRILL_AN_AUS   = "GRILL_AN_AUS";
      public static final String UMLUFT_AN_AUS  = "UMLUFT_AN_AUS";
      public static final String SET_T          = "SET_T";
}
```

Beachten Sie bei den folgenden Aufgabenteilen das Einhalten von Model-View-Controller-Architektur und SwingStandardForm!

d) [10] Schreiben Sie für die oben beschriebene Applikation die main()-Methode in der Klasse BackofenApplikation:

8

```
public class BackofenApplikation                                    {

    // Model, View, Controller und das BackofenPanel werden in main() erzeugt
    public static void main (String[] args)                         {
        Model model = new Model();
        View view = new View(model);
        Controller c = new Controller(model);
        model.addPropertyChangeListener(view);
        JFrame frame = new JFrame();
        Container container = frame.getContentPane();
        container.setLayout(new BorderLayout());
        container.add(new BackofenPanel(c, view), BorderLayout.CENTER);
        frame.setVisible(true);
```

für z.B eine Toolbar später

size close

11

}
}

Aufgabe 5 (10 Punkte):

18
4

a) [4] Welche Voraussetzung muss ein Array erfüllen, damit es mit einer binären Suche durchsucht werden kann?

das Array muss sortiert sein!

Welchen Vorteil besitzt die binäre Suche gegenüber der linearen Suche?

Die binäre Suche ist mit einer Laufzeit von O(log n) schneller als die lineare Suche mit der Laufzeit O(n).

a) [4] Ergänzen Sie die Programmzeile

```
public class View extends JPanel implements PropertyChangeListener {
```

4

b) [10] Schreiben Sie für die oben beschriebene Applikation die Klasse

10

```
public class ControlPanel extends JPanel {
    public ControlPanel(ActionListener al){
        this.setLayout(new GridLayout(3,0));
        this.add(new TEingabePanel(al));
        JButton grill = new JButton("Grill an/aus");
        JButton umluft = new JButton("Umluft an/aus");
        grill.setActionCommand(Befehl.GRILL_AN_AUS);
        umluft.setActionCommand(Befehl.UMLUFT_AN_AUS);
        grill.addActionListener(al);
        umluft.addActionListener(al);
        this.add(grill);
        this.add(umluft);
    }
}
```

6

c) [6] Schreiben Sie für die oben beschriebene Applikation die Klasse

```
public class BackofenPanel extends JPanel {
    public BackofenPanel(ActionListener al, View view){
        this.setLayout(new BorderLayout());
        this.add(view, BorderLayout.CENTER);
        this.add(new ControlPanel(al), BorderLayout.WEST);
```

Aufgabe 2 (20 Punkte): 19

a) [6 *vgl. Hinweis 8*] Geben Sie an, ob die folgenden Aussagen richtig oder falsch sind: 6

	richtig	falsch
JFrame hat keine eigene Methode zur Aufnahme einer JToolBar	☒	○
Der Container, der eine JMenuBar-Komponente enthält, muß einen LayoutManager vom Typ MenuLayout benutzen	○	☒
Eine Zeichenkomponente muss die Superklasse JPaintComponent haben	○	☒

b) [2] Welcher LayoutManager sollte für einen Container verwendet werden, der jeder Kindkomponente gleich viel Fläche zuteilt? 2

GridLayout

c) [2] Geben Sie zwei verschiedene Swing-Klassen für atomare Komponenten an: 2

JButton, JLabel, JTextField....

d) [2] Welchen Eventtyp senden JMenuItems beim Anklicken? 2

ActionEvents

e) [2] Wofür steht die Abkürzung I18N? 2

Internationalisierung

f) [2] Welches Layout soll der Container haben, in den ein ToolBar eingefügt wird? 2

BorderLayout

g) [4] Geben Sie möglichst genau an, woran man erkennt, daß die Objekte einer Klasse `AlarmEvents` aussenden können: 3

Folgende zwei Methoden müssen vorhanden sein:

```
public void addAlarmListener(AlarmListener al){
    pcs.addAlarmListener(al);
}
public void removeAlarmListener(AlarmListener al){
    pcs.removeAlarmListener(al);
}
```

PropertyChangeSupport

3