

	Umbruch von Julian	WiSe21/22	21.01.2022
	Version 1 (29. Dezember 2022)		
	Dr. rer. nat. Robert Heinrich 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 **11** 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.

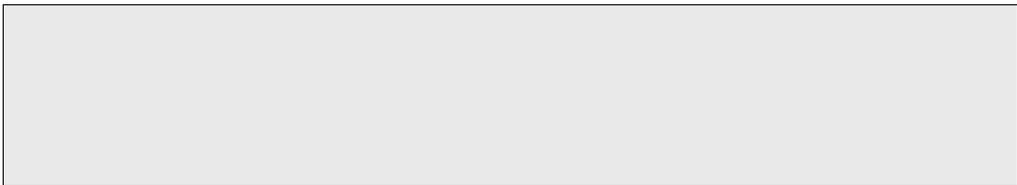
Aufgabe 1 – Schleifen

(2P)

Vervollständigen Sie folgenden Quelltext so, dass die Methode `countChars` die Anzahl der Vorkommen des Zeichens `character` in der Zeichenkette `word` zurückgibt.

Hinweis: Die Methode `String::charAt(int index)` gibt das Zeichen an der Stelle `index` einer Zeichenkette zurück.

```

public int countChars(String word, char character) {
    int count = 0;
    
    return count;
}

```

Aufgabe 2 – Operatoren

(2P)

Vervollständigen Sie folgenden Quelltext so, dass die Methode `operators`

- „FizzBuzz“ zurückgibt, wenn `i` sowohl ein Vielfaches von 5 als auch von 7 ist,
- „Fizz“ zurückgibt, wenn `i` nur ein Vielfaches von 5 ist,
- „Buzz“ zurückgibt, wenn `i` nur ein Vielfaches von 7 ist,
- und ansonsten die Zahl `i` als Zeichenkette codiert zurückgibt.

Hinweis: Der Operator `%` führt eine Division durch und gibt den Rest (Remainder) zurück.

```

public String operators(int i) {
    if (  ) {
        return "FizzBuzz";
    } else if (  ) {
        return "Fizz";
    } else if (  ) {
        return "Buzz";
    }
    return ;
}

```

Aufgabe 3 – Arrays**(2P)**

Vervollständigen Sie folgenden Quelltext so, dass die Methode copy eine Kopie des angegebenen Bereichs des Arrays original zurückgibt. Dabei soll copy die Werte aus original vom Index inklusive from bis zum Index inklusive to enthalten. Sie können davon ausgehen, dass original instanziiert ist (`original != null`), dass die Indizes from und to im Bereich des Arrays original liegen (`from >= 0` und `to < original.length`) und dass from kleiner als oder gleich to ist (`from <= to`).

```

public class Wrapper {
    public double[] copy
    public Wrapper(double[] original, int from, int to) {
        copy = ;
        for (  ) {
            
        }
        return copy;
    }
}

```

Aufgabe 4 – Enum**(1P)**

Schreiben Sie den Quelltext für eine Aufzählung (Enum) mit dem Namen Punctuation, die die Satzzeichen COLON, COMMA, PERIOD und SEMICOLON enthält.

Aufgabe 5 – While-Schleife**(1P)**

Gegeben sind die Methoden `query()` mit dem Rückgabetypp **boolean** und `command()` mit dem Rückgabetypp **void**. Vervollständigen Sie folgenden Quelltext so, dass `query` und `command` so lange aufgerufen werden, bis `query` das erste Mal **false** zurückgibt. Dabei soll `query` immer vor `command` aufgerufen werden.

```
while (  ) {  
      
}
```

Aufgabe 6 – Dynamische Bindung**(2P)**

Geben Sie für den Aufruf der `main`-Methode in folgendem Quelltext die Ausgabe auf der Konsole zeilenweise an.

```
public class Main {  
    public static void main(String[] args) {  
        Parent p = new Child();  
        p.g();  
    }  
}  
class Parent {  
    static int x = 41;  
    void f() { System.out.println(x); h(); }  
    void g() { x = 17; }  
    static void h() { System.out.println(x); }  
}  
class Child extends Parent {  
    static int x = 29;  
    void g() { f(); }  
    static void h() { System.out.println(x); }  
}
```

	Umbruch von Julian	WiSe21/22	21.01.2022
	Version 1 (29. Dezember 2022)		
	Dr. rer. nat. Robert Heinrich Programmieren – Präsenzübung		
Dauer: 20 min		LÖSUNGSVORSCHLAG	Bestanden mit: 7.5P
Bemerkungen: Aufgaben von Gruppe B			

Aufgabe 1 – Schleifen

(2P)

Vervollständigen Sie folgenden Quelltext so, dass die Methode countChars die Anzahl der Vorkommen des Zeichens character in der Zeichenkette word zurückgibt.

Hinweis: Die Methode String::charAt(int index) gibt das Zeichen an der Stelle index einer Zeichenkette zurück.

```
public int countChars(String word, char character) {
    int count = 0;
    for (char c : word) {
        if (character == c) {
            count++;
        }
    }
    return count;
}
```

Aufgabe 2 – Operatoren

(2P)

Vervollständigen Sie folgenden Quelltext so, dass die Methode operators

- „FizzBuzz“ zurückgibt, wenn i sowohl ein Vielfaches von 5 als auch von 7 ist,
- „Fizz“ zurückgibt, wenn i nur ein Vielfaches von 5 ist,
- „Buzz“ zurückgibt, wenn i nur ein Vielfaches von 7 ist,
- und ansonsten die Zahl i als Zeichenkette codiert zurückgibt.

Hinweis: Der Operator % führt eine Division durch und gibt den Rest (Remainder) zurück.

```
public String operators(int i) {
    if (i % 5 == 0 && i % 7 == 0) {
        return "FizzBuzz";
    } else if (i % 5 == 0) {
        return "Fizz";
    } else if (i % 7 == 0) {
        return "Buzz";
    }
    return "" + i;
}
```

Aufgabe 3 – Arrays**(2P)**

Vervollständigen Sie folgenden Quelltext so, dass die Methode `copy` eine Kopie des angegebenen Bereichs des Arrays `original` zurückgibt. Dabei soll `copy` die Werte aus `original` vom Index inklusive `from` bis zum Index inklusive `to` enthalten. Sie können davon ausgehen, dass `original` instanziiert ist (`original != null`), dass die Indizes `from` und `to` im Bereich des Arrays `original` liegen (`from >= 0` und `to < original.length`) und dass `from` kleiner als oder gleich `to` ist (`from <= to`).

```
public class Wrapper {
    public double[] copy
    public Wrapper(double[] original, int from, int to) {
        copy = new double[to - from + 1] ;
        for (int i = 0; i < copy.length; i++) {
            copy[i] = original[i + from];
        }
        return copy;
    }
}
```

Aufgabe 4 – Enum**(1P)**

Schreiben Sie den Quelltext für eine Aufzählung (Enum) mit dem Namen `Punctuation`, die die Satzzeichen `COLON`, `COMMA`, `PERIOD` und `SEMICOLON` enthält.

```
public enum Punctuation {
    COLON, COMMA, PERIOD, SEMICOLON
}
```

Aufgabe 5 – While-Schleife**(1P)**

Gegeben sind die Methoden `query()` mit dem Rückgabetypp `boolean` und `command()` mit dem Rückgabetypp `void`. Vervollständigen Sie folgenden Quelltext so, dass `query` und `command` solange aufgerufen werden, bis `query` das erste Mal `false` zurückgibt. Dabei soll `query` immer vor `command` aufgerufen werden.

```
while (query()) {
    command();
}
```

Aufgabe 6 – Dynamische Bindung**(2P)**

Geben Sie für den Aufruf der main-Methode in folgendem Quelltext die Ausgabe auf der Konsole zeilenweise an.

```
public class Main {
    public static void main(String[] args) {
        Parent p = new Child();
        p.g();
    }
}
class Parent {
    static int x = 41;
    void f() { System.out.println(x); h(); }
    void g() { x = 17; }
    static void h() { System.out.println(x); }
}
class Child extends Parent {
    static int x = 29;
    void g() { f(); }
    static void h() { System.out.println(x); }
}
```

41

41