



Prof. Dr.-Ing. Anne Koziolek

Programmieren – Präsenzübung

Dauer: 20 min

Lösung: inoffiziell

Bestanden mit: 7.5P

Bemerkungen: Aufgaben von Gruppe B

Bearbeitungshinweise:

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 1 – Kontrollfluss

(2P)

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 2 – Schleifen

(2P)

Ergänzen Sie den folgenden Quelltext der Methode checkArray(int x, int[] xs) 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 (int i : xs) {
        if (i == x) {
            return true;
        }
    }
    return false;
}
```

Aufgabe 3 – Arrays

(2P)

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 instanziert ist.

Im folgenden Beispiel wird die Methode auf eine 3×3 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; [REDACTED]; i++) {
        for (int j = 0; [REDACTED]; j++) {
            [REDACTED]
        }
    }
    return matrix;
}
```

Aufgabe 4 – Zugriffsmethoden

(2P)

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 instanziert 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() {
        [REDACTED]
    }

    public void setItem(String item) {
        [REDACTED]
    }
}
```

Aufgabe 5 – Vererbung

(2P)

Geben Sie die Ausgabe des folgenden Quelltextes 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.print("A ");  
        a.print();  
    }  
  
    public static void doPrint(B b) {  
        System.out.print("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);  
    }  
}
```



Prof. Dr.-Ing. Anne Koziolek
Programmieren – Präsenzübung

Dauer: **20 min**

LÖSUNGSVORSCHLAG

Bestanden mit: **7.5P**

Bemerkungen: Aufgaben von Gruppe B

Aufgabe 1 – Kontrollfluss

(2P)

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) {  
    if (i % 2 == 0) {  
        if (i > 3) {  
            i--;  
        } else {  
            i++;  
        }  
    }  
    return i;  
}
```

Aufgabe 2 – Schleifen

(2P)

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 (entry : xs) {  
        if (entry != x) {  
            return false;  
        }  
    }  
    return true;  
}
```

Aufgabe 3 – Arrays

(2P)

Ergänzen Sie den folgenden Quelltext der Methode `fillMatrix(int[][] matrix)` 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 instanziert ist.

Im folgenden Beispiel wird die Methode auf eine 3×3 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 < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            matrix[i][j] = i * j;
        }
    }
    return matrix;
}
```

Aufgabe 4 – Zugriffsmethoden

(2P)

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 instanziert 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() {
        return this.item;
    }

    public void setItem(String item) {
        if (item != null && !item.isEmpty()) {
            this.item = item;
        }
    }
}
```

Aufgabe 5 – Vererbung

(2P)

Geben Sie die Ausgabe des folgenden Quelltextes 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.print("A ");  
        a.print();  
    }  
  
    public static void doPrint(B b) {  
        System.out.print("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);  
    }  
}
```

A A
B B
A B