

# Sommersemester 2022 Präsenzübung B

Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

	A	B	C	D	E	$\Sigma$
Erreichte Punktzahl						

## Bearbeitungshinweise (abgeänderter Auszug aus den Beispielaufgaben SS 2022)

Für eine korrekte Lösung müssen nicht alle Lücken ausgefüllt werden. Die Größe der Lücken steht weiterhin nicht unbedingt in Relation zu den erreichbaren Punkten oder zur Länge des Textes, der für eine korrekte Lösung eingefüllt werden muss. Der von Ihnen eingefüllte Text muss kompilierbar sein. Verwenden Sie nur Elemente des Pakets `java.lang` der *Java SE 17 Edition*. Achten Sie im Folgenden auch auf Groß- und Kleinschreibung. Setzen Sie *nur* die in den folgenden Aufgabenstellungen angegebenen Informationen um. Sie müssen bei der Beantwortung der Präsenzübungsaufgaben keine Vorgaben zu Checkstyle-Regeln einhalten wie sonst bei der Bearbeitung der Übungsblätter.

## Aufgabe A: Kontrollfluss

(2 Punkte)

Ergänzen Sie den folgenden Quelltext der Methode `doSomething(int)` so, dass

- wenn `i` gerade ist, `i` um 1 erhöht wird
- wenn `i` gerade und größer als 3 ist, `i` um 1 verringert wird.

```
public static int doSomething(int i){  
      
      
      
      
      
      
      
      
      
    return i;  
}
```

## Aufgabe B: Schleife

(2 Punkte)

Ergänzen Sie den folgenden Quelltext der Methode `checkArray(int, int[])` so, dass die Methode überprüft, ob *ausschließlich* das Element `x` in `xs` enthalten ist. Ist dies der Fall, so wird wahr zurückgegeben, ansonsten falsch.

```
public boolean checkArray(int x, int[] xs){  
    for (   
        :   
    ){  
        if (   
        ){  
              
              
              
            return   
        ;  
        }  
    }  
    return   
    ;  
}
```

**Aufgabe C: Array****(2 Punkte)**

Ergänzen Sie den folgenden Quelltext der Methode `fillMatrix(int[][])` so, dass die Methode die übergebene Matrix an der jeweiligen Stelle mit dem Produkt des Zeilen- und Spaltenindex füllt. Die Methode gibt anschließend die übergebene Matrix wieder zurück. Sie können zusätzlich davon ausgehen, dass die Matrix bereits instanziiert ist.

Im folgenden Beispiel wird die Methode `fillMatrix(int[][])` auf eine 3x3 Matrix angewendet:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{pmatrix}$$

```
public int[][] fillMatrix(int[][] matrix) {  
    for (int i = 0;                 ; i++) {  
        for (int j = 0;                 ; j++) {  
                              
        }  
    }  
    return matrix;  
}
```


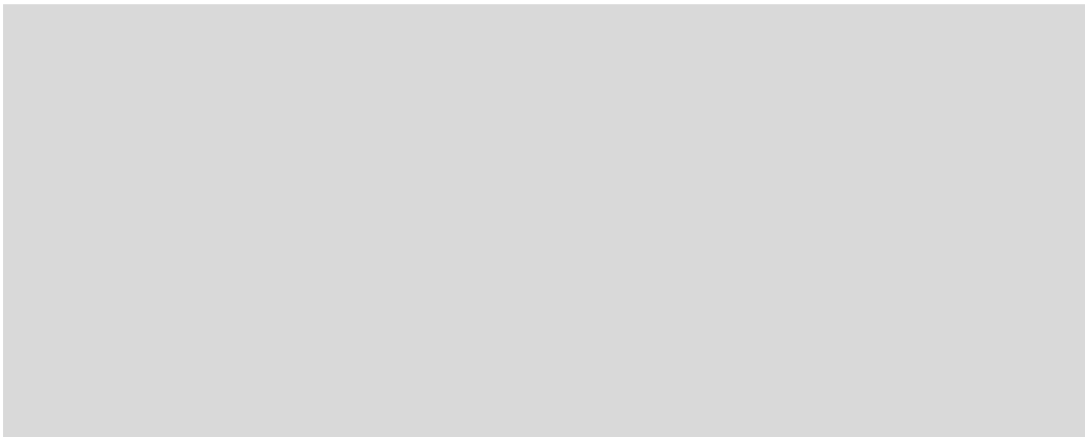
## Aufgabe D: Zugriffsmethoden

**(2 Punkte)**

Ergänzen Sie die folgende Klassendefinition um den Getter `getItem` und Setter `setItem`, die jeweils auf das Attribut `item` zugreifen.

Die `setItem()` Methode verändert `item` *nur*, wenn der übergebene String bereits instanziiert und nicht leer ist.

Hinweis: Für einen String `s` gibt die Methode `isEmpty()` mit `s.isEmpty()` wahr zurück, wenn `s` leer ist.

```
public class MagicSuperCompiler {  
  
    private String item;  
  
    public String getItem() {  
          
    }  
  
    public void setItem(String item) {  
          
    }  
}
```

**Aufgabe E: Vererbung****(2 Punkte)**

Geben Sie die Ausgabe des folgenden Quelltexts *zeilenweise* an.

```
class A {
    public void print() {System.out.println("A");}
}

class B extends A {
    public void print() {System.out.println("B");}
}

public class Main {
    public static void doPrint(A a) {
        System.out.println("A");
        a.print();
    }

    public static void doPrint(B b) {
        System.out.println("B");
        b.print();
    }

    public static void main(String[] args) {
        A x = new A();
        B y = new B();
        A z = new B();
        doPrint(x);
        doPrint(y);
        doPrint(z);
    }
}
```

Ausgabe:

