

Name:

Matrikelnummer:

Bachelorprüfung: Objektorientierte Softwareentwicklung

WS10/11

Erlaubte Hilfsmittel: keine

Lösung ist auf den Klausurbögen anzufertigen. (eventuell Rückseiten nehmen)

Bitte legen Sie einen Lichtbildausweis und den Studentenausweis auf den Tisch.

Bearbeitungszeit: 90 Minuten

Unterschrift

Benotung

Aufgabe:	1	2	3	4	5	Gesamt	Note
Punkte:	20	20	20	20	20	100	
erreicht:							

Aufgabe 1 (20 Punkte)**(Modellieren und Schreiben von Klassen)**

Sie sollen in dieser Aufgabe Klassen entwickeln, die es erlauben, Rauminformationen zu speichern.

- a) Entwickeln Sie eine Klasse **Raum**. Ein Objekt dieser Klasse repräsentiert einen Raum in einem Gebäude. Es soll zwei Eigenschaften haben: die Raumnummer, die durch eine Zeichenkette dargestellt ist und die Anzahl der Sitzplätze in dem Raum.

Schreiben Sie einen Konstruktor und überschreiben Sie die Methode `equals` auf adequate Weise. Überschreiben Sie auch die Methode `toString` auf sinnvolle Weise.

- b) Schreiben Sie eine Unterklasse **Rechnerraum** der Klasse **Raum**. Rechneräume haben an jedem Arbeitsplatz einen PC. Die Klasse **Rechnerraum** soll als zusätzliche Eigenschaft einen Aufzählungswert haben, der angibt, welches Betriebssystem auf den Rechnern installiert ist. Gültige Aufzählungswerte sollen sein: Unix, Linux, Windows, MacOS.

Die Methode `toString` soll überschrieben werden, um die zusätzliche Information mit zu berücksichtigen.

- c) Schreiben Sie eine Klasse **Gebäude**. Diese Klasse soll als Eigenschaft eine Liste von Räumen enthalten. Benutzen Sie hierzu Klassen aus dem Paket `java.util`.

Schreiben Sie eine Methode `int wievielPlatz(String raum)`, die angibt, wieviel Sitzplätze der Raum mit der übergebenen Raumnummer hat.

Name:

Matrikelnummer:

Name:

Matrikelnummer:

Name:

Matrikelnummer:

Aufgabe 2 (20 Punkte)
(Programmfluss)

- a) Führen Sie die folgende Klasse von Hand aus. Schreiben Sie dabei auf, mit welchen Werten die einzelnen Variablen während des Programmdurchlaufs belegt sind. Schreiben Sie auf, was auf dem Bildschirm ausgegeben wird.

```

                                Aufgabe2a.java
1 class Aufgabe2a{
2     public static void main(String [] args){
3         for (int x= 42; x> 12;x=x-6){
4             System.out.println(x);
5             switch (x%5){
6                 case 4: x=x-1;
7                 case 3: x=x-2;
8                 default: x= x-3;
9             }
10        }
11    }
12 }
```

Name:

Matrikelnummer:

b) Betrachten Sie folgende Klasse:

```

1 class Aufgabe2b{
2     static int f(int x,int y){
3         return g(y+1,x+1);
4     }
5     static int g(int x,int y){
6         if (x==y) return x;
7         if (y%x == 0) return x-y;
8         return f(y,x);
9     }
10    public static void main(String [] args){
11        System.out.println(g(3,16));
12    }
13 }
```

Berechnen Sie schrittweise das Ergebnis des Ausdrucks $g(3,16)$.

Name:

Matrikelnummer:

Aufgabe 3 (20 Punkte)

(Stringverarbeitung)

Schreiben Sie folgende Funktionen auf Zeichenketten:

- a) Für Strings ist eine Funktion:

```
static int commonPrefixSize(String str1, String str2)
```

zu implementieren. Es soll berechnet werden, wieviel Buchstaben die beiden Strings einen gemeinsamen Anfang haben.

Beispiel:

`commonPrefixSize("erwin", "erwartung")` ist 3, weil die Wörter den gemeinsamen Anfang `erw` haben.

Name:

Matrikelnummer:

b) Für Strings ist eine Funktion:

```
static String doubleEveryLetter(String str1);
```

zu implementieren. Es soll ein String zurückgegeben werden, der aus den Parameter `str1` berechnet werden soll, indem jeder Buchstabe verdoppelt wird.

Beispiel:

`doubleEveryLetter("theo")` ist `tthheeo`.

Name:

Matrikelnummer:

Aufgabe 4 (20 Punkte)
(rekursive Strukturen)

Gegeben Sei folgende Schnittstelle von einfach verketteten Listen, wie aus der Vorlesung bekannt.

```
----- List.java -----  
1 public interface List<A> {  
2     A getElement();  
3     List<A> getNext();  
4  
5     boolean isEmpty();  
6     void add(A a);  
7 }
```

Implementieren Sie folgende Funktionen rekursiv:

a) `static String unwords(List<String> xs);`

Es soll ein String erzeugt werden, in dem alle Listenelemente hintereinander durch Leerzeichen getrennt auftreten. Lösen Sie diese Aufgabe rekursiv.

Name:

Matrikelnummer:

b) `static <A> boolean contains(List<A> xs, A x);`

Es soll genau dann wahr sein, wenn die Liste `xs` ein Element enthält, das gleich dem Objekt `x` ist. Lösen Sie diese Aufgabe iterativ.

Name:

Matrikelnummer:

Aufgabe 5 (20 Punkte)

Zusammenhänge

Erklären Sie in kurzen Worten.

- a) Wie manifestiert sich in Java der Unterschied zwischen den Begriffen »dasselbe« und »das Gleiche«?

- b) Was ist gegenüber anderen Exception-Klassen an RuntimeException besonders?

Name:

Matrikelnummer:

c) Was versteht man unter einer anonymen Klasse?

d) Was ist Überschreiben und was ist Überladen von Methoden?

Name:

Matrikelnummer:

- e) Was ist eine abstrakte Klasse und was ist bei der Benutzung einer abstrakten Klasse zu berücksichtigen?