

## Hauptklausur: PRGI MD

### Aufgabe 1 (18 Punkte)

Führen Sie die folgende Klassen 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.

```
a) class Aufgabe1a{
    public static void main(String [] args){
        for (int j=3;j>=0;j=j-1){
            int ein = j;
            do {
                System.out.println(ein % 2);
                ein = ein / 2;
            }while (ein > 0);
            System.out.println("");
        }
    }
}

b) class Aufgabe1b{
    static int rechne(int i){
        int j=3
        for (;i<45;){
            j=j+1;
            if (j==5) return -42;
            i=i+1;
        }
        return j*i;
    }

    public static void main(String [] args){
        System.out.println(rechne(42));
    }
}
```

### Aufgabe 2 (24 Punkte)

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

```
a) class Aufgabe2a {
    static void printNumber(int z){
        System.out.println(z.toString());
    }
}
```

```
b) class Aufgabe2b {
    String name = "Shakespeare";
    public static void main(String [] args){
        System.out.println(Aufgabe2b.name);
    }
}

c) class Aufgabe2c {
    static String doppel(String x){
        return x+x;
    }

    static public Object trace(Object o){
        System.out.println(o);
        return o;
    }

    public static void main(String [] args){
        System.out.println(doppel(trace("Ilja")));
    }
}

d) class Aufgabe2d{
    static public int getAtIndex(int [] a,int i){
        try {
            if (i<0) return a[0];
            return a[i];
        }catch (ArrayIndexOutOfBoundsException _){
            System.out.println("Index ist zu groß");
        }
    }
}

e) interface Aufgabe2e_1{
    String getName();
}

class Aufgabe2e_2 implements Aufgabe2e_1{
    String name ;

    Aufgabe2e_2(String name){this.name=name;}

    String getName(){
        return name;
    }
}
```

```
f) package de.tfh.berlin.panitz;

    public class Aufgabe2f_1{
        public void printString(String s){
            System.out.println(s);
        }
    }

    package de.tfh.berlin.panitz.aufgaben;
    public class Aufgabe2f_2{
        public static void main(String [] _){
            new Aufgabe2f_1().printString("hallo welt");
        }
    }
}
```

**Aufgabe 3 (28 Punkte)**

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

```
abstract class AbList{
    abstract public AbList empty();
    abstract public AbList cons(Object x, AbList xs);
    abstract public boolean isEmpty();
    abstract public Object head();
    abstract public AbList tail();
}
```

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

- a) `boolean contains(Object o)`: ergibt `true`, wenn ein Element der Liste gleich dem Parameter `o` ist.  
Beispiel: `("a", "b", "a", "d").contains("a")` ergibt: `true`.
- b) `int howOften(Object o)`: zählt, wie oft ein gleiches Object zum Argument `o` in der Liste ist.  
Beispiel: `("a", "b", "a", "d").howOften("a")` ergibt: `2`.
- c) `AbList add(Object o)`: gibt eine neue Liste zurück, in die das Objekt `o` vorne eingefügt wird, wenn es noch nicht in der `this`-Liste enthalten ist.  
Beispiel: `("a", "b", "a", "d").add("a")` ergibt: `("a", "b", "a", "d")`,  
`("b", "d").add("a")` ergibt: `("a", "b", "d")`.
- d) `AbList oddIndexElements()`: gibt eine neue Liste zurück, die nur jedes zweite Element der `this`-Liste hat.  
Beispiel: `("a", "b", "c", "d").oddIndexElements()` ergibt `("a", "c")`.

Name:

Matrikelnummer:

---

#### Aufgabe 4 (17 Punkte)

Schreiben Sie eine Methode

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

die wahr ist, wenn an jedem ungeraden Index das Zeichen `c` im String `str` ist.

Beispiel: `everySecondIs("acdcg", 'c')` ergibt `true`.

`everySecondIs("acdcghj", 'c')` ergibt `false`.

#### Aufgabe 5 (13 Punkte)

Betrachten Sie sich folgende Klassen:

```
class Aufgabe5_1{
    static int i = 0;
    public String getInfo(){i=i+1;return "5_1";}
}

class Aufgabe5_2 extends Aufgabe5_1{
    static int i = 0;
    public String getInfo(){i=i+1;return "5_2";}
}

class Aufgabe5 {
    static String getInfo(Aufgabe5_1 o){return o.getInfo();}

    public static void main(String [] args){
        Aufgabe5_1 a1 = new Aufgabe5_1();
        Aufgabe5_1 a2 = new Aufgabe5_2();
        System.out.println(a1.getInfo());
        System.out.println(getInfo(a1));
        System.out.println(getInfo(a2));
        System.out.println(a2.getInfo());
        System.out.println(Aufgabe5_1.i);
    }
}
```

Führen Sie die Methode `main` von Hand aus. Was wird auf dem Bildschirm ausgegeben?

### Benotung

Aufgabe:	1	2	3	4	5	Summe	Übungen	Gesamt
Punkte:	18	24	28	17	13	100	10	
erreicht:								