Semesterschluss-Prüfung Programmierung 1

*Es sind **alle** Aufgaben und Teilaufgaben zu lösen.*
Versuchen Sie, Ihre Lösungen soweit wie möglich direkt auf diese Aufgabenblätter zu notieren; selbstverständlich dürfen Sie aber auch noch weitere Lösungsblätter hinzufügen.

Das P1-Team wünscht allen Studierenden viel Glück !

Gesamtpunktzahl:

Aufgabe 1 (Multiple Choice; 8 Punkte)

Bei den nachfolgenden Teilaufgaben ist **jeweils genau eine Antwort richtig**.

Teilaufgabe 1.1 (2 Punkte)

Gegeben sei folgendes Programmfragment:

```
if (alpha >= beta)
    charlie = delta;
else
    epsilon = foxtrot;
```

Welches der folgenden Programmstücke ist äquivalent dazu ?

☐
```
if (beta <= alpha)
    charlie = delta;
else if (beta > alpha)
    epsilon = foxtrot;
```

☐
```
if (alpha >= beta);
    charlie = delta;
if (alpha < beta);
    epsilon = foxtrot;
```

☐
```
if (beta >= alpha)
    charlie = delta;
else
    epsilon = foxtrot;
```

☐
```
if (beta < alpha)
   charlie = delta;
else
   epsilon = foxtrot;
```

Teilaufgabe 1.2 (2 Punkte)

Gegeben sei die folgende Klasse:

```
public class Counter
{
     private static int count = 0;

     public static int getCount()
     {
          return count;
     }

     public void incrementCount()
     {
          count++;
     }
}
```

Welches ist der Output des folgenden Programmstücks ?

```
Counter counter1 = new Counter();
Counter counter2 = new Counter();

counter1.incrementCount();
counter2.incrementCount();
counter1.incrementCount();

System.out.println("The value of count is " +counter1.getCount());
```

☐ **The value of count is 1**

☐ **The value of count is 3**

☐ **The value of count is 2**

☐ **The value of count is 0**

Teilaufgabe 1.3 (2 Punkte)

Gegeben seien folgende 3 Klassen:

```
class A {
     public void show() {
          System.out.println( "Ich bin eine A-Methode" );
     }
}

class B extends A {
     public void show() {
          System.out.println( "Ich bin eine B-Methode" );
     }
}

class C extends A {
     public void show( int number ) {
       System.out.println("Ich bin eine C-Methode mit Argument" );
     }
}
```

Welcher Output wird durch folgenden Code generiert?

```
A a1 = new A();
A a2 = new B();
A a3 = new C();
a1.show();
a2.show();
a3.show();
```

☐ Ich bin eine A-Methode
Ich bin eine A-Methode
Ich bin eine A-Methode

☐ Ich bin eine A-Methode
Ich bin eine B-Methode
Ich bin eine C-Methode mit Argument

☐ Ich bin eine A-Methode
Ich bin eine B-Methode
Ich bin eine A-Methode

☐ Ich bin eine B-Methode
Ich bin eine B-Methode
Ich bin eine B-Methode

Teilaufgabe 1.4 (2 Punkte)

Gegeben sei das folgende Code-Fragment:

```
int[] x = {5,6,7,8,9};
int[] y = x;
y[2] = 3;
```

Welche der folgenden Aussagen trifft nach der Ausführung des Programm-Stücks zu?

☐ `x[2]` hat den Wert 7

☐ `x[2]` hat den Wert 6

☐ `x[2]` hat den Wert 3

☐ `y[3]` hat den Wert 7

Aufgabe 2 (2 Punkte)

Das folgende Codefragment hat zwei schwerwiegende Fehler. Korrigieren Sie diese!

```
int[] numbers = {1,2,3,7,9,11};
int i = numbers.length;
while (i != 0) {
    System.out.println( numbers[i] + numbers[i-1] );
    i = i-2;
}
```

Aufgabe 3 (3 Punkte)

Was ist der Output bei der Ausführung des folgenden Code-Fragments ?

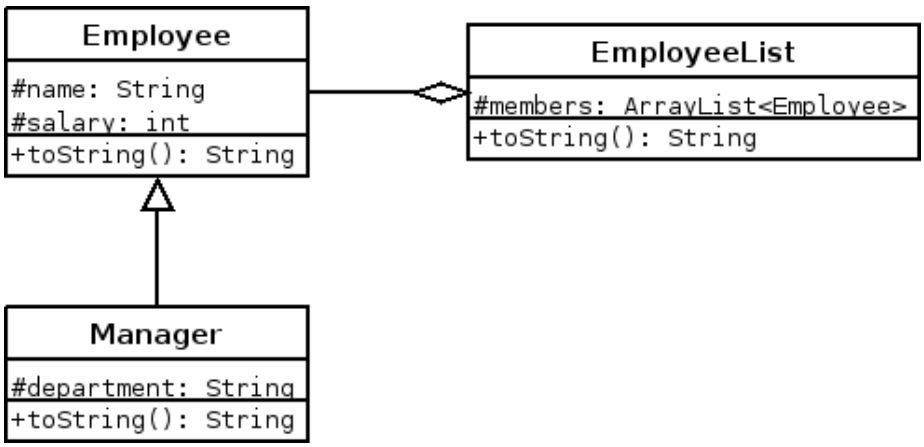
```
public class Exceptions {

	Exceptions() {
		try {
			for (int n = 3; n >= -3; n--)
				System.out.println(100 / n);
		} catch (ArithmeticException e) {
			System.out.println("0-Divide!");
		} finally {
			System.out.println("Done");
		}
	}

	public static void main(String[] args) {
		new Exceptions();
	}
}
```

Aufgabe 4 (10 Punkte)

Schreiben Sie drei Klassen **`Employee`**, **`Manager`** und **`EmployeeList`**. Schreiben Sie die Klassendefinitionen und Methoden gemäss UML Diagramm sowie einen Konstruktor pro Klasse. Die Methode **`toString`** soll jeweils eine vernünftige Repräsentation des Objektes zurückgeben.



(Platz für die Lösung der Aufgabe 4)

Aufgabe 5 (14 Punkte)

Sie sollen eine Klasse **Arithmetic** definieren, deren Zweck die Zerlegung von ganzen Zahlen grösser als 1 in ihre einzelnen Faktoren ist.

Die Klasse enthalte mindestens die beiden Attribute **number** und **factors**, einen Konstruktor mit einem Parameter, sowie die Methoden **split** und **toString:**

Attribute:

- **`private int number`**
  enthalte die ganze Zahl > 1, die zerlegt werden soll.
- **`private ArrayList factors`**
  enthalte alle einzelnen Faktoren von **number**, die grösser als 1 sind. Kommt eine Zahl mehrmals als Faktor vor, so soll die ArrayList entsprechend viele Komponenten enthalten.

Konstruktor (3 Punkte)

- **`Arithmetic(int i)`**
  initiiere zuerst mit Hilfe des Parameters das Attribut **number** und danach mit Hilfe von **split** das Attribut **factors.**

Methoden (10 Punkte)

- **`private void split()`**
  zerlege das Attribut **number** in seine Faktoren – beginnend mit 2 und in Schritten von 1 fortschreitend – und baue damit das Attribut **factors** auf.
- **`public String toString()`**
  ergebe auf dem Bildschirm die Darstellung der aktuellen Zerlegung in der folgenden Form.

  Beispiel: **`84 = 2 * 2 * 3 * 7`**

  (Faktoren der Grösse nach aufsteigend geordnet)

(1 Punkt)
Beachten Sie, dass bei Generierung eines Objekts der Klasse **Arithmetic** auch eine ganze Zahl ≤ 1 als aktueller Parameter verwendet werden kann, und berücksichtigen Sie solche Fälle an geeigneter Stelle.

Hinweis:
*Eine Zahl **t** heisst Faktor (Teiler) einer ganzen Zahl **n**, falls gilt:*
n % t == 0 *(Java)* bzw. n mod t = 0 *(Math)*

Für korrekte Syntax erhalten Sie maximal 4 Punkte.

(Platz für die Lösung der Aufgabe 5)

Aufgabe 6 (3 Punkte)

Ergänzen Sie die folgende Aussage:

Eine Klasse erweitert höchstens eine ..... und kann mehrere ..... implementieren. Eine ..... Klasse hat keine Instanzen; sie kann aber einige ihrer ..... implementieren. Die Neudefinition einer Methode in einer Unterklasse nennt man ..... Das Verhalten einer solchen Methode hängt ab vom ..... des aufrufenden Objekts.

Aufgabe 7 (3 Punkte)

Die folgende **rekursive** Methode berechnet den Umkehrstring eines gegebenen Strings, d.h. den gegebenen String in umgekehrter Reihenfolge. Welche Zeile muss an der Stelle (***) eingefügt werden, damit die Methode korrekt arbeitet ?

```
String reverse(String s) {
      String result;
      if (s.length() <= 1)
            result = s;
      else
            // (***)
      return result;
```

Aufgabe 8 (4 Punkte)

Implementieren Sie eine Methode **`removeChar`**, die einen **`String`** und ein **`char`** entgegennimmt, alle Zeichen gleich diesem **`char`** herausstreicht und diesen bereinigten **`String`** wieder zurückgibt.

Bsp: **`removeChar( "Programmieren", 'r' ) => "Pogammieen"`**

Aufgabe 9 (3 Punkte)

Die Klasse **`ThonSandwich`** erweitere die Klasse **`Sandwich`**. Welche der folgenden Zuweisungen sind erlaubt ? Für eine richtige Antwort gibt es +0.5 Punkte, für eine falsche -0.5 Punkte. Eine negative Gesamtsumme wird auf 0 aufgerundet.

	erlaubt	**nicht erlaubt**
`Sandwich x = new Sandwich();`	☐	☐
`ThonSandwich y = new ThonSandwich();`	☐	☐
`x = y;`	☐	☐
`y = x;`	☐	☐
`y = new Sandwich();`	☐	☐
`x = new ThonSandwich();`	☐	☐

Aufgabe 10 (5 Punkte)

Geben Sie bei folgenden Aussagen an, ob Sie **wahr** oder **falsch** sind; für eine richtige Antwort gibt es +1 Punkt, für eine falsche -1 Punkt. Eine negative Gesamtsumme wird auf 0 aufgerundet.

	wahr	**falsch**
Eine abstrakte Klasse, die ein Interface implementiert, muss alle Methoden dieses Interfaces implementieren.	☐	☐
Der Wert einer **`static`**-Variable kann in jeder Instanz anders sein.	☐	☐
Nur konkrete Klassen können instanziert werden, abstrakte nicht.	☐	☐
Eine statische Methode ist nur innerhalb ihrer Klasse sichtbar.	☐	☐
Eine abgeschlossene (final) Klasse kann keine Interfaces implementieren.	☐	☐