Klausuraufgaben MDB PR1

Aufgabe 1) Geben Sie jeweils eine vollständige Deklaration für die folgenden Methoden an:
a) [5 Punkte]

```
/* Der Rückgabewert gibt an, ob drei aufeinanderfolgende Zahlen a, b, c der Größe nach ansteigend
 * sortiert 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 Punkte]

```
/* 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)
```

c) [7 Punkte]

```
/* Gibt die Anzahl der Indices zurück, für die Array a und 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 und keine
 *Exceptions werfen!
 */
public static int equalCount (int[] a, int[] b)
```

d) [8 Punkte]

```
/* Bestimmt den Mittelwert aller Zahlen in einer Textdatei, in der ganze Zahlen, Gleitkommazahlen
 * und Texte in beliebiger Reihenfolge und jeweils durch Leerraum getrennt stehen. Ist die Datei
 * nicht vorhanden oder nicht lesbar oder hat die Datei das falsche Format, wird der Wert 0
 * zurückgegeben. Beachte: Die Methode soll für alle möglichen Parameterwerte korrekt
 * funktionieren und keine Exceptions werfen!
 */
public static double meanValueOf(File path)
```

Aufgabe 2) Vereinfachen Sie die Codefragmente a) bis c) so weit wie möglich, aber ohne die Funktionalität zu verändern:
a) [3 Punkte]

```
if (markerflag == false) {
        markerflag = true;
} else {
        markerflag = false;
}
```

b) [5 Punkte] // Beachte: Term kann auch einen negativen Wert haben!

```
public static int newCount (int count, int term) {
        for (int i = 0; i < term; i++) {
                count = count + 1;
        }
        return count;
}
```

c) [4 Punkte]

```
private void runSubMenu (char zeichnen) {
        switch (zeichnen) {
        case 'a':       runSubMenuA();
                        break;
        case 'A':       runSubMenuA();
                        break;
        case 'b':       runSubMenuB();
                        break;
        case 'B':       runSubMenuB();
                        break;
        }
}
```

Aufgabe 3)

a) [4 Punkte] Nennen Sie die vier grundlegenden Klassen des IO-Systems von Java:

b) [3 Punkte] Ordnen Sie den folgenden mit printf() verwendbaren Umwandlungszeichen die richtige Bedeutung (durch Ankreuzen) zu:

d	o depressed	o double	o decimal	o direct
f	o fixed	o formal	o fine	o float
s	o sign	o subclass	o string	o supertype

c) [4 Punkte] 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 98.7e03 ausgibt?

o ja, format hat folgenden Wert:

o nein, das geht nicht, weil

d) [4 Punkte] Geben Sie für die folgenden Methoden der Klasse Object jeweils den vollständigen Methodenkopf an:

equals

hashCode

e) [2 Punkte] Wofür steht die Abkürzung DRY?

Aufgabe 4a) [20 Punkte] 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 getPictureData ();
	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 Kopierkonsturktor und einen weiteren Konstruktor mit der Parameterliste (Dimension d, int farbtiefe) besiten. Die Ausgabe eines PictureData-Objektes 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, soll in der Ausgabe die Zeichenkette "SW-Bild" die Zeichenkette "Farbtiefe" ersetzen und die Ausgabe soll nach dem Muster *SW-Bild: 800 X 540* erfolgen.
Vermeiden Sie unnötige Codevervielfachungen.

Aufgabe 4b) [20 Punkte] Gegeben sei:

```
public abstract class BasicGewaechshaus {
	protected int t; // Temperatur in °C
	protected int lf; // relative Luftfeuchtigkeit in %

	public BasicGewaechshaus (int t, int lf) {
		this.t = t;
		this.lf = lf;
	}
```

```
	protected String getAirConditions () {
		return lf + "% @ " + t + "°C";
	}
	protected abstract String sizeToString();
	protected abstract String getPlantType();


	public String toString() {
		return "Gewaechshaus fuer " + getPlantType () + " [" + sizeToString() + ", " +
			getAirConditions() + " ]";
	}
}

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

Geben Sie die Deklaration einer Klasse Gewaechshaus an, die von Basic Gewaechshaus erbt und die nicht abstrakt ist. Die Klasse Gewaechshaus besitzt zwei Instanzvariablen vom Typ String bzw. Dimension, in der die pflanzenart bzw. die groesse gespeichert werden.
Die Klasse Gewaechshaus soll einen vollständigen Konstruktor, einen Kopierkonsturktor und einen weiteren Konstruktor mit der Parameterliste (int t, int lf, String plantType, int breite, int laenge) besiten. Die Ausgabe eines Gewaechshaus-Objektes in der Art System.out.println(gewaechshaus); soll eine Ausgabe nach dem Muster *grosses Gewaechshaus fuer Tomaten [20 X 100 qm, 33% @ 24°C]* erzeugen. Hat das Gewaechshaus eine Groesse (=laenge*breite) von mehr als 10 Quadratmeter, dann gilt es als grosses Gewaechshaus, ansonsten als kleines Gewaechshaus, dann soll die Ausgabe soll nach dem Muster kleines *Gewaechshaus fuer Moehren [1 X 2 qm, 41% @ 22°C]* erzeugt werden.
Vermeiden Sie unnötige Codevervielfachungen.

```
// Anwendungsbeispiel: Die Anweisungen
Dimension flaecheMoehrenbeet = new Dimension(1, 2);
BasicGewaechshaus tomatenPlantage
	= new  Gewaechshaus(24, 33, "Tomaten", new Dimension(20, 100);
BasicGewaechshaus moehrenGarten
	= new  Gewaechshaus(22, 41, "Moehren", flaecheMoehrenbeet);
System.out.println(tomatenPlantage);
System.out.println(moehrenGarten);
```

// erzeugen die Ausgabe:
grosses Gewaechshaus fuer Tomaten [20 X 100 qm, 33% @ 24°C]
Gewaechshaus fuer Moehren [1 X 2 qm, 41% @ 22°C]

Aufgabe 5) Geben Sie ohne Begründung an, welche der folgenden Aussagen zutreffen und welche nicht zutreffen:

	trifft zu	trifft nicht zu
Eine abstrakte Klasse kann nicht mit new instanziiert werden	o	o
Eine abstrakte Klasse muss mit new instanziiert werden	o	o
Eine abstrakte Klasse kann mit new instanziiert werden	o	o
Eine abstrakte Klasse hat immer mindestens eine abstrakte Methode	o	o
Eine abstrakte Klasse muss eine abstrakte Methode haben	o	o
Exceptions sind Interfaces	o	o
Arrays sind Referenztypen	o	o
Ein Objekt kann mehrere Typen haben	o	o
Ein Package kann mehrere Interfaces implementieren	o	o
Eine lokale Variable darf nicht protected sein	o	o
Eine lokale Variable darf nicht public sein	o	o
Eine lokale Variable kann private sein	o	o
Eine lokale Variable kann nicht public sein	o	o
Die Klasse aller Objekte ist zur Compilezeit bekannt	o	o
Der Typ von allen Variablen ist zur Compilezeit bekannt	o	o
Genaugenommen kann keine Variable ein Objekt speichern	o	o
Es kann sein, dass eine Variable eine Referenz auf ein Objekt speichert	o	o
Der Standardkonstruktor hat den Rückgabetyp void	o	o
Der Kopierkonstruktor hat immer den Rückgabetyp void	o	o
Ein Kopierkonstruktor kann den Rückgabetyp void haben	o	o
Alle Methoden eines Interfaces müssen abstract sein	o	o
Ein Package kann mehrere Interfaces enthalten	o	o
Packages können Exceptions enthalten	o	o
int[] ist ein Referenztyp	o	o
double ist ein Referenztyp	o	o
Jede Klasse außer java.lang.Object implementiert ein Interface	o	o
Jede Klasse außer java.lang.Object hat eine Superklasse	o	o
Ein Interface hat genau eine Superklasse	o	o
Eine Klasse kann mehrere Interfaces implementieren	o	o
Ein Kopierkonsturktor kann einen Parameter vom Typ int haben	o	o

Aufgabe 6) [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 Aufgabe 07 {
	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 (Exc 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.