

Klausur: PRGI MD

Aufgabe 1 (18 Punkte)

- 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.

```
_____ Aufgabel1a.java _____  
1 class Aufgabel1a{  
2     public static void main(String [] args){  
3         for (int i=11;i>-1;i=i-9){  
4             i=i-1;  
5             for (int j=0;j<i;j=j+4){  
6                 System.out.print(i);  
7             }  
8             System.out.println("");  
9             i=i+1;  
10            if (i<=4) break;  
11        }  
12    }  
13 }
```

- b) Betrachten Sie folgende Klasse:

```
_____ Aufgabel1b.java _____  
1 class Aufgabel1b{  
2     static int dreier(int x){  
3         if (x<=0) return 0;  
4         if (x%3==0) return x+dreier(x-1);  
5         return dreier(x-2);  
6     }  
7 }
```

Reduzieren Sie den Ausdruck `dreier(9)` zu seinem Ergebnis.

Aufgabe 2 (24 Punkte)

Die folgenden Javaprogramme enthalten Fehler. Beschreiben Sie die Fehler und korrigieren Sie sie:

```
_____ Aufgabe2a.java _____  
a) class Aufgabe2a {  
2     static public void main(String [] args){  
3         System.out.println(addHallo("Illja"));  
4     }  
}
```

Name:

Matrikelnummer:

```
5
6 public String addHallo(Object otherString){
7     return "hallo "+otherString;}
8 }
```

Aufgabe2b.java

```
b) class Aufgabe2b {
2     private static String name = "Shakespeare";
3     public static void main(String [] args) throws Exception{
4         Object o = new Aufgabe2b().name;
5         System.out.println(o+42);
6     }
7 }
```

Aufgabe2c1.java

```
c) class Aufgabe2c1{
2     String name ;
3
4     Aufgabe2c1(String name){this.name=name;}
5 }
```

Aufgabe2c2.java

```
1 class Aufgabe2c2 extends Aufgabe2c1{
2     Aufgabe2c2(){ }
3 }
```

Aufgabe2d.java

```
d) class Aufgabe2d
2     extends Aufgabe2b, Aufgabe2a
3     implements Comparable {
4     public static void main(String [] args){
5     }
6 }
```

Aufgabe2e1.java

```
e) class Aufgabe2e1{ }
```

Aufgabe2e2.java

```
1 class Aufgabe2e2 extends Aufgabe2e1{
2     String name ;
3
4     Aufgabe2e2(String name){this.name=name;}
5     public static void main(String [] args){
6         Aufgabe2e1 a = new Aufgabe2e2("Richter");
7         System.out.println(a.name);
8     }
9 }
```

Name: _____

Matrikelnummer: _____

```
_____ Aufgabe2f.java _____  
f) class Aufgabe2f extends Object{  
  2   String toString(){  
  3       return "Aufgabe2f";  
  4   }  
  5 }
```

Aufgabe 3 (24 Punkte)

Gegeben sei die folgende abstrakte Klasse für Listen, gemäß unserer Spezifikation aus der Vorlesung.

```
_____ AbList.java _____  
1 abstract class AbList{  
2     abstract public AbList empty();  
3     abstract public AbList cons(Object x, AbList xs);  
4     abstract public boolean isEmpty();  
5     abstract public Object head();  
6     abstract public AbList tail();  
7 }
```

Schreiben Sie für die Klasse folgende Methoden, die ihr hinzugefügt werden können:

- a) `AbList upperCaseStrings()`: erzeugt eine neue Liste, indem für jedes Element in einem String in Großbuchstaben umgewandelt wird.
Beispiel: `("Aa", "i1i", "l").upperCaseStrings()` ergibt: `("AA", "I1I", "L")`.
- b) Sei zusätzlich folgende Schnittstelle gegeben:

```
_____ Condition.java _____  
1 interface Condition {  
2     boolean condition(Object o);  
3 }
```

Implementieren Sie für die Klasse `AbList` die Methode:

`AbList selectNegative(Condition cond)`: erzeugt eine neue Liste, die genau die Objekte `o` der `this`-Liste enthält, für die gilt `cond.condition(o)` ergibt `false`.

- c) `public boolean equals(AbList other)`: ergibt `true`, wenn beide Listen gleiche Elemente in gleicher Reihenfolge haben.

Aufgabe 4 (14 Punkte)

Schreiben Sie eine Methode

```
public static boolean charIsInString(char c, String str)
```

Das Ergebnis soll genau dann `true` sein, wenn ein Zeichen im `String str` das Zeichen `c` ist.

Beispiel: `charIsInString("azuihü", 'ü')` ergibt `true`.

Name:

Matrikelnummer:

Aufgabe 5 (20 Punkte)

- a) Entwerfen und implementieren Sie eine abstrakte Klasse, für die Beschreibung von geometrischen Figuren. Eine geometrische Figur habe dabei folgende Funktionalität:
- eine Methode für die Berechnung des Flächeninhalts.
 - eine Methode, die prüft, ob die geometrische Figur den gleichen Flächeninhalt wie eine andere geometrische Figur hat.
- b) Modellieren und implementieren Sie eine Klasse für Rechtecke. Ein Rechteck sei dabei eine geometrische Figur mit den folgenden Eigenschaften:
- einer Breite
 - einer Höhe
- Ein Objekt Ihrer Rechteckklasse soll dabei folgende Funktionalität anbieten:
- ein Test, ob es sich um ein Quadrat handelt.
- c) Modellieren und implementieren Sie eine Klasse für Kreise. Ein Kreis sei daei eine geometrische Figur. Ein Kreis habe einen Radius.
- d) Schreiben Sie ein kleines Testprogramm, in dem Sie von ihren beiden Klassen je ein Objekt erzeugen, und die Methoden dieser Objekte ausprobieren.

Benotung

Aufgabe:	1	2	3	4	5		Summe	Übungen	Gesamt
Punkte:	18	24	24	14	20		100	10	
erreicht:									