

WS2015/16 Klausur Programmierung 1 (Konzepte)

Montag, 08.02.2016

Es sind keine Hilfsmittel außer dem Schreibgerät zugelassen.

Es sind keine Unterlagen zugelassen.

Es ist ausschliesslich das vom Dozenten ausgeteilte Papier zugelassen.

Bearbeitungszeit: 80 min

Bitte Vollständig ausfüllen:

Vorname:

Nachname:

Matrikelnummer:

Letzter Prüfungs

Leistungsbewertung:

Maximal sind 100 Punkte erreichbar.

Zum Bestehen sind 47 Punkte notwendig.

Aufgabe	1	2	3	4	5	6	7	Gesamt	LV-Note
Punkte max:	9	11	10	10	14	28	18	100	XXXXXX
Punkte	9	10	10	10	10	14	15	78	1,3

Weitere Hinweise:

- 1) Kommunikation jeder Art ist nur mit dem Dozenten gestattet.
- 2) Lassen Sie auf allen Blätter links, rechts, oben und unten mindestens je 2 cm Rand
- 3) Richtige Groß-/ Kleinschreibung ist zu beachten
- 4) Die Lösungen sind programmiertechnisch sinnvoll, verständlich und den jeweiligen Konventionen entsprechend auszuführen. Unnötig komplizierte oder unsinnige Programmkonstrukte führen zu Punktabzug.
- 5) Es sind die in der Lehrveranstaltung eingeführten Fachbegriffe zu verwenden.
- 6) Ausschweifende oder unnötig komplizierte Erklärungen führen zu Punktabzug.
- 7) Bei Aufgaben mit Ankreuzen einer von zwei Antwortmöglichkeiten führt jede falsche Antworten zur Nicht-Wertung einer richtigen Antwort. Bei Unsicherheit sollten Sie einzelne Fragen ggf. nicht beantworten.
- 8) Falls Sie Teile Ihrer Lösung auf Extrablätter schreiben:
 - * Verwenden Sie für jede Aufgabe ein eigenes Extrablatt.
 - * Beschriften Sie alle Extrablätter nur einseitig -> die Rückseite ist freizulassen.
 - * Auf allen Extrablätter lassen Sie links, rechts, oben und unten mindestens je 2 cm Rand und tragen Sie oben Ihren Namen und Ihre Matrikelnummer ein.

Aufgabe1 (9 Punkte):

Schreiben Sie ein einfaches, aber vollständiges Java-Programm namens winter im Package zwei015, das den Text Gar nicht kalt dieser "Winter" ausgibt.

```
package zwei015;

public class winter {
    public static void main (String[] args) {
        System.out.println(Gar nicht kalt dieser Winter
        "Gar nicht kalt dieser 'Winter'");
    }
}
```

Aufgabe 2 (11 Punkte):

a) [7] Vervollständigen Sie die Methode:

// Gibt true zurück, falls die Elemente im Array aufsteigend sortiert sind. Sind die Elemente // nicht aufsteigend sortiert, dann wird false zurückgegeben.

```
public static boolean isSortedAscending(double[] a){
```

```
    for(int i=0; i < a.length; i++) {
        if(a[i] <= a[i+1]) {
            continue;
        } else {
            return false;
        }
    }
    return true;
}
```

b) [4] Vervollständigen Sie die Methode

```
public static File[] toFileArray(String[] filenames) {
```

```
    File[] fa = new File(filenames.length);
    for(int i=0; i < filenames.length; i++) {
        fa[i] = new File(filenames[i]);
    }

    return fa;
}
```

Aufgabe 3 (10 Punkte): Geben Sie eine Implementierung an für:

10
3

3.a) [3 Punkte]

*/** Gibt nur dann true zurück, falls text das Zeichen in theCharacter enthält. */*
`public static boolean containsCharacter(String text, char theCharacter) {`

```
for(int i=0; i<#text.length(); i++) {  
    if (text.charAt(i) == theCharacter) {  
        return true;  
    }  
}
```

```
return false;
```

```
}
```

3.b) [7 Punkte] Als Bestandteil einer Berechnung und der Ausgabe der Ergebnisse in einer Tabelle, die z.B. so aussehen kann:

x	y(x)	z(x)
0.100	-1859.00000	0.0099833
0.200	-1464.00000	0.0397339
0.300	-1081.10526	0.0886561

~~7~~
7

existieren u.a. die Methoden

```
public static double y(double x)  
public static double z(double x)
```

Die Ausgabe der ^STabellenkörpers erfolgt mithilfe der Methode `printNumberLine()` zur formatierten Ausgabe einer Zahlenzeile. Geben Sie die Implementierung dieser Methode an:

*/** Ausgabe einer Zeile mit den Werten von x, y(x) und z(x) in drei Spalten, die alle die Spaltenbreite b haben. Anzahl der Nachkommastellen wie im Beispiel */*

3) `public static void printNumberLine(double x, int b) {`

```
System.out.printf("%" + b + ".3f%" + b + ".5f%" + b + ".7f",  
x, y(x), z(x));
```

(%suf)

```
}
```

10

Aufgabe 4 (10 Punkte):

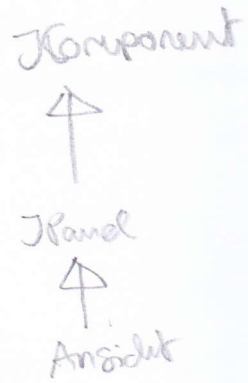
Hinweis zur Bewertung: Punktzahl = (richtige Antworten – falsche Antworten) * 1 P

6

a) [6] Es sei JPanel die direkte Superklasse von Ansicht und es sei JComponent die direkte Superklasse von JPanel. Ausserdem haben alle Klassen einen Standardkonstruktor.

Welche der folgenden Codezeilen ist korrekt?

	korrekt	fehlerhaft
Ansicht a = new Ansicht();	<input checked="" type="checkbox"/>	<input type="checkbox"/>
JPanel p = new Ansicht();	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ansicht b = new JPanel();	<input type="checkbox"/>	<input checked="" type="checkbox"/>
JComponent c = a;	<input checked="" type="checkbox"/>	<input type="checkbox"/>
a = new JComponent();	<input type="checkbox"/>	<input checked="" type="checkbox"/>
JComponent d = c;	<input checked="" type="checkbox"/>	<input type="checkbox"/>



4

b) [4] Ordnen Sie den folgenden mit printf() verwendbaren Umwandlungszeichen die richtige Bedeutung (durch Ankreuzen) zu:

- d double decimal direct dual
- f fixed final fine float
- s sausage subclass string simple
- e elegant extra size exponential exact

Aufgabe 5 (14 Punkte):

Geben Sie für die im folgenden beschriebenen Methoden jeweils eine Implementierung an:

10

a) [5] Von der Klasse `NamedPoint` sei bekannt, dass sie als einzigen Konstruktor `public NamedPoint(double x, double y, String bezeichner)` hat.

*/** Liest aus der Eingabe nacheinander die zwei Gleitkommazahlen x und y und die Zeichenkette bezeichner und erzeugt daraus ein NamedPoint-Objekt, das zurückgegeben wird. Sie können ohne Prüfung davon ausgehen, dass der Inhalt der Eingabe korrekt ist und der in der Eingabe enthaltene Bezeichner keinen Leerraum enthält. */*

5

```
public static NamedPoint createNamedPoint(Scanner in) {  
    return new NamedPoint(in.nextDouble(), in.nextDouble(), in.next());  
}
```

b) [9] Die Methode `drawSquares(Graphics g)` soll mithilfe des `Graphics`-Objekts mehrere **schwarze** Quadrate zeichnen. Ein von `drawSquares(..)` gezeichneter Bildschirmbereich sieht mit zusätzlichem 50-Pixel-Hilfsraster und Koordinatenachsen so aus:

5

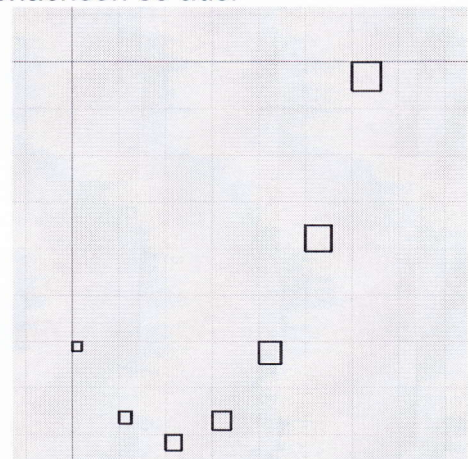
Für die Koordinaten (x_q, y_q) der linken oberen Ecke der Rechtecke der Quadrate gilt:

x_q hat für die aufeinanderfolgenden Quadrate die Werte 0, 50, 100, 150, etc. im Bereich $0 \leq x_q \leq 300$

$$y_q = 400 - 1/100 * (x_q - 100)^2$$

Die Seitenlänge l_q eines Quadrats hängt von seiner Position x_q nach der Formel

$$l_q(x_q) = 10 + 0,0725 * x_q$$



```
public void drawSquares(Graphics g) {
```

```
    int yq = 0;
```

```
    int lq = 0;
```

```
    for (int xq = 0; xq <= 300; xq += 50) {  
        yq = (int) (400 - 0,001 * ((xq - 100) * (xq - 100)));
```

```
        lq = 10 (int) (10 + 0,0725 * xq);
```

```
        g.drawRect(xq, yq, lq, lq);  
    }  
}
```

3

}

14 Aufgabe 6 (28 Punkte)

Gegeben sei:

```
public class Joghurt {
    protected String marke;
    protected int gewicht;

    public Joghurt(String marke, int gewicht) {
        this.marke = marke;
        this.gewicht = gewicht;
    }

    public String getProductCategorie(){
        return "Joghurt";
    }

    public String toString() { // vgl. Aufgabe a)
        return marke + " Joghurt, " + gewicht + "g";
        // magic string π
    }
}

// Hinweis: Frucht ist eine Immutable-Klasse
public class Frucht {
    public final String sorte;
    public Frucht(String sorte) {
        this.sorte = sorte;
    }
}
```

2 a) [4 Punkte] Ergänzen Sie die Methode `toString()` von `Joghurt` so, dass die Ausgabe eines `Joghurt`-Objekts (im Bsp.: Marke `Jalmj`) in der Art `System.out.println (joghurt)`; eine Ausgabe nach dem Muster `Jalmj Joghurt, 480g` erzeugt.

12 b) [24 Punkte] Geben Sie die Deklaration einer Klasse `Fruchtjoghurt` an, die von `Joghurt` erbt. Die Klasse `Fruchtjoghurt` besitzt eine Instanzvariable vom Typ `Frucht` und eine Instanzvariable vom Typ `double`, in der der `fruchtanteil` (angegeben in %) gespeichert wird.

Die Klasse `Fruchtjoghurt` soll folgende Konstruktoren haben:

einen vollständigen Konstruktor, ✓

einen Konstruktor, mit einer Parameterliste wie der vollständige Konstruktor, jedoch mit einem `String`-Objekt zur Angabe der Fruchtart ✓

einen Kopierkonstruktor. ✓

Die Ausgabe eines `Fruchtjoghurt`-Objekts (im Bsp.: Marke `Fampf` und Produkt-Kategorie `Kirsch-Joghurt`) in der Art `System.out.println (fruchtjoghurt)`; soll eine Ausgabe nach dem Muster

`Fampf Kirsch-Joghurt, 128g (12,25% Frucht)` erzeugen. ✓

Vermeiden Sie unnötige Codevervielfachungen!

```
class Fruchtjoghurt extends Joghurt
```

```
// Attribute
```

```
protected Frucht frucht;  
protected double fruchtanteil;
```

```
// vollständiger Konstruktor
```

```
public Fruchtjoghurt(String marke, int gewicht, Frucht frucht,  
double fruchtanteil) {
```

```
    super(marke, gewicht);  
    this.frucht = new Frucht(frucht (frucht.sorte);  
    this.fruchtanteil = fruchtanteil;  
}
```

```
// Konstruktor mit Angabe der Sorte durch String .. (vgl. Aufgabe)
```

```
public Fruchtjoghurt(String marke, int gewicht, Frucht frucht,  
double fruchtanteil, String fruchtsorte sorte) {
```

```
super(marke, gewicht);  
this(marke, gewicht, frucht, fruchtanteil);
```

```
fruchtsorte = frucht.sorte; // ← virtuelles
```

```
// Kopierkonstruktor
```

```
public Fruchtjoghurt(Fruchtjoghurt other) {
```

```
    this(other.marke, other.gewicht, other.frucht,  
    other.fruchtanteil);  
}
```

```
// ....
```

```
public String toString() {
```

```
    return marke + " " + frucht.sorte + " Joghurt, " +  
    gewicht + " g " + ("Format" + fruchtanteil + "% Frucht");
```

getProductCat() (-4)

Aufgabe 7 (18 Punkte):

Hinweis zur Bewertung: Punktzahl = (richtige Antworten – falsche Antworten) * 1,5 P

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

trifft zu / trifft nicht zu

Stringvergleiche mit == können ein falsches Ergebnis liefern

super() ist ein Aufruf der Methode super

Interfaces sind primitive Typen

Jede Klasse muss toString() überschreiben

Eine abstrakte Klasse kann Konstruktoren haben

Eine abstrakte Klasse darf abstrakte Methoden enthalten

Die Klasse Frucht aus Aufgabe 6 hat einen Standardkonstruktor

Jede Bauplanklasse hat mindestens eine statische Methode

Immutables des Typs J können als Elemente in Listen des Typs ArrayList<J> gespeichert werden

Collection ist Supertyp von Set

Collections ist eine List von Collection-Elementen

ArrayList ist Subtyp von Collection

ohne Antwort / richtig / falsch / bewertet:

2 10 - 10

Punkte:

15