

Programmieren OMB – Eisenbiegler

Aufgabe 1 (22 Punkte)

In dieser Aufgabe soll eine statische Methode programmiert werden, die zu einem String *x* und zu einem String *y* bestimmt, wie oft der String *y* in *x* als Teilstück enthalten ist. Parameter der Methode sind die beiden Strings *x* und *y*. Die Methode hat einen Rückgabewert vom Typ *int*.

Beispielläufe:

X	Y	Rückgabewert
„Zement“	„me“	1
„Besserwisser“	„sser“	2
„Hund“	„Bär“	0

Hinweise:

s.length() Länge des Strings *s*

s.charAt(i) das *i*-te Zeichen im String *s* (wobei $0 \leq i < s.length()$)

Aufgabe 1 (20 Punkte)

Schreiben Sie eine Klassenmethode *f*, die zu zwei Arrays von *int*- Werten *a* und *b* bestimmt, ob es eine Zahl gibt, die sowohl im Array *a* als auch im Array *b* enthalten ist. Die beiden Parameter von *f* haben den Typ *int[]*, der Rückgabewert ist vom Typ *boolean*.

Aufgabe 2 (15 Punkte)

```
public class X {  
  
    public static void f(int x) {  
        try {  
            System.out.println("A" + x);  
        }  
        try {  
            System.out.println("B" + x);  
            if (x==4) {  
                System.out.println("C" + x);  
                throw new Exception();  
            }  
            System.out.println("D" + x);  
        } catch (Exception e) {  
            System.out.println("E" + x);  
        }  
        System.out.println("F" + x);  
        if (x>2)  
            throw new Exception();  
        System.out.println("G" + x);  
    } catch (Exception e) {  
        System.out.println("H" + x);  
    }  
}  
  
    public static void main (String[] args) {  
  
        f(1);  
        f(3);  
        f(4);  
  
    }  
}
```

Zu welcher Ausgabe kommt es bei der Ausführung von X.main?

Aufgabe 2 (10 + 5 Punkte)

```
public class A {  
  
    public static int f(int[] x, int i) throws Exception {  
        if ((2 * i < x.length) && (i >= 0))  
            return x[2 * i];  
        else  
            throw new Exception();  
    }  
  
    public static void g(int[] x, int i) {  
        try {  
            System.out.println(f(x, i));  
        } catch (Exception e) {  
            System.out.println("Fehler");  
        }  
    }  
  
    public static void main(String[] args) {  
        try {  
            int x[] = { 1, 7, 3, 5 };  
            g(x, 0);  
            g(x, 1);  
            g(x, 5);  
        } catch (Exception e) {  
            System.out.println("ERROR");  
        }  
    }  
}
```

- A) Welche Ausgabe wird bei der Ausführung der main-Methode von Tester erzeugt?
- B) Was passiert bei folgendem Aufruf? Beschreiben Sie den Ablauf detailliert.

```
A.g(null, 4);
```

Aufgabe 4 (20 Punkte)

```
public class Test {  
  
    public static void main(String[] args) {  
        int x[] = { 2, 17, 5, 3, 4, 2 };  
        int r = 0;  
        while (r > -1) {  
            System.out.println(r);  
            r = x[x[r]];  
        }  
    }  
}
```

Zu welcher Ausgabe kommt es bei der Ausführung von Test.main ?

Aufgabe 4 (20 Punkte)

```
public class B {  
  
    public static void main(String[] args) {  
        for (int y = 0; y < 8; y++)  
            try {  
                System.out.println("y= " + y);  
                int z = f(y);  
                System.out.println("z= " + z);  
            } catch (Exception e) {  
                System.out.println("Fehler bei y= " + y);  
            }  
    }  
  
    public static int x[] = new int[] { 5, 4, 3, 2, 1 };  
  
    public static int f(int y) {  
        int k = x[y];  
        System.out.println("k= " + k);  
        for (int j = 1; j < 3; j++) {  
            k = k * x[y + j];  
            System.out.println("k= " + k);  
        }  
        return k;  
    }  
}
```

Eine `ArrayIndexOutOfBoundsException` tritt immer dann auf, wenn man auf eine unzulässige Array-Position zugreift. Zulässig sind nur die Positionen von 0 - n-1, wobei n die Länge des Array ist.

Zu welcher Ausgabe kommt es, wenn man die main-Methode der Klasse B aufruft?

Aufgabe 5 (10+ 10 Punkte)

Gegeben die folgende Klasse C mit der statischen Methode f. Die Methode f ist rekursiv.

```
public class C {  
  
    public static int f(String s, int n) {  
        if (n==4)  
            return 2*n;  
        if (n>14)  
            return f(s,11 - s.length());  
        return n;  
    }  
  
    public static void main (String [] args){  
        System.out.println(f("Hase", 5));  
        System.out.println(f("Bär", 18));  
        System.out.println(f("Igel", 1899));  
        System.out.println(f("Maus", -4));  
        System.out.println(f("Maus", 4));  
    }  
}
```

- A) Zu welcher Ausgabe kommt es, wenn die Klassenmethode C.main ausgeführt wird?
- B) Schreiben Sie eine neue statische Methode f, die die gleichen Funktionen realisiert wie die obige Methode f, jedoch nicht-rekursiv ist.