

KLAUSUR 1 Gruppe A Wahlpflicht 2BK11 8.1.2003 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

Skripte und Aufschriebe beliebiger Art

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Was sind Klassenvariablen und warum benötigt man sie ?

Geben Sie dazu ein Beispiel.

b) Wie heißt die Klassenvariable im folgenden Programm ?

Warum ist das folgende Programm syntaktisch falsch (Compiler meldet Fehler) ?

```
public class TiereTest{
    public static void main(String[] args){
        Gaul g1 = new Gaul();
        Gaul g2 = new Gaul();
        Gaul.setKiloFleischpreis(10);
        g1.setName("Kurt");
    }
}

class Gaul{
    private String name;
    static private double kiloFleischpreis;

    public void setName(String n){
        name = n;
    }

    public void setKiloFleischpreis(double fp){
        kiloFleischpreis = fp;
    }

    public double getKiloFleischpreis(){
        return(kiloFleischpreis);
    }
}
```

2) a) Was versteht man unter dem Begriff "Vererbung" ?
Welchen Vorteil hat man beim Vererben ?

b) Warum ist das folgende Programm syntaktisch falsch (Compiler meldet Fehler) ?

```
public class TiereTest{
    public static void main(String[] args){
        double g;
        Hase h = new Hase();
        h.gewicht = 1200;
        g = h.getGewicht();
    }
}

class Nutztier{
    private String name;
    private double gewicht;

    public void setGewicht(double g){
        gewicht = g;
    }

    public double getGewicht(){
        return(gewicht);
    }
}

class Hase extends Nutztier{
    private double fellpreis;

    public void setFellpreis(double p){
        fellpreis = p;
    }

    public double getFellpreis(){
        return(fellpreis);
    }
}
```

Lösungen:

1)

a) Da es viel Schreibaufwand macht, für jedes Exemplar einer Klasse, deren Attribute immer den gleichen Wert haben (Beispiel Kuh und Methode Literpreis) den gleichen Wert festzusetzen, ist es viel sinnvoller, dies mit einer einzigen Anweisung zu machen. Dazu wird eine sogenannte Klassenvariable definiert. Eine Klassenvariable unterscheidet sich von einem "normalen" Attribut durch den Vorsatz "static".

b)

Der folgende Aufruf

```
Gaul.setKiloFleischpreis(10);
```

setzt voraus, dass die Methode setKiloFleischpreis() static ist.

2)

a) Da es viel Schreibaufwand macht, die gleichen Methoden für die Manipulation eines Attributs (wie z.B. dem Alter und dem Namen) zu implementieren, ist es geschickter dies nur einmal in einer eigenen Klasse zu machen und diese Methoden dann in die Unterklassen zu vererben.

b)

Die folgende Anweisung

```
h.gewicht = 1200;
```

greift von aussen auf den private-Bereich (private Datenmember) einer Klassenvariable zu.

KLAUSUR 1 Gruppe B Wahlpflicht 2BKI1 9.1.2003 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

Skripte und Aufschriebe beliebiger Art

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Was sind Klassenvariablen und warum benötigt man sie ?

Geben Sie dazu ein Beispiel.

b) Wie heißt die Klassenvariable im folgenden Programm ?

Warum ist das folgende Programm syntaktisch falsch (Compiler meldet Fehler) ?

```
public class TiereTest{
    public static void main(String[] args){
        Gaul g1 = new Gaul();
        Gaul g2 = new Gaul();
        Gaul.setKiloFleischpreis(10);
        g1.setName("Kurt");
    }
}

class Gaul{
    private String name;
    static private double kiloFleischpreis;

    static public void setName(String s){
        name = s;
    }

    static public void setKiloFleischpreis(double fp){
        kiloFleischpreis = fp;
    }

    static public double getKiloFleischpreis(){
        return(kiloFleischpreis);
    }
}
```

2) a) Was versteht man unter dem Begriff "Vererbung" ?
Welchen Vorteil hat man beim Vererben ?

b) Welchen Wert gibt das folgende Programm auf dem Bildschirm aus ? Begründen Sie !

```
public class TiereTest{
    public static void main(String[] args){
        Hase h = new Hase();
        h.setGewicht(1200);
        h.setFellpreis(30);
        h.test();
    }
}

class Nutztier{
    private String name;
    private double gewicht;

    public void test(){
        System.out.println(gewicht);
    }

    public void setGewicht(double g){
        gewicht = g;
    }

    public double getGewicht(){
        return(gewicht);
    }
}

class Hase extends Nutztier{
    private double fellpreis;

    public void test(){
        System.out.println(fellpreis);
    }

    public void setFellpreis(double p){
        fellpreis = p;
    }

    public double getFellpreis(){
        return(fellpreis);
    }
}
```

Name, Vorname:

Hilfsmittel:

Skripte und Aufschriebe beliebiger Art

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Was macht das folgende Programm "Prog1" ?

```
public class Hauptprogramm {  
    public static void main(String[] args) {  
        JFrame f = new JFrame();  
        f.setLocation(200,200);  
        f.setSize(400,200);  
        f.setVisible(true);  
    }  
}
```

b) Kann ein Programmierer in dem obigen Programm "Prog1" in dem Fenster f Buttons, Textfelder usw. anbringen ? Begründen Sie !

c) Was muß ein Programmierer machen, damit er Buttons, Textfelder, usw. in einem Fenster anbringen kann, d.h. welche Technik aus der OOP ist dazu unbedingt notwendig ? Begründen Sie (rein verbal) ohne ein Programm !

d) Schreiben Sie ein kleines Programm, in dem ein Button in ein Fenster angebracht wird.

2) a) Was sind Zeichenflächen ? Wie werden diese noch genannt (englischer Ausdruck) ?

b) Kann ein Programmierer in dem obigen Programm "Prog1" in dem Fenster f eine Zeichenfläche anbringen und darin zeichnen ? Begründen Sie (rein verbal) ohne ein Programm !

c) Was muß ein Programmierer machen, damit in dem Fenster Geraden, Ellipsen, usw. gezeichnet werden, d.h., welche Technik aus der OOP ist dazu unbedingt notwendig ? Begründen Sie (rein verbal) ohne ein Programm !

3) a) Ist es sinnvoll, wenn der Java-Code eines Programmierers die Methode `paintComponent(...)` enthält ? Begründen Sie !

b) Wann wird die Methode `paintComponent(Graphics g)` "automatisch" aufgerufen ?

Lösung:

1) a) Ein Fenster (Objekt der Klasse JFrame) wird erzeugt. Die linke, obere Ecke des Fensters beginnt an den Koordinaten (200, 200), ist 400 Pixel breit und 200 Pixel lang.

b) nein, denn f ist ein Objekt der Klasse JFrame. Damit sind die Attribute dieser Klasse fest vorgegeben und damit unveränderbar.

c) Er muß ein Objekt einer Klasse erzeugen, die von JFrame erbt.

Diese von ihm selbst erstellte Klasse kann dann Buttons, usw. als Attribute enthalten.

d)

```
package DemoFenster2;
import java.awt.*;
import javax.swing.*;

public class MyFenster extends JFrame{
    private JButton myb;
    private JPanel myp;
    private Container cont;

    public MyFenster() {
        myb=new JButton("Go");
        myp = new JPanel();
        cont = getContentPane();
        myp.add(myb);
        cont.add(myp);
        setTitle("Meine Zeichnung");
        setLocation(30,60);
        setSize(600,400);
        setVisible(true);
    }
}
```

2) a) An Zeichenflächen, auch Panels genannt, werden Buttons, Textfelder usw. montiert. Diese Zeichenflächen wiederum werden an Fenster montiert.

b) nein, denn f ist ein Objekt der Klasse JFrame. Damit sind die Attribute dieser Klasse fest vorgegeben und damit unveränderbar.

c) Er muß ein Objekt einer Klasse erzeugen, die von JFrame erbt.

Diese von ihm selbst erstellte Klasse kann dann Buttons, usw. als Attribute enthalten.

Er muß ein Objekt einer Klasse erzeugen, die von JPanel erbt.

Diese von ihm selbst erstellte Klasse kann dann Geraden, Ellipsen, usw. in der Methode paintComponent(Graphics g) durch entsprechende Aufrufe (der Geraden, Ellipsen, usw.) zeichnen lassen.

3) a) paintComponent(...) soll niemals direkt innerhalb eines selbst geschriebenen Programms aufgerufen werden.

b) bei der ersten Anzeige des Fensters (setVisible(true)),
nachdem ein Fenster vergrößert (verkleinert) wurde,
nach dem Wegziehen überlagernder Fenster (verdeckte Elemente werden aufgedeckt),
nach einer Größenänderung des Fensters,
nach dem Aufruf des Befehls repaint() im Programm.

KLAUSUR 1 Gruppe B Wahlpflicht 2BKI1 19.1.2006 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

Skripte und Aufschriebe beliebiger Art

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Was sind abstrakte Klassen und warum benötigt man sie ?

Geben Sie dazu ein Beispiel.

b) Geben Sie alle syntaktischen Fehler im folgenden Programm an.

```
public class TiereTest{
    public static void main(String[] args){
        Gaul g1 = new Gaul();
        Gaul g2 = new Gaul();
        Gaul.setKiloFleischpreis(10);
        g1.setName("Kurt");
    }
}

class Gaul{
    private String name;
    private double kiloFleischpreis;

    static public void setName(String s){
        name = s;
    }

    static public void setKiloFleischpreis(double fp){
        kiloFleischpreis = fp;
    }

    static public double getKiloFleischpreis(){
        return(kiloFleischpreis);
    }
}
```


2) a) Was versteht man unter dem Begriff "Vererbung" ?
Welchen Vorteil hat man beim Vererben ?

b) Welchen Wert gibt das folgende Programm auf dem Bildschirm aus ? Begründen Sie !

```
public class TiereTest{
    public static void main(String[] args){
        Hase h = new Hase();
        h.setGewicht(1200);
        h.setFellpreis(30);
        h.test();
    }
}

class Nutztier{
    private String name;
    private double gewicht;

    public void test(){
        System.out.println(gewicht);
    }

    public void setGewicht(double g){
        gewicht = g;
    }

    public double getGewicht(){
        return(gewicht);
    }
}

class Hase extends Nutztier{
    private double fellpreis;

    public void test(){
        System.out.println(fellpreis);
    }

    public void setFellpreis(double p){
        fellpreis = p;
    }

    public double getFellpreis(){
        return(fellpreis);
    }
}
```

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Wie kann man in Java in einer Methode einen input-Parameter realisieren? (6 P)

Geben Sie ein Beispiel.

b) Wie kann man in Java in einer Methode einen output-Parameter realisieren?

Geben Sie ein Beispiel.

c) Welchen Wert hat die Variable `zahl` an der mit `-->` bezeichneten Stelle im Programmausschnitt unten?

```
...
int zahl=13;
verringere(zahl);
// <-- Welchen Wert hat zahl ??
...

public static void verringere (int z){
    z=z-1;
}
...
```

2) (4 P)

Was ist die Signatur einer Methode ?

3) (4 P)

Die Klasse `Schaf` soll schon existieren (mit genau einem zweiparametrischen Konstruktor aus integer Parametern, die das Gewicht und das Alter festlegen).

Erstellen Sie das Feld `"schafstall"` mit 2 Schafen.

Das erste Schaf wiegt 20 Kilo und ist 2 Jahre alt,

das zweite Schaf wiegt 30 Kilo und ist 3 Jahre alt.

4) a) Was sind Klassenvariablen und warum benötigt man sie ?
Geben Sie dazu ein Beispiel.

(6 P)

b) Wie heißt die Klassenvariable im folgenden Programm ?
Warum ist das folgende Programm syntaktisch falsch (Compiler meldet Fehler) ?

```
public class TiereTest{
    public static void main(String[] args){
        Gaul g1 = new Gaul();
        Gaul g2 = new Gaul();
        Gaul.setKiloFleischpreis(10);
        g1.setName("Kurt");
    }
}

class Gaul{
    private String name;
    static private double kiloFleischpreis;

    public void setName(String pname){
        name = pname;
    }

    public void setKiloFleischpreis(double pkiloFleischpreis){
        kiloFleischpreis = pkiloFleischpreis;
    }

    public double getKiloFleischpreis(){
        return(kiloFleischpreis);
    }
}
```

Lösungen:

1)

Durch einen formalen Parameter in einer Methode

Beispiel:

```
public void setName(String pname){  
    name = pname;  
}
```

b) Durch die Rückgabe mit return in einer Methode

Beispiel:

```
public double getKiloFleischpreis(){  
    return(kiloFleischpreis);  
}
```

c)

13

2)

Zwei Methoden haben die gleiche Signatur, wenn:

a) die Namen der Methoden sind gleich UND

b) die Reihenfolge (und die Anzahl) der Parameter (mit zugehörigen Typen) sind gleich

Wichtig:

Der Return-Typ (der Datentyp der durch Return zurückgegeben wird) muss nicht gleich sein.

3)

```
Schaf [] schafstall = {new Schaf (20, 2), new Schaf (30, 3)};
```

4)

a) Da es viel Schreibaufwand macht, für jedes Exemplar einer Klasse, deren Attribute immer den gleichen Wert haben (Beispiel Kuh und Methode Literpreis) den gleichen Wert festzusetzen, ist es viel sinnvoller, dies mit einer einzigen Anweisung zu machen.

Dazu wird eine sogenannte Klassenvariable definiert. Eine Klassenvariable unterscheidet sich von einem "normalen" Attribut durch den Vorsatz "static".

b)

Der folgende Aufruf

```
Gaul.setKiloFleischpreis(10);
```

setzt voraus, dass die Methode setKiloFleischpreis() static ist.

KLAUSUR 1 Wahlpflicht 2BKI1 13.6.2007 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest9{
    public static void main(String[] args){
        int i;
        System.out.println("i="+i);
    }
}

class Kuh {
    private int alter;

    public Kuh(int palter){
        alter = palter;
        System.out.println("alter="+alter);
    }
}
```

2) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest9{
    public static void main(String[] args){
        Kuh myCow=new Kuh();
    }
}

class Kuh {
    private int alter;

    public Kuh(int palter){
        alter = palter;
        System.out.println("alter="+alter);
    }
}
```

3) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest9{
    public static void main(String[] args){
        Kuh myCow=new Kuh();
        myCow.setAlter(10);
    }
}

class Kuh {
    private int alter;

    private void setAlter(int palter){
        alter = palter;
        System.out.println("alter="+alter);
    }
}
```

4) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest9{
    public static void main(String[] args){
        Kuh myCow=new Kuh();
        Kuh.setAll(12);
    }
}

class Kuh {
    private int alter;
    private static int literPreis;

    public void setLiterpreis(int pliterPreis){
        literPreis = pliterPreis;
    }

    static public void setAll(int pliterPreis){
        setLiterpreis(pliterPreis);
    }
}
```

5)

Welche Bildschirmausgabe erzeugt das folgende syntaktisch korrekte Programm?
Begründen Sie!

```
public class MainTest9{
    public static void main(String[] args){
        int alter1=7;
        int alter2=3;
        Kuh myCow=new Kuh();
        myCow.vertausche(alter1, alter2);
        System.out.println(alter1+" "+ alter2+" ");
    }
}

class Kuh {
    private int alter;
    private static int literPreis;

    public void vertausche(int i, int j){
        int temp1, temp2;
        temp1=i;
        temp2=j;
        i=temp2;
        j=temp1;
    }
}
```

6)

Welche Bildschirmausgabe erzeugt das folgende syntaktisch korrekte Programm?
Begründen Sie!

```
public class MainTest9{
    public static void main(String[] args){
        Kuh myCow=new Kuh(13);
        changeKuh(myCow, 7);
        System.out.println(myCow.alter);
    }

    public static void changeKuh(Kuh pk, int palter){
        pk.alter=pk.alter+palter;
    }
}

class Kuh {
    public int alter;

    public Kuh(int palter){
        alter=palter;
    }
}
```

7) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben).

Falls das Programm syntaktisch korrekt ist: Welche Bildschirmausgabe erzeugt es?

```
public class MainTest9{
    public static void main(String[] args){
        Hund mydog=new Hund("Bello", 7);
        System.out.println("ich heiße:"+mydog.name);
    }
}

class Fleischfresser {
    protected String name;

    public Fleischfresser(String pname){
        name=pname;
    }
}

class Hund extends Fleischfresser{
    public int alter;

    public Hund (String name, int palter){
        super(name);
        alter=palter;
        name="Rex";
    }
}
```

8) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programmausschnitt. Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest9{
    public static void main(String[] args){
        Wein kippab = new Wein("Weissherbst", 7);
        System.out.println("Wert="+kippab.bestimmeWert());
        Getraenk kanaller;
        Getraenk trinkab = new Getraenk ("Lassrein");
        kanaller = kippab;
        kippab = kanaller;
    }
}

abstract class Getraenk{
    private String name;

    public Getraenk(String pname){
        name = pname;
    }

    abstract public int bestimmeWert();
}

class Wein extends Getraenk{
    public int alter;

    public Wein (String name, int palter){
        super(name);
        alter=palter;
    }

    public int bestimmeWert(){
        return(alter*10);
    }
}
```

Lösungen

1) Fehlermeldung des Compilers. Die lokale Variable i wurde vor ihrer lesenden Verwendung nicht initialisiert.

2) Fehlermeldung des Compilers. Da schon ein Konstruktor mit einem Parameter existiert, wird kein Standardkonstruktor vom Compiler erzeugt.

3) Fehlermeldung des Compilers. myCow ist ein Objekt der Klasse Kuh und kann deswegen nicht auf die private Methode setAlter() zugreifen.

4) Fehlermeldung des Compilers. setAll() ist static und darf deshalb nur eine static Methode aufrufen. Deshalb muss setLiterpreis() static sein.

5) Ausgabe: 7 3

Java macht bei der Parameterübergabe nur call by value. Der Wert der übergebenen Parameter wird deshalb nicht verändert:

6) Ausgabe: 20

Als Parameter wird eine Referenz auf myCow übergeben. Über diese Referenz wird dann das Attribut des Objekts verändert.

7) Da der formale Parameter name "call by value" ist, kann er nicht verändert werden. Deswegen wird "Bello" ausgegeben.

8) Abstrakte Klasse darf nicht instanziiert werden.

//Getraenk trinkab = new Getraenk ("Lassrein");

"Unterklasse = Oberklasse" nicht möglich

//kipfab = kanaller;

KLAUSUR 1 Wahlpflicht 2BKI1 4.12.2007 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 9 P

a) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

b) Was gibt das Programm auf dem Bildschirm aus, wenn es keine Fehlermeldungen gibt?

```
public class MainTest4 {
    public static void main(String[] args){
        Schwein myS = new Schwein();
        myS.showData();
    }
}

class Schwein{
    private int alter;

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

2) 6 P

a) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

b) Was gibt das Programm auf dem Bildschirm aus, wenn es keine Fehlermeldungen gibt?

```
public class MainTest5 {
    public static void main(String[] args){
        int howOld;
        Schwein myS = new Schwein();
        System.out.println("Alter="+ howOld);
    }
}

class Schwein{
    private int alter;

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

3) 9 P

gegeben sei das folgende Programm:

```
public class MainTest6 {
    public static void main(String[] args){
        Katze myK;           // a) <---
        myK= new Katze ();   // b) <---
    }
}

class Katze {
    private int alter;

    public void showAlter(){
        System.out.println("Alter="+alter);
    }
}
```

a) Was veranlaßt die Deklaration

Katze myK;

genau im obigen Programm im Arbeitsspeicher?

(Auszug aus dem Arbeitsspeicher bringen mit konkreter Adresse)

b) Was veranlaßt danach die Anweisung

myK= new Katze ();

genau im obigen Programm im Arbeitsspeicher?

(Auszug aus dem Arbeitsspeicher bringen mit konkreter Adresse)

4) 6 P

Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest4 {
    public static void main(String[] args){
        Schwein myS = new Schwein();
        myS.showData();
    }
}

class Schwein{
    private int alter;

    public Schwein(int palter){
        alter=palter;
    }

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

5) 20 P

Schreiben (nicht das Beispiel Modell, Gmodell aus dem Unterricht benutzen) Sie ein fehlerfreies Java-Programm, in dem eine **sinnvolle** Vererbung benutzt wird.

Die Unterklasse und die Oberklasse muss jeweils einen (vom Programmierer geschriebenen) Konstruktor besitzen.

Lösungen:

1) a) keine Fehlermeldungen: wenn ein Objekt einer Klasse erzeugt wird, werden alle Attribute auf bestimmte Standardwerte gesetzt; Attribute mit dem Datentyp int werden auf 0 gesetzt.

b) 0

2) a) Fehlermeldung: wenn eine lokale Variable vor einer lesenden Verwendung nicht initialisiert wird, gibt es eine Fehlermeldung beim Kompilieren.

b) nichts, da das Programm nicht kompiliert wurde.

3) a)

Variable	Adresse	Inhalt
...		
myK	0300	?
...		

Im Arbeitsspeicher wird
Platz für die lokale
Variable myK reserviert

b)

Variable	Adresse	Inhalt
...		
myK	0300	0500
	0500	alter = 0

} Im Arbeitsspeicher wird
Platz für das Objekt
reserviert

4) Beim Anlegen eines Objekts der Klasse Schwein wird der Standardkonstruktor benötigt. Da es aber schon einen Konstruktor (mit einem Parameter) gibt, wird der Standardkonstruktor nicht automatisch erzeugt. Es gibt deshalb eine Fehlermeldung.

KLAUSUR 2 Wahlpflicht 2BKI2 29.4.2008 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

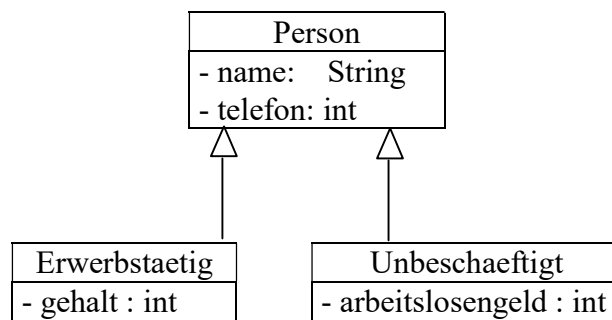
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) Gegeben ist das folgende abgespeckte UML-Diagramm:



a) Erzeugen Sie aus dem folgenden UML-Diagramm die entsprechenden Klassen mit den jeweiligen Konstruktoren, set- und get-Methoden. Entnehmen Sie insbesondere dem UML-Diagramm, ob es sich um eine "normale" oder eine abstrakte Klasse bzw. um ein Interface handelt. Die Konstruktoren müssen jeweils Parameter enthalten. 25P

b) Erzeugen Sie in der Methode main (jeweils durch eine Anweisung) den Erwerbstätigen "Schaffer" mit der Telefonnummer 4711, dem Gehalt von 1500 Euro und den Unbeschäftigten "Schröder" mit der Telefonnummer 4712 und Arbeitslosengeld von 400 Euro. 4P

c) Herr Schaffer bekommt 100 Euro mehr Gehalt. Realisieren Sie dies durch eine Anweisung, die zum alten Gehalt die 100 Euro dazuaddiert. In der Anweisung muss also der alte Gehalt ermittelt werden. 3P

2)

a) Geben Sie ein sinnvolles Beispiel für eine abstrakte Klasse, das nicht im Unterricht benutzt wurde oder in den "Folien" bzw. hier als Aufgabe vorkam. Es reicht eine kurze verbale Beschreibung (keine Implementierung). 2P

b)

4P

Bewerten Sie folgende Aussagen (ohne Begründung) mit wahr oder falsch:

A1)

Wenn eine abstrakte Methode in einer Klasse verwendet wird, dann muss diese Klasse eine abstrakte Klasse sein.

A2)

Wenn eine Klasse abstrakt ist, dann muss mindestens eine sich darin befindlichen Methode abstrakt sein.

c)

2P

Aus der Klasse Person in Aufgabe 1 soll eine abstrakte Klasse gemacht werden.

Was muss in an der Klasse Person geändert werden? (kurze verbale Beschreibung).

d)

10P

In der Klasse Person soll die abstrakte Methode `bestimmeMaximalenKredit()` eingeführt werden.

Was muss im Programm bei Aufgabe 1 geändert werden?

Geben Sie bitte den Quellcode (nur die Änderungen) an.

Überlegen Sie sich eine sinnvolle Implementierung für `bestimmeMaximalenKredit()` !

Lösung:

1)

```
public class Main_E2FI_8_4_08_nr1 {
    public static void main(String[] args){
        Erwerbstaetig e = new Erwerbstaetig("Schaffer", 4711, 1500); // 2P
        Unbeschaeftigt u = new Unbeschaeftigt("Schroeder", 4712, 400); //2P
        e.setGehalt(e.getGehalt()+100); // 3P
    }
}

class Person{ // 11P
    public String name;
    public int telefon;

    public Person (String pName, int pTelefon){
        name = pName;
        telefon = pTelefon;
    }

    public void setTelefon(int pTelefon){
        telefon = pTelefon;
    }

    public void setName(String pName){
        name = pName;
    }

    public int getTelefon(){
        return(telefon);
    }

    public String getName(){
        return(name);
    }
}

class Unbeschaeftigt extends Person{ // 7P
    private int arbeitlosengeld ;

    public Unbeschaeftigt(String pName, int pTelefon,
                           int pArbeitslosengeld){
        super(pName, pTelefon);
        arbeitlosengeld = pArbeitslosengeld;
    }

    public void setArbeitslosengeld(int pArbeitslosengeld){
        arbeitlosengeld = pArbeitslosengeld;
    }

    public int getArbeitslosengeld(){
        return(arbeitslosengeld);
    }
}
```

```

class Erwerbstaetig extends Person{ // 7P
    private int gehalt ;

    public Erwerbstaetig(String pName, int pTelefon, int pGehalt){
        super(pName, pTelefon);
        gehalt = pGehalt;
    }

    public void setGehalt(int pGehalt){
        gehalt = pGehalt;
    }

    public int getGehalt(){
        return(gehalt);
    }
}

```

2)

a) Fahrzeug // 2P

b)

A1) wahr // 2P

A2) falsch // 2P

c)

abstract class Person // 2P

d)

```

abstract class Person{
    public String name;
    public int telefon;

    public abstract int bestimmeMaximalenKredit(); // 2P
    // ...
}

```

```

class Erwerbstaetig extends Person{
    private int gehalt ;
    // ...

    public int bestimmeMaximalenKredit(){ // 4P
        return(2*gehalt);
    }
}

```

```

class Unbeschaeftigt extends Person{
    private int arbeitlosengeld ;
    // ...

    public int bestimmeMaximalenKredit(){ // 4P
        return(3*arbeitlosengeld);
    }
}

```


KLAUSUR 1 Wahlpflicht 2BKI2 16.12.2008 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 14P

- a) Was ist eine Klasse ?
- b) Was ist ein Objekt?
- c) Was enthält eine Klasse?
- d) Was ist der Sinn einer Methode?
- e) Was ist ein Attribut?
- f) Was bedeutet private?
- g) Was bedeutet public?

2) 4P

gegeben sei folgender Programmausschnitt (Hund ist eine Klasse):

```
...  
Hund myh1;  
myh1 = new Hund();  
...
```

Was veranlassen diese zwei Zeilen Programmcode im Arbeitsspeicher?

Bezeichnung Adresse Inhalt

myh1	0800	

3) (Alle Teilaufgaben müssen in **einem** Programm realisiert werden)

32P

- a) Erstellen Sie die Klasse Punkt, (mit den entsprechenden Attributen, Methoden und zwei Konstruktoren), die einen Punkt (Pixel) in einer grafischen zweidimensionalen Oberfläche repräsentieren soll.
- b) Erzeugen Sie einen Punkt mit der x-Koordinate 5 und der y-Koordinate 10.
- c) Verändern Sie mit Hilfe der entsprechenden set- und get-Methode die Koordinaten um den Wert +3 in x- und den Wert -7 in y-Richtung.
- d) Ermitteln Sie mit Hilfe der entsprechenden Methoden die neuen Koordianten des Punktes und geben diese auf dem Bildschirm aus.
- e) Erzeugen Sie einen anderen, neuen Punkt mit der x-Koordinate 6 und der y-Koordinate 11.

Lösung:

1) 14 Punkte

- a) Ein Bauplan, nach dem ein Objekt erstellt wird
- b) Ein nach einem Bauplan im Arbeitsspeicher angelegter, reservierter Speicher.
- c) Attribute und Methoden.
- d) Attribute zu lesen und zu beschreiben.
- e) Daten in einer Klasse.
- f) Auf private Member darf nur innerhalb einer Klasse zugegriffen werden.
- g) Auf public Member darf innerhalb und außerhalb einer Klasse zugegriffen werden.

2) 4 Punkte

Bezeichnung	Adresse	Inhalt
myh1	0800	0900
	...	
	0900	Attribute
		des
		Objekts

3)

```
public class MainKlassen6 {
    public static void main(String[] args) throws Exception{
        Punkt p1, p2;
        // 3 P
        p1= new Punkt(5,10);
        // 3 P
        p1.setX(p1.getX()+3);
        p1.setY(p1.getY()-7);
        // 3 P
        System.out.println("x= "+p1.getX());
        System.out.println("y= "+p1.getY());
        // 3 P
        p2= new Punkt(6,11);
    }
}
```

```
class Punkt{
    private double x; // 1P
    private double y; // 1P

    public Punkt(){ // 3P
        x=0;
        y=0;
    }

    public Punkt(double px, double py){ // 3P
        setPunkt(px,py);
    }

    public void setX (double px){ // 3P
        x=px;
    }

    public void setY (double py){ // 3P
        y=py;
    }

    public double getX(){// 3P
        return(x);
    }

    public double getY(){// 3P
        return(y);
    }
}
```

KLAUSUR 1 Wahlpflicht 2BKI2 Nachtermin 1 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 14 P

a) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

b) Was gibt das Programm auf dem Bildschirm aus, wenn es keine Fehlermeldungen gibt?

```
public class MainTest4 {
    public static void main(String[] args){
        Schwein myS = new Schwein();
        myS.showData();
    }
}

class Schwein{
    private int alter;

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

2) 11 P

a) Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

b) Was gibt das Programm auf dem Bildschirm aus, wenn es keine Fehlermeldungen gibt?

```
public class MainTest5 {
    public static void main(String[] args){
        int howOld;
        Schwein myS = new Schwein();
        System.out.println("Alter="+ howOld);
    }
}

class Schwein{
    private int alter;

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

3) 14 P

gegeben sei das folgende Programm:

```
public class MainTest6 {
    public static void main(String[] args){
        Katze myK;           // a) <---
        myK= new Katze ();   // b) <---
    }
}

class Katze {
    private int alter;

    public void showAlter(){
        System.out.println("Alter="+alter);
    }
}
```

a) Was veranlaßt die Deklaration

Katze myK;

genau im obigen Programm im Arbeitsspeicher?

(Auszug aus dem Arbeitsspeicher bringen mit konkreter Adresse)

b) Was veranlaßt danach die Anweisung

myK= new Katze ();

genau im obigen Programm im Arbeitsspeicher?

(Auszug aus dem Arbeitsspeicher bringen mit konkreter Adresse)

4) 11 P

Welche Fehlermeldung(en) bringt der Compiler im folgenden Programm?

Begründen Sie! (Es muss nicht notwendig eine Fehlermeldung geben)

```
public class MainTest4 {
    public static void main(String[] args){
        Schwein myS = new Schwein();
        myS.showData();
    }
}

class Schwein{
    private int alter;

    public Schwein(int palter){
        alter=palter;
    }

    public void showData(){
        System.out.println("Alter="+alter);
    }
}
```

Lösungen:

1) a) keine Fehlermeldungen: wenn ein Objekt einer Klasse erzeugt wird, werden alle Attribute auf bestimmte Standardwerte gesetzt; Attribute mit dem Datentyp int werden auf 0 gesetzt.

b) 0

2) a) Fehlermeldung: wenn eine lokale Variable vor einer lesenden Verwendung nicht initialisiert wird, gibt es eine Fehlermeldung beim Kompilieren.

b) nichts, da das Programm nicht kompiliert wurde.

3) a)

Variable	Adresse	Inhalt
...		
myK	0300	?
...		

Im Arbeitsspeicher wird
Platz für die lokale
Variable myK reserviert

b)

Variable	Adresse	Inhalt
...		
myK	0300	0500
	0500	alter = 0



Im Arbeitsspeicher wird
Platz für das Objekt
reserviert

4) Beim Anlegen eines Objekts der Klasse Schwein wird der Standardkonstruktor benötigt. Da es aber schon einen Konstruktor (mit einem Parameter) gibt, wird der Standardkonstruktor nicht automatisch erzeugt. Es gibt deshalb eine Fehlermeldung.

Name, Vorname:

Hilfsmittel:

2 Seiten handschriftlich gefertigte Aufschriebe

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Was macht das folgende Programm "Prog1" ?

6P

```
public class Hauptprogramm {  
    public static void main(String[] args) {  
        JFrame f = new JFrame();  
        f.setLocation(200,200);  
        f.setSize(400,200);  
        f.setVisible(true);  
    }  
}
```

b) Kann ein Programmierer in dem obigen Programm "Prog1" in dem Fenster f Buttons, Textfelder usw. anbringen ? Begründen Sie !

5P

c) Was muß ein Programmierer machen, damit er Buttons, Textfelder, usw. in einem Fenster anbringen kann, d.h. welche Technik aus der OOP ist dazu unbedingt notwendig ?

Begründen Sie (rein verbal) ohne ein Pogramm !

6P

d) Schreiben Sie ein kleines Programm, in dem ein Button in ein Fenster angebracht wird. **6P**

2) a) Was sind Zeichenflächen ? Wie werden diese noch genannt (englischer Ausdruck) ? **5P**

b) Kann ein Programmierer in dem obigen Programm "Prog1" in dem Fenster f eine Zeichenfläche anbringen und darin zeichnen ?

5P

Begründen Sie (rein verbal) ohne ein Pogramm !

c) Was muß ein Programmierer machen, damit in dem Fenster Geraden, Ellipsen, usw. gezeichnet werden, d.h., welche Technik aus der OOP ist dazu unbedingt notwendig ?

6P

Begründen Sie (rein verbal) ohne ein Pogramm !

3) a) Ist es sinnvoll, wenn der Java-Code eines Programmierers die Methode paintComponent(...) enthält ? Begründen Sie !

5P

b) Wann wird die Methode paintComponent(Graphics g) "automatisch" aufgerufen ?

6P

Lösung:

1) a) Ein Fenster (Objekt der Klasse JFrame) wird erzeugt. Die linke, obere Ecke des Fensters beginnt an den Koordinaten (200, 200), ist 400 Pixel breit und 200 Pixel lang.

b) nein, denn f ist ein Objekt der Klasse JFrame. Damit sind die Attribute dieser Klasse fest vorgegeben und damit unveränderbar.

c) Er muß ein Objekt einer Klasse erzeugen, die von JFrame erbt.

Diese von ihm selbst erstellte Klasse kann dann Buttons, usw. als Attribute enthalten.

d)

```
package DemoFenster2;
import java.awt.*;
import javax.swing.*;

public class MyFenster extends JFrame{
    private JButton myb;
    private JPanel myp;
    private Container cont;

    public MyFenster() {
        myb=new JButton("Go");
        myp = new JPanel();
        cont = getContentPane();
        myp.add(myb);
        cont.add(myp);
        setTitle("Meine Zeichnung");
        setLocation(30,60);
        setSize(600,400);
        setVisible(true);
    }
}
```

2) a) An Zeichenflächen, auch Panels genannt, werden Buttons, Textfelder usw. montiert. Diese Zeichenflächen wiederum werden an Fenster montiert.

b) nein, denn f ist ein Objekt der Klasse JFrame. Damit sind die Attribute dieser Klasse fest vorgegeben und damit unveränderbar.

c) Er muß ein Objekt einer Klasse erzeugen, die von JFrame erbt.

Diese von ihm selbst erstellte Klasse kann dann Buttons, usw. als Attribute enthalten.

Er muß ein Objekt einer Klasse erzeugen, die von JPanel erbt.

Diese von ihm selbst erstellte Klasse kann dann Geraden, Ellipsen, usw. in der Methode paintComponent(Graphics g) durch entsprechende Aufrufe (der Geraden, Ellipsen, usw.) zeichnen lassen.

3) a) paintComponent(...) soll niemals direkt innerhalb eines selbst geschriebenen Programms aufgerufen werden.

b) bei der ersten Anzeige des Fensters (setVisible(true)),
nachdem ein Fenster vergrößert (verkleinert) wurde,
nach dem Wegziehen überlagernder Fenster (verdeckte Elemente werden aufgedeckt),
nach einer Größenänderung des Fensters,
nach dem Aufruf des Befehls repaint() im Programm.

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

1) a) Was versteht man unter dem Begriff "Vererbung" ? (8P)

Welchen Vorteil hat man beim Vererben ?

b) Warum ist das folgende Programm syntaktisch falsch (Compiler meldet Fehler) ? (8P)

```
public class TiereTest{
    public static void main(String[] args){
        double g;
        Hase h = new Hase();
        h.gewicht = 1200;
        g = h.getGewicht();
    }
}

class Nutztier{
    private String name;
    private double gewicht;

    public void setGewicht(double g){
        gewicht = g;
    }

    public double getGewicht(){
        return(gewicht);
    }
}

class Hase extends Nutztier{
    private double fellpreis;

    public void setFellpreis(double p){
        fellpreis = p;
    }

    public double getFellpreis(){
        return(fellpreis);
    }
}
```

2) a) Wie kann man in Java in einer Methode einen input-Parameter realisieren? (8 P)

Geben Sie ein Beispiel.

b) Wie kann man in Java in einer Methode einen output-Parameter realisieren? (8 P)

Geben Sie ein Beispiel.

c) Welchen Wert hat die Variable zahl an der mit --> bezeichneten Stelle im Programmausschnitt unten?

(8 P)

```
...
int zahl=13;
verringere(zahl);
// <-- Welchen Wert hat zahl ??
...

public static void verringere (int z){
    z=z-1;
}
...
```

3) (10 P)

Die Klasse Schaf soll existieren (mit nur Konstruktor: einem zweiparametrischen Konstruktor aus integer Parametern, die das Gewicht und das Alter festlegen).

Erstellen Sie das Feld "schafstall" mit 2 Schafen.

Das erste Schaf wiegt 20 Kilo und ist 2 Jahre alt,
das zweite Schaf wiegt 30 Kilo und ist 3 Jahre alt.

Lösungen:

1)

a) Da es viel Schreibaufwand macht, für jedes Exemplar einer Klasse, deren Attribute immer den gleichen Wert haben (Beispiel Kuh und Methode Literpreis) den gleichen Wert festzusetzen, ist es viel sinnvoller, dies mit einer einzigen Anweisung zu machen. Dazu wird eine sogenannte Klassenvariable definiert. Eine Klassenvariable unterscheidet sich von einem "normalen" Attribut durch den Vorsatz "static".

b)

Die folgende Anweisung

```
h.gewicht = 1200;
```

greift von aussen auf den private-Bereich (private Datenmember) einer Klassenvariable zu.

2)

Durch einen formalen Parameter in einer Methode

Beispiel:

```
public void setName(String pname){  
    name = pname;  
}
```

b) Durch die Rückgabe mit return in einer Methode

Beispiel:

```
public double getKiloFleischpreis(){  
    return(kiloFleischpreis);  
}
```

c)

13

3)

```
Schaf [] schafstall = {new Schaf (20, 2), new Schaf (30, 3)};
```

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

1) 12P

a) Erzeugen Sie die Klasse Auto mit dem Attribut "kfzZeichen" und den entsprechenden set- und get-Methoden bzw. Konstruktoren.

b) Erzeugen Sie die Klasse Fahrer mit dem Attribut "name" und den entsprechenden set- und get-Methoden bzw. Konstruktoren.

c) Es soll der Fall modelliert werden, dass zu einem Auto (mit KFZ-Zeichen ES BK II) genau einen Fahrer mit dem Namen „Maier“ (und umgekehrt) gehört.

Realisieren Sie dies dadurch, dass in main() ein Auto und ein Fahrer erzeugt werden und diese dann entsprechend verlinkt werden.

Der Einfachheit halber können alle Attribute public gemacht werden.

2) Es wird eine Animation erzeugt, in der ein Wagen immer wieder auf dem Bildschirm von links nach rechts um eine bestimmte Schrittbreite bewegt wird (wenn er am rechten Rand angekommen ist, macht er wieder links weiter). Dazu werden folgende Klassen erzeugt:

In der Klasse SpielJFrame befindet sich ein Timer, dessen zugehörige Methode actionPerformed() immer wieder automatisch aufgerufen wird.

In der Klasse SpielJPanel befindet sich die Methode paintComponent(Graphics g), die auch immer wieder automatisch aufgerufen wird und in der dann der Wagen gezeichnet wird (z.B. mit g.fillRect(ortX, ortY, 30, 25).

In der Klasse Wagen befinden sich die Attribute und Methoden, die den Wagen beschreiben.

a) 13P

Zeichnen Sie ein UML-Klassendiagramm, (mit Attributen und Methoden) das diesen Vorgang modelliert.

1

b) 4P

Was soll in den Methode public void actionPerformed(...) gemacht werden ?

c) 11P

Zeichnen Sie ein UML-Klassendiagramm (nur Attribute angeben, keine Methoden) für den Fall, dass genau zwei Wagen bewegt werden (kein Feld bzw. kein dynamisches Array).

Es soll außerdem noch berücksichtigt werden, dass die Größe der Zeichenfläche durch den Anwender verändert werden kann (der Wagen soll immer bis an den rechten rand fahren).

Lösung:

```
public class MainBKI2_16_3_10_nr1 {
    public static void main(String[] args){
        Fahrer myFahrer = new Fahrer();
        Auto myCar = new Auto();
        myFahrer.setName("Maier");
        myCar.setKfzZeichen("ES BK II");
        // Verlinkung: myCar --> myFahrer
        myCar.f = myFahrer;
        // Verlinkung: myFahrer --> myCar
        myFahrer.a=myCar;
        System.out.println("Name des Autofahrers = "+myCar.f.name);
        System.out.println("KF-Zeichen des Fahrers =
                                "+myFahrer.a.kfzZeichen);
    }
}

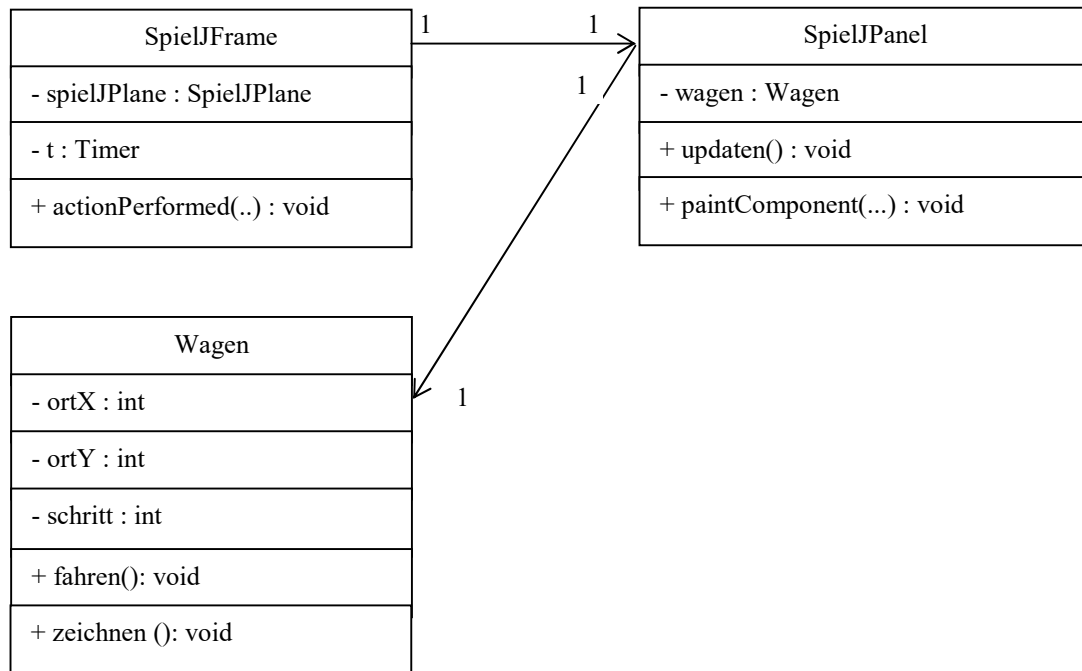
class Auto{
    public String kfzZeichen;
    public Fahrer f;

    public void setKfzZeichen(String pKfzZeichen){
        kfzZeichen=pKfzZeichen;
    }
}

class Fahrer{
    public String name;
    public Auto a;

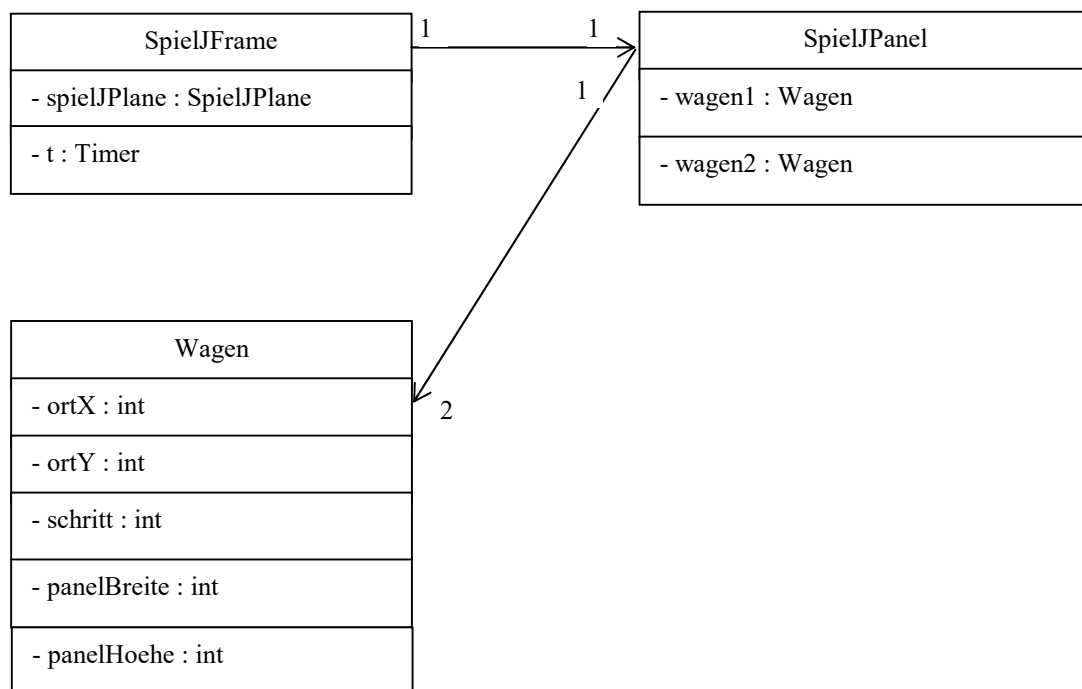
    public void setName(String pName){
        name = pName;
    }
}
```

2a)



b)
updaten und zeichnen

c)



Vielleicht Nachtermin??

1)

d)

14P

Zeichnen Sie ein UML-Klassendiagramm (nur Attribute angeben, keine Methoden) für den Fall, dass mehrere Wagen mit einem Schieberegler bewegt werden (Feld bzw. dynamisches Array). Außerdem sollen Wagen neu angelegt, entfernt und der aktuelle Wagen (z.B. durch Farbgebung) markiert werden können.

2)

a)

Erzeugen Sie die Klasse Auto mit dem Attribut "kfzZeichen" und den entsprechenden Methoden bzw. Konstruktoren.

b)

Erzeugen Sie die Klasse Fahrer mit dem Attribut "name" und den entsprechenden Methoden bzw. Konstruktoren.

c) Es soll der Fall modelliert werden, dass zu einem Auto genau ein Fahrer (und umgekehrt) gehört.

Realisieren Sie dies dadurch, dass in main() ein Auto und ein Fahrer erzeugt werden und diese dann entsprechend verlinkt werden.

Der Einfachheit halber können alle Attribute public gemacht werden.

d)

Es soll der Fall modelliert werden, dass zu einem Auto genau ein Fahrer (und nicht umgekehrt) gehört.

Erzeugen Sie die Klasse Auto mit dem Attribut "kfzZeichen" und den entsprechenden Methoden bzw. Konstruktoren. Im Gegensatz zu a) soll die Verlinkung nicht in main() erstellt werden.

KLAUSUR 2 Wahlpflicht 2BKI2 Nachtermin 1 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Aufgabe 1

Die Klasse Schwanz soll aus Gründen der Einfachheit nur aus den Attributen laenge, einem einparametrigem Konstruktor und den entsprechenden, get- und set-Methoden bestehen.

Die Klasse Hund soll aus Gründen der Einfachheit nur aus den Attributen schwanz, gewicht, einem zweiparametrigem Konstruktor und den entsprechenden get- und set-Methoden bestehen.

1) Implementieren Sie die Klasse Schwanz, die den obigen Anforderungen genügt.

2) Implementieren Sie die Klasse Hund, die den obigen Anforderungen genügt.

3) Betrachten Sie die folgenden Anweisungen in main()

```
public static void main(...){
    Schwanz mys1=new Schwanz(15);
    Hund myh1=new Hund(13, mys1);
    println("L="+myh1.getSchwanz().getLaenge());
    mys1.setLaenge(30);
    println("L="+myh1.getSchwanz().getLaenge());
}
```

Welche Ausgaben werden auf dem Bildschirm erzeugt?

Begründen Sie!

4) Ein Paar Hundezwillinge werden geboren.

Kurz nach der Geburt sind nur ihre Namen unterschiedlich (Rex und Hasso).

Der Rest ist gleich.

a) Simulieren Sie diesen Fall durch ein paar Anweisungen in main()

Ein Jahr nach der Geburt sind auch ihre Schwänze verschieden.

b) Simulieren Sie diesen Fall durch ein paar Anweisungen in main()

Aufgabe 2:

Erstellen Sie die Klasse Wagen und die Klasse Fahrer.

Da viele Fahrer „Autonarren“ sind, soll der Fall modelliert werden, dass zu einem Wagen genau ein Fahrer gehört und umgekehrt. Außerdem muß der Fahrer im Konstruktor von Wagen erzeugt werden.

1) Schreiben Sie ein Java-Programm, das einen Wagen und einen Fahrer erstellt, wobei zu diesem Wagen genau der erzeugte Fahrer gehört (und umgekehrt).

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1a) 2P

Was ist falsch an folgendem Java-Programmausschnitt? (mit Begründung!)

```
...  
int[] v;  
v = new int[3];  
v[3] = -3;  
...
```

b) 2P

Was ist falsch an folgendem Java-Programmausschnitt? (mit Begründung!)

```
...  
double[] w, zahlen;  
zahlen[0] = 12;  
...
```

c) 2P

Ist der folgende Java-Programmausschnitt syntaktisch korrekt?

Wenn ja, welche Werte haben die Elemente der Variablen w?

```
...  
int[] w;  
w = new int[2];  
w[0] = -123;  
...
```

2) 4 P

Die Klasse Schaf soll schon existieren (mit genau einem zweiparametrischen Konstruktor aus integer Parametern, die das Gewicht und das Alter festlegen).

Erstellen Sie das Feld "schafstall" mit 2 Schafen.

Das erste Schaf wiegt 20 Kilo und ist 2 Jahre alt,

das zweite Schaf wiegt 30 Kilo und ist 3 Jahre alt.

3)

a)

18P

Erzeugen Sie die Klasse Kreis mit genau dem Attribut (und nur dem Attribut) "radius", den zu "radius" gehörigen get-und set-Methoden und 2 Konstruktoren.

Erzeugen Sie die Methode berechneFlaeche und berechneUmfang.

b)

22P

Machen Sie in main() hintereinander Folgendes

b1) Erstellen Sie in dem Feld "kreise" 100 Kreise mit den Radien 0, 1, 2, ...99

b2) Geben Sie die Flächeninhalte dieser Kreise auf dem Bildschirm aus.

b3) Verdoppeln Sie die Radien dieser Kreise

b4) Speichern Sie diese Kreise (mit doppeltem Radius) zur Sicherheit in einem neuen Feld mit dem Namen "kreisSicherung".

b5) Angenommen, jemand fügt an das Ende in main()die 2 folgenden Programmierzeilen:

```
kreisSicherung[10].setRadius(13);  
System.out.println("Radius =" + kreise[10].getRadius());
```

Was wird auf dem Bildschirm ausgegeben?

Begründen Sie!

Lösungen:

1a) 2P

`v[3] = -3;` überschreibt nicht reservierten Speicher

1b) 2P

Für das Feld `zahlen` wurde kein Speicher reserviert.

`zahlen` ist eine lokale Variable, deren Wert undefiniert ist.

c) Die Werte des Felds `w` sind nach dem Erstellen mit 0 vorbelegt, `w[1]` wurde verändert, also: 2P

`w[0] = -123`

`w[1] = 0`

2) 4P

`Schaf [] schafstall = {new Schaf (20, 2), new Schaf (30, 3)};`

3) a)

```
class Kreis{
    private double radius;                // 2P
    public static final double pi = 3.14;

    public Kreis(double pRadius){         // 2P
        radius = pRadius;
    }

    public Kreis(){                       // 2P
    }

    public void setRadius(double pRadius){ // 3P
        radius = pRadius;
    }

    public double getRadius(){            // 3P
        return(radius);
    }

    public double berechneUmfang(){       // 3P
        return(2*pi*radius);
    }

    public double berechneFlaeche(){      // 3P
        return(pi*radius*radius);
    }
}
```

```

public class MainBK11_13_11_10_Nr3 {
    public static void main(String[] args) {
        int i;
        Kreis[] kreise; //1P
        kreise = new Kreis[100]; //1P
        for(i=0;i<100;i++){ //4P
            kreise[i]=new Kreis(i);
        }

        for(i=0;i<100;i++){ //4P
            System.out.println("Radius="+kreise[i].berechneFlaeche());
        }

        for(i=0;i<100;i++){ //4P
            kreise[i].setRadius(kreise[i].getRadius()*2);
        }

        Kreis[] kreisSicherung = new Kreis[100]; //2P
        for(i=0;i<100;i++){ //4P
            kreisSicherung[i]=kreise[i];
        }

        // Radius=20 //1P
        // kreisSicherung[10] und kreise [10] zeigen auf das //1P
        // gleiche Objekt.
    }
}

```