

## Nachklausur: 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 i=3;i<11;i=i+2){
            i=i-1;
            System.out.print(i+" ");
            i=i+1;
        }
    }
}

b) class Aufgabe1b{
    static void drucke(String [] xs){
        int j=xs.length-1;
        for (;true;){
            System.out.println(xs[j]);
            j=j-2;
            if (j<0)break;
        }
    }

    public static void main(String [] args){
        String [] ys = {"a","b","c","d","e","f","g","h","i"};
        drucke(ys);
    }
}

c) class Aufgabe1c{
    public static void main(String [] args){
        int j=1;
        while (true) {
            for (int i=0;i<10;i=i+j){
                System.out.println(i);
            }
            System.out.println(j);
            j=j+1;
            if (j>3) break;
        }
    }
}
```

**Aufgabe 2 (20 Punkte)**

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

- a) class Aufgabe2a {  
    public String getValue(){return "hello";}  
  
    static public void main(String [] args){  
        System.out.println(Aufgabe2a.getValue());  
    }  
}
- b) class Aufgabe2b {  
    String name = "Shakespeare";  
    public static void main(String [] args){  
        System.out.println(Aufgabe2b.name);  
    }  
}
- c) class Aufgabe2c {  
    String toString(){  
        return "Aufgabe2c";  
    }  
  
    public static void main(String [] args){  
        System.out.println(new Aufgabe2c().toString());  
    }  
}
- d) class Aufgabe2d{  
    static public int getAtIndex(int [] a,int i){  
        try {  
            if (0<i && i<a.length) return a[i];  
        }catch (ArrayIndexOutOfBoundsException \_){  
            return a[0];  
        }  
    }  
}
- e) class Aufgabe2e{  
    String name ;  
  
    Aufgabe2e(String name){this.name=name;}  
  
    int calculateName(){  
        Integer i = (Integer) name;  
        return i.intValue();  
    }

```

    }
}

f) class Aufgabe2f{
    public void f1(String [] args){
        int i = 6;
        private void f2(String s){
            System.out.println(s+1);
        }
        f2(args[i]);
    }
}

```

### Aufgabe 3 (27 Punkte)

Gegeben sei die folgende abstrakte Klasse für Listen, gemäß unserer Spezifikation aus der Vorlesung. Listenelemente sind ganze Zahlen:

```

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

```

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

- a) `int sum( )`: ergibt die Summe der Zahlen, die in der Liste gespeichert sind.  
Beispiel: `(1,2,3,4).sum()` ergibt: 10.
- b) `AbList squareAll()`: erzeugt eine neue Liste, die die Quadratzahlen der in der Liste enthaltenen Zahlen enthält.  
Beispiel: `(1,2,3,4).squareAll()` ergibt: `(1,4,9,16)`.
- c) `int maximum()`: gibt die größte Zahl in der Liste zurück.  
Beispiel: `(1,4,2,3).maximum()` ergibt: 4.

### Aufgabe 4 (19 Punkte)

Schreiben Sie eine Methode

```
public static int diagonalSum(int [][] a)
```

die für ein zweidimensionales Feld, die Summe der Zahlen auf der Diagonalen .

Beispiel:

für

```
int [][] xs= {{1,76,78,2},{767,2,687,67868},{-768,7,3,78},{676,3,0,4}};
```

ergibt `diagonalSum(xs)` die Zahl 10.

Name:

Matrikelnummer:

---

### Aufgabe 5 (16 Punkte)

Betrachten Sie die folgende Klasse:

```
class Aufgabe6{
    int g1(int i){
        NullPointerException e = new NullPointerException();
        if (i>40) return 15;
        throw e;
    }

    int g2(int k)throws Exception{
        try {
            return g1(k+16);
        }catch (NullPointerException _){
            throw new Exception();
        }
    }

    String g3(int k){
        String result="";
        try {
            result=g2(k)+"";
            result=g2(k-10)+"";
        }catch (NullPointerException _){
            return result+"null pointer";
        }catch (Exception _){
            return result+"exception";
        }
        return result;
    }

    public static void main(String [] args){
        System.out.println(new Aufgabe6().g3(30));
    }
}
```

Führen Sie das Programm von Hand aus. Was wird auf dem Bildschirm ausgegeben? Erklären Sie wie es zu dieser Ausgabe kommt.

### Benotung

<b>Aufgabe:</b>	1	2	3	4	5		Summe		Übungen	Gesamt
<b>Punkte:</b>	18	20	27	19	16		100		10	
<b>erreicht:</b>										