

Übung 2, Lösungsvorschlag



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Aufgabe 1 (Wissensfrage)

Beantworten Sie die folgenden Fragen:

- a) Welche drei Typen von Operatoren haben Sie kennengelernt? Was zeichnet diese aus?
- b) Was ist bei der Division (arithmetische Operation) zweier Ganzzahlen (int) zu beachten? Welchen Unterschied spielt hierbei der Datentyp der Variablen, in der das Ergebnis der Division gespeichert wird?
- c) Welche Booleschen Operatoren gibt es in Java? Erklären Sie auch deren Funktionsweise.
- d) Was ist der Unterschied zwischen Booleschen Operatoren und Vergleichsoperatoren? Was haben sie gemeinsam?

Lösung Aufgabe 1

- a) **1. Arithmetische Operatoren:** Verarbeitung von numerischen Daten durch Addition, Subtraktion, Multiplikation, Division, Modulo (+, -, *, /, %).
2. Vergleichsoperatoren: Vergleich zweier Variablen (<, <=, >, >=, ==, !=). Das Ergebnis ist ein Wahrheitswert (Boolean).
3. Boolesche Operatoren: Logische Verarbeitung boolescher Variablen (!, &&, ||). Das Ergebnis ist wiederum ein Wahrheitswert.
- b) Werden zwei Variablen vom Datentyp int dividiert, werden mögliche Nachkommastellen ignoriert. Das Ergebnis entspricht somit immer dem **abgerundeten** Wert des Quotienten. Dies ist unabhängig vom Datentyp der Variablen, welcher das Ergebnis zugewiesen wird. **Hinweis:** Es ist in Java jedoch nicht möglich das Ergebnis einer Division von zwei Variablen vom Datentyp double einer Variablen vom Datentyp int zuzuweisen. In diesem Fall kommt es zu einem „Type Mismatch Error“.
- c) Negation: !a Aus wahr wird falsch und aus falsch wird wahr.
Und: a && b Beide Aussagen müssen wahr sein, damit die Gesamtaussage wahr wird
Oder: a || b Eine der beiden Aussagen muss wahr sein, damit die Gesamtaussage wahr wird
- d) Beide Typen von Operatoren liefern boolesche Werte, allerdings erwarten Boolesche Operatoren boolesche Werte als Operanden. Somit werden Boolesche Operatoren für die Implementierung von „Logik“ genutzt und verknüpfen mehrere Wahrheitswerte zu einem neuen. Währenddessen arbeiten Vergleichsoperatoren insbesondere mit numerischen Datentypen und ergeben stets einen Wahrheitswert (Booleschen Wert). Sie werden später noch lernen, dass Vergleiche auch zwischen anderen Objekten möglich sind, sofern die Vergleichsoperatoren dort definiert sind.

Beide Arten von Operatoren werden innerhalb von Kontrollstrukturen benötigt, um den Ablauf eines Programm zu steuern.

Aufgabe 2 (Operatoren auswerten)

Werten Sie folgende Ausdrücke aus und geben Sie jeweils den Wert der Variablen `x` an. Werten Sie hierzu die einzelnen Booleschen oder arithmetischen Ausdrücke Schritt für Schritt aus.

Teilaufgabe	Ausdruck	Wert von <code>x</code>
A	<code>int a = 5; boolean x = (a>3);</code>	
B	<code>int a = 9; boolean x = !(a>5);</code>	
C	<code>boolean x = true false;</code>	
D	<code>int a = 5; boolean x = (a>=2 && a<=4) (a-4==1);</code>	
E	<code>int x = 5 * (2 - 4);</code>	
F	<code>int a = 4; int x = 8 * 2 - 4 * a;</code>	
G	<code>int x = 10 - 5 / 2;</code>	

Lösung Aufgabe 2

Teilaufgabe	Ausdruck	Wert von <code>x</code>
A	<code>int a = 5; boolean x = (a>3);</code>	true
B	<code>int a = 9; boolean x = !(a>5);</code>	false
C	<code>boolean x = true false;</code>	true
D	<code>int a = 5; boolean x = (a>=2 && a<=4) (a-4==1);</code>	true
E	<code>int x = 5 * (2 - 4);</code>	-10
F	<code>int a = 4; int x = 8 * 2 - 4 * a;</code>	0
G	<code>int x = 10 - 5 / 2;</code>	8

Aufgabe 3 (Erstes Mal Java Programmieren)

In dieser Aufgabe werden Sie Ihr erstes Java-Programm schreiben. Erstellen Sie hierzu in Eclipse ein Java-Projekt (z.B. „GdP 2019 Uebung 2“). Im „Package Explorer“ auf der linken Seite können Sie nun das Projekt öffnen. Erstellen Sie innerhalb des Ordners „src“ (Abk: source) eine neue Klasse mit dem Namen „Aufgabe3a“ bzw. „Aufgabe3b“. Aktivieren Sie während des Erstellens der Klasse, dass sie eine „Main-Methode“ anlegen lassen (Abbildung 1).

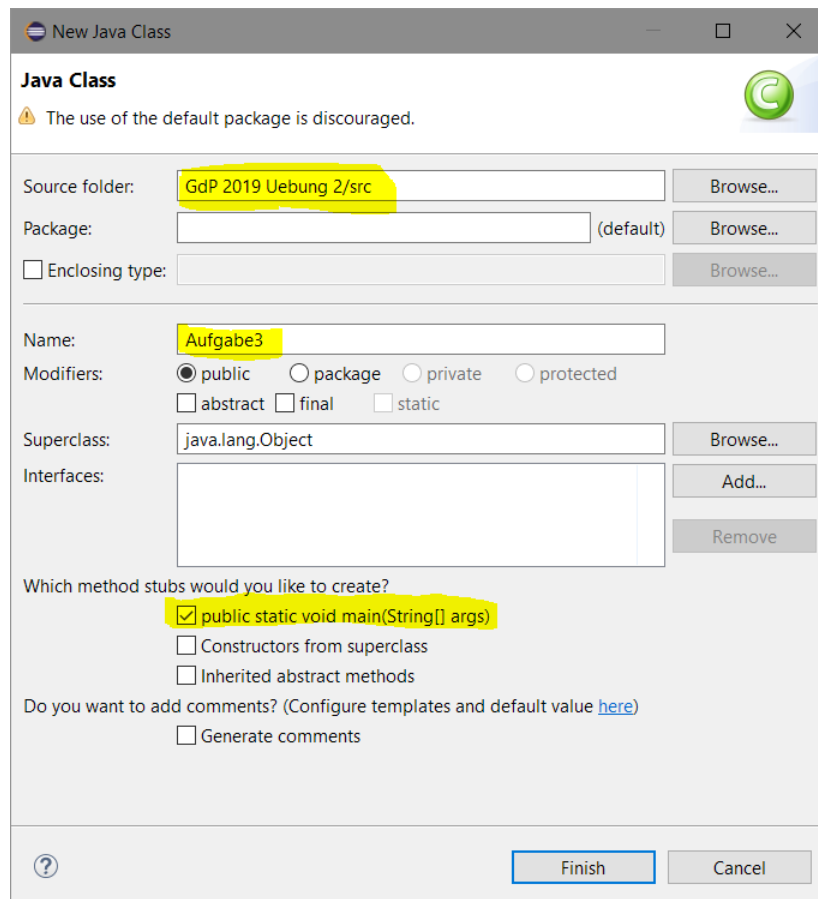


Abbildung 1: Erstellen einer neuen Java Klasse

Hinweis: Nachdem Sie in der Vorlesung die Konzepte von Methoden und Klassen kennengelernt haben, sollten Sie auch selbstständig (ohne Hilfe von Eclipse) in der Lage sein, eine Klasse inklusive Main-Methode vollständig zu deklarieren.

Der so erzeugte Code sollte in etwa wie folgt aussehen:

```
1 public class Aufgabe3a {
2
3     public static void main(String[] args) {
4         // TODO Auto-generated method stub
5     }
6
7 }
8 }
```

Schreiben Sie im Folgenden zwei Java-Programme, die die unten gegebenen Algorithmen implementieren. Verwenden Sie zur Ausgabe von Zeichenketten (z.B. „Gerade Zahl:“) bzw. Variablen (z.B. x) folgende Aufrufe:

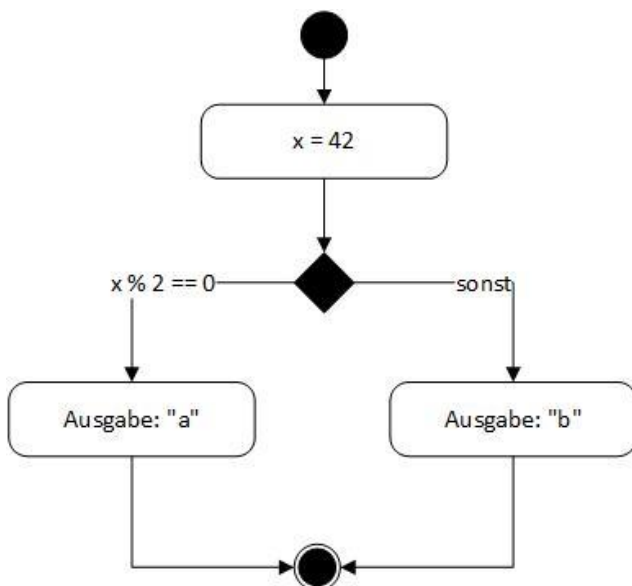
```
System.out.println(„Gerade Zahl:“);
```

```
System.out.println(x);
```

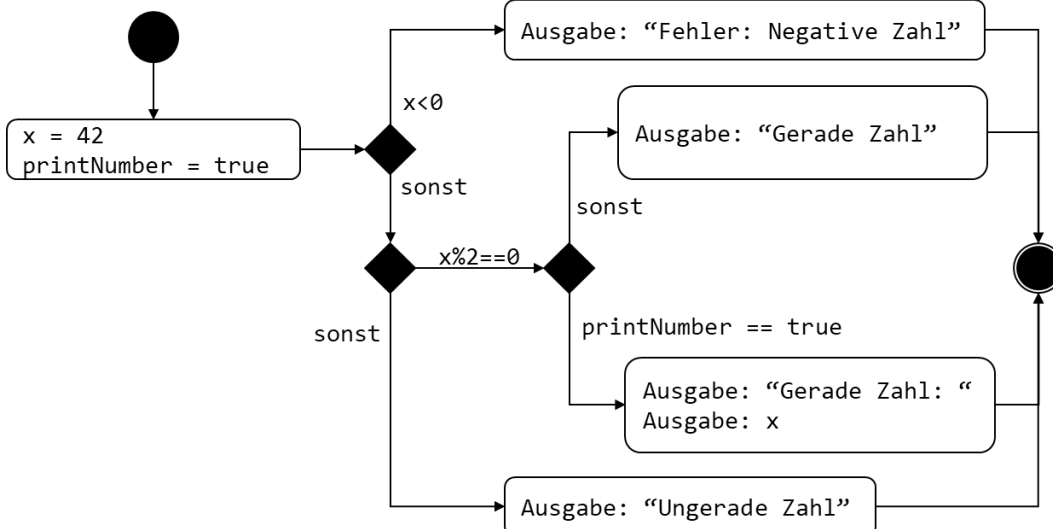
Sie benötigen für jedes Programm eine eigene Klasse mit einem eigenen Namen und implementieren die Algorithmen direkt unterhalb des Kommentars `// TODO Auto-generated method stub`

Führen Sie die Programme anschließend in Eclipse aus und prüfen Sie, dass die Ausgabe auf der Konsole korrekt erscheint.

a)



b)



Lösung Aufgabe 3

a)

```
1 public class Aufgabe3a {
2     public static void main(String[] args) {
3         int x = 42;
4         if (x % 2 == 0) {
5             System.out.println("a");
6         } else {
7             System.out.println("b");
8         }
9     }
10 }
```

b)

```
1 public class Aufgabe3b {
2     public static void main(String[] args) {
3         int x = 42;
4         boolean printNumber = true;
5         if (x < 0) {
6             System.out.println("Fehler: Negative Zahl");
7         } else {
8             if (x % 2 == 0) {
9                 if (printNumber == true) {
10                    System.out.println("Gerade Zahl: ");
11                    System.out.println(x);
12                } else {
13                    System.out.println("Gerade Zahl");
14                }
15            } else {
16                System.out.println("Ungerade Zahl");
17            }
18        }
19    }
20 }
```

Hinweis: In Zeile 9 kann anstatt des Vergleichs mit `==true` auch die Variable `printNumber`, die bereits den Datentyp Boolean speichert, direkt als Bedingung der if-Abfrage genutzt werden.

Aufgabe 4 (Operatoren: Switch)

Für die folgende Aufgabe ist es notwendig, dass Sie in Eclipse eine neue Klasse anlegen (z.B. „Aufgabe4“). Sie können hierzu dasselbe Projekt verwenden, das Sie in der vorherigen Aufgabe erstellt haben. Denken Sie auch hier wieder daran, eine „Main-Methode“ anlegen zu lassen.

Gegeben ist der folgende Java Programm-Code:

```
1 public class Aufgabe4 {
2     public static void main(String[] args) {
3         int note = 1;
4         if (note == 1) {
5             System.out.println("Sehr gut");
6         } else if (note == 2) {
7             System.out.println("Gut");
8         } else if (note == 3) {
9             System.out.println("Befriedigend");
10        } else if (note == 4) {
11            System.out.println("Ausreichend");
12        } else if (note == 5) {
13            System.out.println("Mangelhaft");
14        } else if (note == 6) {
15            System.out.println("Unzureichend");
16        } else {
17            System.out.println("Fehler: Die Eingabe ist keine Note!");
18        }
19    }
20 }
```

- a) Überführen Sie die `if`-Anweisung in eine `switch`-Anweisung. Nutzen Sie hierzu zuerst Stift und Papier.
- b) Implementieren Sie nun das Programm mit der `switch`-Anweisung in Eclipse und stellen Sie sicher, dass die Ausgabe auf der Konsole korrekt funktioniert. Verändern Sie hierzu in Zeile 3 den Wert der Variablen `note`.
Z.B. sollte die Ausgabe für `int note = 4` lauten: „Ausreichend“

Lösung Aufgabe 4

```
1  public class Aufgabe4 {
2      public static void main(String[] args) {
3          int note = 1;
4          switch (note) {
5              case 1:
6                  System.out.println("Sehr gut");
7                  break;
8              case 2:
9                  System.out.println("Gut");
10                 break;
11             case 3:
12                 System.out.println("Befriedigend");
13                 break;
14             case 4:
15                 System.out.println("Ausreichend");
16                 break;
17             case 5:
18                 System.out.println("Mangelhaft");
19                 break;
20             case 6:
21                 System.out.println("Unzureichend");
22                 break;
23             default:
24                 System.out.println("Fehler: Die Eingabe ist keine Note!");
25                 break;
26         }
27     }
28 }
29 }
```