

Studiengänge: Elektro- und Informationstechnik, Mechatronik,
Informatik, Flug- und Fahrzeuginformatik, Wirtschaftsinformatik

Prüfung

Grundlagen der Programmierung 2

Objektorientierte Programmierung

Softwareentwicklung 2

Prüfer: Prof. Glavina, Prof. Hahndel, Prof. Schmidt, Prof. Windisch

Zeit: 90 Minuten

Hilfsmittel: keine

Studiengang	Dozent	Matrikel- nummer	Semester	Raum	Platz

(Bitte sorgfältig schreiben!)

Aufgabe	1	2	3	4	5	6	Σ	Note
Punkte								

Schreiben Sie Ihre Antworten direkt in die dafür vorgesehenen Stellen der Seiten dieser Prüfungsangabe. Die Angabe besteht (inkl. Deckblatt) aus 14 (+ 2) Seiten. (**Wichtig:** Die beiden letzten Seiten dürfen Sie zu Ihrer Arbeitserleichterung abtrennen damit Sie bei den Aufgaben 4 und 6 nicht ständig umblättern müssen!)

Bitte geben Sie die ersten 14 Seiten wieder ab, auch wenn einzelne Seiten nicht beschrieben sein sollten. Tragen Sie bitte SOFORT Ihre persönlichen Angaben sowie den Dozenten ein (oben, vor diesem Absatz).

Viel Erfolg!

- d) Erläutern Sie folgenden Methodenkopf (Inhalt und Funktion der Methode sind ohne Belang) und geben Sie auch zu jedem der Parameter an, ob die Übergabe per „call by value“ oder „call by reference“ erfolgt:

```
public static int zaehlePunkte(Student s, Integer mat, double n)
```

A blank sheet of graph paper featuring a uniform grid of small squares. The grid consists of 20 columns and 15 rows, creating a total of 300 square units. The lines are thin and black, set against a white background. There are no margins, text, or other markings on the page.

- e) Beschreiben Sie kurz die Unterschiede zwischen Arrays in Java und C.

A blank sheet of graph paper featuring a uniform grid of small squares. The grid consists of 20 columns and 20 rows, creating a total of 400 square units. The lines are thin and black, set against a white background. There are no margins, text, or other markings on the page.

f) Welche Ausgabe erzeugt das folgende Programm:

```

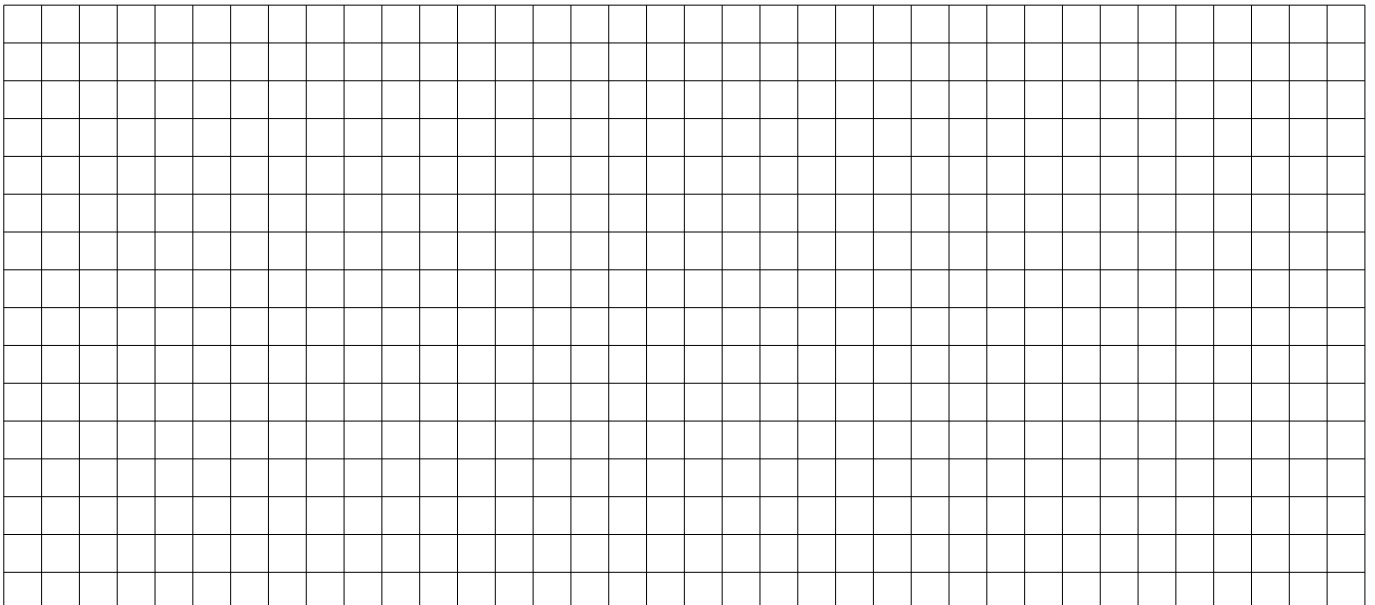
class Abstand {
    private double abstand;

    Abstand(double a){
        abstand=a;
        System.out.println("Abstand_" + a + "_erzeugt!");
    }
    Abstand(double x1, double y1, double x2, double y2){
        this(Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)));
        System.out.println("Abstand_berechnet!");
    }
    void setAbstand(double a) { abstand = a; }
    public String toString() { return "" + abstand + "_"; }
}

class Variablen {
    public static void main(String args[]){
        int x, y = 42;
        String r = "Sommer", s = "Winterreifen", t;
        Abstand d1, d2, d3;

        x = y;
        y = 2*x;
        x++;
        System.out.println(x + ",_" + y);
        t = s;
        s = r + "traum";
        System.out.println(t + ",_" + s);
        d1 = new Abstand(1.0);
        d2 = new Abstand(0,0,2,0);
        d3 = d1;
        d1.setAbstand(2.0);
        System.out.println("" + d1 + d2 + d3 );
        System.out.println((d1==d2) + ",_" + (d1==d3));
    }
}

```



Aufgabe 2: (Klassen, ca. 15%)

a) Gegeben seien folgende Klassendeklarationen und Objektinstanziierungen:

```
class KlasseA { int a; }
class KlasseB extends KlasseA { int b; }
class KlasseC extends KlasseB { int c; }
KlasseA objektA = new KlasseA();
KlasseB objektB = new KlasseB();
KlasseC objektC = new KlasseC();
```

Bewerten Sie die folgenden Codezeilen hinsichtlich ihrer Zulässigkeit und geben Sie eine kurze Erläuterung

```
objektA = objektC;
```

[illegible]

```
objektB.a = objektA.a;
```

[illegible]

```
objektB = objektA ;
```

[illegible]

Aufgabe 4: (Collections Framework, ca 20%)

Gegeben seien die Klasse Spieler

```
class Spieler {
    private String name;
    private int tore = 0;

    Spieler(String n) { name = n; }
    void addTor()      { tore++; }
    int getTore()      { return tore; }
}
```

sowie folgender Ausschnitt der Klasse FussballStat:

```
import java.util.*;

public class FussballStat {
    public static void main(String[] args) {
        private List<Mannschaft> mannschaften =
                                new LinkedList<Mannschaft>();

        Spieler[] spieler1 = { new Spieler("Podolski") , // ...
                                new Spieler("Schweinsteiger") };

        LinkedList<Spieler> spieler2 = new LinkedList<Spieler>();
        spieler2.add(new Spieler("Lucio")); // ...
        ArrayList<Spieler> spieler3 = new ArrayList<Spieler>();
        spieler3.add(new Spieler("Robben")); // ...
        Spieler[] spieler4;

        mannschaften.add(new Mannschaft(spieler1));
        mannschaften.add(new Mannschaft(spieler2));
        mannschaften.add(new Mannschaft(spieler3));
        spieler4 = mannschaften.get(0).getSpieler();
        // ...
    }

    // Weitere Methoden ...
}
```


- a) Ergänzen Sie das folgende Klassenfragment Mannschaft um sämtliche notwendigen Konstruktoren und Getter, damit die oben angegebene main-Methode in Klasse FussballStat lauffähig ist.

```
import java.util.*;
```

```
class Mannschaft {
private Spieler[] spieler;
```

A large grid of graph paper. It consists of 20 columns and 20 rows of small squares. A vertical line is drawn between the first and second columns, creating a margin. A horizontal line is drawn between the first and second rows from the bottom, creating a margin.

- b) Ergänzen Sie die Klasse `FussballStat` um eine Methode mit dem folgenden Methodenkopf

Spieler [] getPlayerMitNToren(int n)

Diese Methode liefert ein Array mit allen Spielern aus `mannschaften`, die `n` Tore geschossen haben.

[illegible]

Aufgabe 6: (GUI-Programmierung , ca. 20%)

Gegeben seien die oben gezeigte Abbildung einer Anwendung zur Verkehrszählung, sowie das dazugehörige Programmfragment.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Verkehrszaehlung extends JFrame implements ActionListener{
    private int pkw = 0, lkW = 0;
    private JLabel anzahlLabel = new JLabel("Anzahl");
    private JPanel centerPanel = new JPanel();
    private JButton lkWButton = new JButton("LKW");
    private JButton pkWButton = new JButton("PKW");
    private JTextField lkWTextfield = new JTextField("0");
    private JTextField pkWTextfield = new JTextField("0");
    private JButton resetButton = new JButton("Reset");

    Verkehrszaehlung(){
        super("VZ");
        // (a) ++++++

        pack();
        setVisible(true);
    }

    public static void main(String[] args) {
        Verkehrszaehlung anwendung = new Verkehrszaehlung();
    }

    public void actionPerformed (ActionEvent e){
        // (b) ++++++

    }
}

```

Aufgabe 4: (Collections Framework, ca 20%)

Gegeben seien die Klasse Spieler

```
class Spieler {
    private String name;
    private int tore = 0;

    Spieler(String n) { name = n; }
    void addTor()      { tore++; }
    int getTore()      { return tore; }
}
```

sowie folgender Ausschnitt der Klasse FussballStat:

```
import java.util.*;

public class FussballStat {
    public static void main(String[] args) {
        private List<Mannschaft> mannschaften =
                                new LinkedList<Mannschaft>();

        Spieler[] spieler1 = { new Spieler("Podolski") , // ...
                                new Spieler("Schweinsteiger") };

        LinkedList<Spieler> spieler2 = new LinkedList<Spieler>();
        spieler2.add(new Spieler("Lucio")); // ...
        ArrayList<Spieler> spieler3 = new ArrayList<Spieler>();
        spieler3.add(new Spieler("Robben")); // ...
        Spieler[] spieler4;

        mannschaften.add(new Mannschaft(spieler1));
        mannschaften.add(new Mannschaft(spieler2));
        mannschaften.add(new Mannschaft(spieler3));
        spieler4 = mannschaften.get(0).getSpieler();
        // ...
    }

    // Weitere Methoden ...
}
```

Aufgabe 6: (GUI-Programmierung , ca. 20%)

Gegeben seien die oben gezeigte Abbildung einer Anwendung zur Verkehrszählung, sowie das dazugehörige Programmfragment.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Verkehrszaehlung extends JFrame implements ActionListener{
    private int pkw = 0, lkW = 0;
    private JLabel anzahlLabel = new JLabel("Anzahl");
    private JPanel centerPanel = new JPanel();
    private JButton lkWButton = new JButton("LKW");
    private JButton pkWButton = new JButton("PKW");
    private JTextField lkWTextfield = new JTextField("0");
    private JTextField pkWTextfield = new JTextField("0");
    private JButton resetButton = new JButton("Reset");

    Verkehrszaehlung(){
        super("VZ");
        // (a) ++++++

        pack();
        setVisible(true);
    }

    public static void main(String[] args) {
        Verkehrszaehlung anwendung = new Verkehrszaehlung();
    }

    public void actionPerformed (ActionEvent e){
        // (b) ++++++

    }
}

```
