



Achte auf eine übersichtliche und gut erläuterte Darstellung! Du brauchst keine Hilfsmittel!

(Bearbeitungszeit: 60 Minuten)

0000000. Aufgabe**(2 Punkte)**

Bestimme mit ausführlicher Dokumentation den Wahrheitswert (w/f) folgender Aussage:

$!((-4 > -7) \&\& ((3 < 2) \parallel (1 \geq 1)))$

0000001. Aufgabe**(8 Punkte)**

Folgender Programmcode ist gegeben:

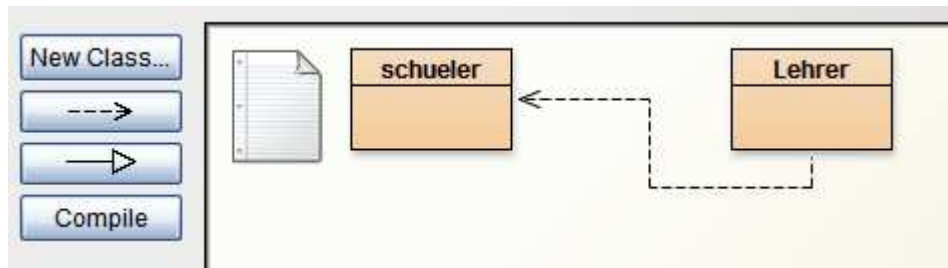
```
import greenfoot.*;
public class Ball extends Actor
{
    private int dx;
    private int dy;
    public Ball()
    {
        dx = 10 - Greenfoot.getRandomNumber(21);
        dy = -5 - Greenfoot.getRandomNumber(6);
    }
    public void act()
    {
        bewege();
    }
    public void bewege()
    {
        setLocation(getX() + dx, getY() + dy);
    }
}
```

- Erläutere anhand des obigen Codes den grundsätzlichen Aufbau eines Java-Programmes.
- Schreibe den Konstruktor so um, dass man beim Erzeugen eines Balles aufgefordert wird, ein Zahlenintervall für die Bewegungsänderungen dx bzw. dy anzugeben (im Moment ist dieser fest -10...10 bzw. -10...0, da getRandomNumber(6) bspw. die Zufallszahlen 0,1,2,3,4,5 erzeugt. Ändere die Definitionen für dx bzw. dy entsprechend ab.
- Ist die Methode getRandomNumber vom Typ „void“?
- Schreibe eine Methode standortX(), die die x-Koordinate des Balles zurückgibt.

0000010. Aufgabe

(5 Punkte)

Bei BlueJ wurden die zwei Klassen Lehrer und schueler erstellt. BlueJ zeigt folgendes Klassendiagramm:



Im Quelltext des Lehrers findet sich dabei folgende Methode:

```
public void noteHolen(schueler name, int fach)
{
    x = name.getNote(fach); System.out.println("Die Note ist " + x + "!");
}
```

während der Code der Klasse schueler dieser ist:

```
public class schueler
{
    private int [] notenliste;
    public schueler(int laenge)
    {
        notenliste = new int[laenge];
        for (int i=0; i<laenge; i++) { notenliste[i]=15; }
    }
    public int getNote(int wo)
    {
        return this.notenliste[wo];
    }
    public void setNote(int welche, int wo)
    {
        notenliste[wo]=welche;
    }
}
```

- Erläutere den gestrichelten Pfeil im Klassendiagramm oben.
- Erläutere, was die Methode Lehrer.noteHolen(...) genau macht.
- Im Schüler-Code wird eine Variable namens „notenliste“ vom Typ „int []“ definiert. Hast du eine Vermutung, wie diese Variable genau aufgebaut ist? Begründe deine Antwort anhand des Konstruktors des Schülers!

0000011. Aufgabe

(5 Punkte)

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 den grundsätzlichen Unterschied.
- Gibt es eine Möglichkeit, sich schnell eine Stoppuhr mit Sekunden und Hundertsteln zu schreiben? Notiere die nötigen Codeänderungen, gerne direkt in den Anhang.
- In der Methode **erhoehen()** der **Nummernanzeige** taucht eine Codezeile zur Variable wert auf: „wert = (wert + 1) % limit ;“. Erläutere kurz, was sie bewirkt.

Quelltext der Klasse Uhrenanzeige

```
public class Uhrenanzeige
{
    private Nummernanzeige stunden;
    private Nummernanzeige minuten;
    private String zeitanzeige;

    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

```
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;
    }
}
```