



Vorgeschlagene Bearbeitungszeit: 60 Minuten. In der Klausur wird es möglich sein, eine (zwei für „Neue“) Teilaufgaben zu streichen! Hier sind mehr Übungsaufgaben als Aufgaben in der Arbeit drankommen!

1. Aufgabe

- Erläutere, was objektorientierte Programmierung ist.
- Erläutere, was ein „Konstruktor“ ist und welche Aufgabe er übernimmt, wenn er aufgerufen wird. Gib ein Konstruktor-Beispiel aus einem der beiden angehängten Quellcodes.

2. Aufgabe

- Erläutere, was die Vorteile einer if-else-Bedingung in der Programmierung ist.
- Erläutere, was die Vorteile einer for-Schleife in der Programmierung ist.

3. Aufgabe

- Was macht folgende for-Schleife in Java:

```
for (int i=1; i<10; i=i+2) {System.out.println(„Das ist eine ungerade Zahl: “+i);}
```

- Schreibe eine for-Schleife, die in einem unsortierten Array mit „Int“ als Einträgen das Maximum herausfindet. Pseudocode ist erlaubt!

4. Aufgabe

Entscheide, ob die folgenden Ausdrücke in Java den booleschen Wert „true“ oder „false“ liefern. Stelle dabei deinen Lösungsweg nachvollziehbar dar!

- a) `!(23>4)` b) `!(false) || (false)` c) `((13>31) || (32>31)) && !(false)`

5. Aufgabe

Wofür ist die Direkteingabe bei BlueJ? Erläutere kurz!

6. Aufgabe

Es gibt primitive Typen in der Informatik. Gib ein Beispiel und erläutere, was man darunter versteht.

7. Aufgabe

Im Anhang findest Du die Quelltexte der zwei Klassen **Uhrenanzeige** und **Nummernanzeige**, die beide zum bekannten Projekt Zeitanzeige gehören. Die folgenden Teilaufgaben beziehen sich darauf:

- Die Klasse **Uhrenanzeige** besitzt zwei Konstruktoren. Erläutere deren grundsätzlichen Unterschied.

- b) Gibt es eine Möglichkeit, sich schnell eine Stoppuhr mit Sekunden und Hunderstel zu bauen? Wie müsste man den Konstruktor verändern?
- c) Erläutere an Beispielen aus dem Quelltext die Unterschiede zwischen Methoden mit und ohne (void) Rückgabewert.
- d) In der Methode **taktSignalGeben** taucht die Anweisung „**minute.Erhoehen()**“ auf. Wieso wird diese Schreibweise verwendet und was geschieht hier?
- e) In der Methode **setzeWert** der **Nummernanzeige** taucht die Bedingung „**if((ersatzwert >= 0) && (ersatzwert < limit))**“ auf. Erläutere kurz den Zweck dieser Bedingung.

Quelltext der Klasse Uhrenanzeige

```

/**
 * Die Klassen Uhrenanzeige implementiert die Anzeige einer Digitaluhr.
 *
 * @author Michael Kölling und David J. Barnes
 * @version 2008.03.30
 */
public class Uhrenanzeige
{
    private Nummernanzeige stunden;
    private Nummernanzeige minuten;
    private String zeitanzeige; // simuliert die tatsächliche Anzeige

    /**
     * Konstruktor für ein Exemplar von Uhrenanzeige.
     */
    public Uhrenanzeige()
    {
        stunden = new Nummernanzeige(24);
        minuten = new Nummernanzeige(60);
        anzeigeAktualisieren();
    }

    public Uhrenanzeige(int stunde, int minute)
    {
        stunden = new Nummernanzeige(24);
        minuten = new Nummernanzeige(60);
        setzeUhrzeit(stunde, minute);
    }

    public void taktSignalGeben()
    {
        minuten.erhoehen();
        if(minuten.gibWert() == 0) {
            stunden.erhoehen();
        }
        anzeigeAktualisieren();
    }

    public void setzeUhrzeit(int stunde, int minute)
    {
        stunden.setzeWert(stunde);
        minuten.setzeWert(minute);
        anzeigeAktualisieren();
    }

    public String gibUhrzeit()
    {
        return zeitanzeige;
    }
}

```

```

private void anzeigeAktualisieren()
{
    zeitanzeige = stunden.gibAnzeigewert() + ":"
        + minuten.gibAnzeigewert();
}
}

```

Quelltext der Klasse Nummernanzeige

```

/**
 * Die Klasse Nummernanzeige repräsentiert Darstellungen von digitalen Werten.
 *
 * @author Michael Kölling und David J. Barnes
 * @version 2008.03.30
 */
public class Nummernanzeige
{
    private int limit;
    private int wert;

    /**
     public Nummernanzeige(int anzeigeLimit)
     {
         limit = anzeigeLimit;
         wert = 0;
     }

     public int gibWert()
     {
         return wert;
     }

     public String gibAnzeigewert()
     {
         if(wert < 10) {
             return "0" + wert;
         }
         else {
             return "" + wert;
         }
     }

     public void setzeWert(int ersatzwert)
     {
         if((ersatzwert >= 0) && (ersatzwert < limit)) {
             wert = ersatzwert;
         }
     }

     public void erhoehen()
     {
         wert = (wert + 1) % limit;
     }
}

```