

Vorname: [redacted]

Nachname: [redacted]

Matrikelnummer: [redacted]

Dies ist mein letzter Prüfungsversuch ja nein

Aufgabe	1	2	3	4	5	6	7	Gesamt	LV-Note
Punkte max:	5	18	20	12	15	20	10	100	XXXXXX
Punkte	5	18/16	20	11	13	16	10	90/91	1,0

Hinweise:

- 1) Beschriften Sie alle verwendeten Blätter nur einseitig -> die Rückseite ist freizulassen.
- 2) Falls Sie Teile Ihrer Lösung auf Extrablätter schreiben:
Verwenden Sie für jede Aufgabe ein eigenes Extrablatt.
- 3) Auf zusätzlichen Blättern lassen Sie links, rechts, oben und unten mindestens je 2 cm Rand und schreiben Sie in die obere rechte Ecke Ihren Namen und Ihre Matrikelnummer.
- 4) Es sind die in der Lehrveranstaltung eingeführten Fachbegriffe zu verwenden.
- 5) Ausschweifende oder unnötig komplizierte Erklärungen führen zu Punktabzug.
- 6) Die Lösungen sind programmiertechnisch sinnvoll, verständlich und den jeweiligen Konventionen entsprechend auszuführen. Unnötig komplizierte oder unsinnige Programmkonstrukte führen zu Punktabzug.
- 7) Kommunikation jeder Art ist nur mit dem Dozenten gestattet.

Aufgabe 1 (5 Punkte): Halten Sie die oben genannten Formalien (Hinweis 1 bis 3) ein.

5

Aufgabe 2 (18 Punkte):

Geben Sie jeweils eine vollständige Deklaration für die folgenden Methoden an:

18/16

a) [5 Punkte]

4

/**

* Der Rückgabewert gibt an, ob drei aufeinanderfolgende Zahlen a, b, c der Größe nach ansteigend angeordnet sind. Bsp.:

* b = isOrderedAscending(-1, 3, 8); // b wird true zugewiesen

* m = isOrderedAscending(1, 1, 2); // m wird false zugewiesen

* k = isOrderedAscending(-1, 2, 0); // k wird false zugewiesen

*/

public static boolean isOrderedAscending(int a, int b, int c)

b) [6] Geben Sie eine Deklaration der folgenden Methode an:

6

/**

erzeugt mit nur zwei Ausgabeanweisungen, von denen jede maximal einen Zeilenvorschub ausgibt, die folgende Ausgabe:

123456789012345

11111

1111

111

11

1

Hinweis: Die Zahlenfolge 11111, 1111, .. 1 folgt dem Schema zahl[j] = (zahl[j-1]-1)/10

*/

public static void printTriangle(PrintWriter out)

zahl[0] = 11111
zahl[1] = 1111/10

c) [7 Punkte]

5

/**

* Gibt die Anzahl der Indices zurück, für die Array a und Array b den gleichen Wert

* enthalten. Bsp.:

* equalCount (new int[] {1, 2, 3, 4}, new int[] {1, 1, 1, 4}) gibt 2 zurück

* equalCount (new int[] {1, 2, 3, 4}, new int[] {0, 1, 2, 3}) gibt 0 zurück

Beachte: Die Methode soll für alle möglichen Parameterwerte korrekt funktionieren! */

```
public static int equalCount (int [] a, int [] b)
```

Aufgabe 3 (20 Punkte):

20

Gegeben sei:

```
public abstract class BasicPictureData {
    protected int width;
    protected int height;
    public BasicPictureData (int width, int height) {
        this.width = width;
        this.height = height;
    }
    public double getPixelCount () {
        return width*height;
    }
    protected String sizeToString () {
        return width + " X " + height;
    }
    protected abstract String getPictureType ();
    public String toString () {
        return getPictureType() + ": " + sizeToString ();
    }
}
```

```
public class Dimension {
    public int width;
    public int height;
    public Dimension (int width, int height) {
        this.width = width;
        this.height = height;
    }
}
```

Geben Sie die Deklaration einer Klasse PictureData an, die von BasicPictureData erbt und die nicht abstrakt ist. Die Klasse PictureData besitzt eine Instanzvariable vom Typ int, in der die farbtiefe (angegeben in der Einheit Bit) gespeichert wird. ✓

Die Klasse PictureData soll einen vollständigen Konstruktor, einen Kopierkonstruktor und einen weiteren Konstruktor mit der Parameterliste (Dimension d, int farbtiefe) besitzen. Die Ausgabe eines PictureDataobjekts in der Art `System.out.println (pictureData);` soll eine Ausgabe nach dem Muster Farbbild: 800 X 540, 18 bit erzeugen. Hat die Farbtiefe den Wert 1, dann soll in der Ausgabe die Zeichenkette "SW-Bild" die Zeichenkette "Farbbild" ersetzen und die Ausgabe soll nach dem Muster SW-Bild: 800 X 540 erfolgen. ✓

Vermeiden Sie unnötige Codeervielfachungen.

Aufgabe 4 (12 Punkte):

11

Vereinfachen Sie die Codefragmente a) bis c) so weit wie möglich, aber ohne die Funktionalität zu verändern:

a) [3] `if (markerflag == false) {` 3
 `markerflag = true;`
`} else {`
 `markerflag = false;`
`}`

b) [5] // Beachte: term kann auch einen negativen Wert haben! 5
`public static int newCount (int count, int term) {`
 `for (int i = 0; i < term; i++) {`
 `count = count + 1;`
 `}`
 `return count;`
`}`

c) [4] `private void runSubMenu (char zeichen) {` 3
 `switch (zeichen) {`
 `case 'a': runSubMenuA ();`
 `break;`
 `case 'A': runSubMenuA ();`
 `break;`
 `case 'b': runSubMenuB ();`
 `break;`
 `case 'B': runSubMenuB ();`
 `break;`
 `}`
`}`

Aufgabe 5 (15 Punkte):

13

a) [6] Nennen Sie die vier grundlegenden Streamklassen des IO-Systems von Java:

java.io.InputStream java.io.Reader
java.io.OutputStream java.io.Writer

6

b) [5] Ordnen Sie den folgenden mit printf() verwendbaren Umwandlungszeichen die richtige Bedeutung (durch Ankreuzen) zu:

- d depressed double direct decimal
- f fixed formal fine ~~float~~
- s sign supreme string severe

c) [4] Die Variable x enthalte den Wert 98765,432. Gibt es einen Formatierungsstring format, der einen der Formatspezifizierer s, d, f oder e verwendet und dafür sorgt, dass die Ausgabeanweisung System.out.printf(format, x); den in x enthaltenen Wert in der Form 98.7e+03 ausgibt?

3

ja, format hat folgenden Wert: _____

4

nein, das geht nicht, weil der Formatspezifizierer „e“ (Exponentialdarstellung) den Wert mit genau einer Vorkommate darstellt.

✓

Aufgabe 6 (20 Punkte):

Geben Sie ohne Begründung an, welche der folgenden Aussagen zutreffen und welche nicht zutreffen:

16

	trifft zu / trifft nicht zu	
Ein Objekt kann mehrere Typen haben	<input checked="" type="radio"/>	<input type="radio"/>
Eine abstrakte Klasse muss mit new instanziiert werden	<input type="radio"/>	<input checked="" type="radio"/>
Eine abstrakte Klasse hat immer mindestens eine abstrakte Methode	<input checked="" type="radio"/>	<input type="radio"/>
Exceptions sind Interfaces	<input type="radio"/>	<input checked="" type="radio"/>
Arrays sind Referenztypen	<input checked="" type="radio"/>	<input type="radio"/>
Ein Package kann mehrere Interfaces implementieren	<input type="radio"/>	<input checked="" type="radio"/>
Eine lokale Variable darf nicht public sein	<input checked="" type="radio"/>	<input type="radio"/>
Der Typ von allen Variablen ist zur Compilezeit bekannt	<input checked="" type="radio"/>	<input type="radio"/>
Es kann sein, dass eine Variable eine Referenz auf ein Objekt speichert	<input checked="" type="radio"/>	<input type="radio"/>
Ein Kopierkonstruktor hat immer den Rückgabetyt void	<input type="radio"/>	<input checked="" type="radio"/>

Aufgabe 7 (10 Punkte):

Betrachten Sie das folgende Java-Programm:

```
public class Exa extends Exception {
}
public class Exb extends Exa {
}
public class Exc extends Exb {
}
//-----
public class Aufgabe07 {
    public static void main(String[] sonja) {
        for (int n = 1; n <= 3; n++) {
            try {
                met01(n);
            } catch (Exa exa) {
                System.out.println("R ex:A");
            } finally {
                System.out.println("S ----");
            }
        }
        // -----
        private static void met01(int n) throws Exa {
            try {
                met02(n);
                System.out.println("T ----");
            } catch (Exc exc) {
                System.out.println("U ex:C");
            } catch (Exb exb) {
                System.out.println("V ex:B");
                throw exb;
            }
        }
        // -----
        protected static void met02(int n) throws Exa {
            try {
                switch (n) {
                    case 1:
                        throw new Exa();
                    case 2:
                        throw new Exb();
                    case 3:
                        throw new Exc();
                }
            } catch (Exc exc) {
                System.out.println("W ex:C");
                throw exc;
            }
        }
    }
}
//-----
```

Geben Sie an, was dieses Programm zur Standardausgabe (d.h. zum Bildschirm) ausgibt.

R ex: A
S ----
V ex: B
R ex: A
S ----
W ex: C
U ex: C
S ----

10

✓

Aufgabe 2

c)

```
public static int equalCount(int[] a, int[] b){  
    for (int i=0; i < a.length ↑ ↗  
    int laenge = a.length; ev. null  
    if (b.length < a.length){  
        laenge = b.length;  
    }  
    int zaehler = 0;  
    for (int i=0; i < laenge; i++){  
        if (a[i] == b[i]){  
            zaehler++;  
        }  
    }  
    return zaehler;  
}
```

5

Aufgabe 3

```
public class PictureData extends BasicPictureData {
    protected int farbtiefe;
    public PictureData(int width, int height, int farbtiefe) {
        super(width, height);
        this.farbtiefe = farbtiefe;
    }
    public PictureData(PictureData p) {
        this(p.width, p.height, p.farbtiefe);
    }
    public PictureData(Dimension d, int farbtiefe) {
        this(d.width, d.height, farbtiefe);
    }
    protected String getPictureType() {
        if (farbtiefe == 1) {
            return "SW-Bild";
        }
        return "Farbbild";
    }
    public String toString() {
        if (getPictureType() farbtiefe > 1) {
            return super.toString() + ", " + farbtiefe + " bit";
        }
        return super.toString();
    }
}
```

✓

20

Aufgabe 4

a)

```
markerflag = !markerflag;
```

✓ 3

b)

```
public static int newCount (int count, int term){  
    if (term > 0){  
        return count + term;  
    }  
    return count;  
}
```

3

✓ 5

c)

```
private void runSubMenu (char zeichen){  
    zeichen = Character  
    switch (zeichen.toLowerCase()) {  
        case 'a': runSubMenuA ();  
            break;  
        case 'b': runSubMenuB ();  
            break;  
    }  
}
```

3

✓ 3