

Programmierung II

Wintersemester 2021/22

Prof. Dr.-Ing. Marco Block-Berlitz

Prof. Dr.-Ing. Dietrich Kammer

Dipl.-Ing. Jan Roeper

Getting Started with Java

Online Compiler

Ohne Installation direkt loslegen

<https://repl.it/languages/java10>

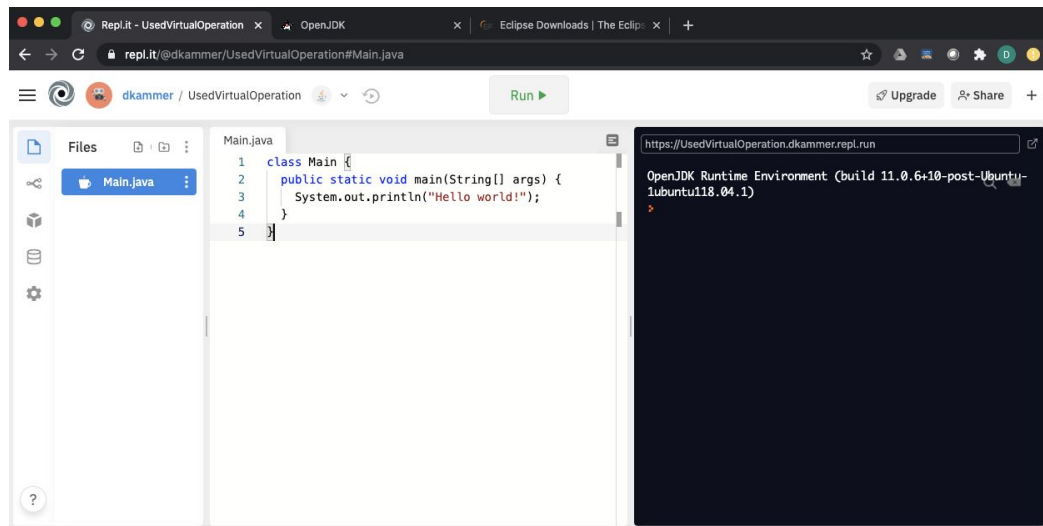
Texteditor und Kommandozeile

Herunterladen des [OpenJDK](#).

Verwenden eines beliebigen Text- oder Code-Editors, bspw. [Visual Studio Code](#).

Integrierte Entwicklungsumgebung (IDE)

Beispielsweise [Eclipse](#) oder [IntelliJ](#).



Zahlensysteme

A01

Lösen Sie die folgenden Aufgaben im Hexadezimalsystem und wandeln Sie anschließend das Ergebnis in das Dezimalsystem um:

- (a) $2F + A2$ (b) $07 + FF$ (c) $FE - A4$

Alternativ können Sie die Darstellungen auch zunächst in das Dezimalsystem überführen, dort ausrechnen und das Ergebnis wieder in das Hexadezimalsystem überführen.

A02

Wandeln Sie die folgenden Binärdarstellungen in Dezimalzahlen um:

- (a) 10011011 (b) 01100100 (c) 11001101

Es handelt sich dabei um die Zweierkomplement-Darstellung mit 8 Bit.

Arbeiten mit Variablen

A03

Welche der folgenden Variablennamen sind gültig bzw. ungültig?

Norbert, \$eins, _abc123, #hallihallo, erne\$sto, const, int,
1a, k_l-e_i-n, %nummer, Class, klasse, !wahr, final, blablubs

A04

Erläutern Sie kurz den Unterschied zwischen

`a=b` und `a==b`

A05

Welchen Datentyp und welche Bezeichnung würden Sie für die folgenden Informationen vergeben:

- | | |
|-----------------------------------|---|
| (a) Alter einer Person | (d) Geldbetrag Ihres Konto |
| (b) Anfangsbuchstabe | (e) Fläche eines Fußballfeldes (cm ²) |
| (c) Position eines Lichtschalters | (f) Kreiszahl π |

Boole'sche Algebra

A06

Geben Sie eine Variable *c* an, die die Funktion aus folgender Wertetabelle berechnet:

B1	B2	(B1 UND B2) ODER (NICHT B2)
0	0	1
0	1	0
1	0	1
1	1	1

Deklarieren Sie notwendige Variablen und weisen ihnen gegebenenfalls Werte zu, damit eine sinnvolle Auswertung des Ausdrucks möglich ist.

A07

Werten Sie die folgenden Programmzeilen aus und geben Sie die Werte von *c*, *d*, *e*, *f* und *g* an:

Hinweis: Der Operator \wedge steht für XOR

```
boolean a = true, b = false, c, d, e, f, g;  
c = a ^ b;  
d = !a || b;  
e = (d && !c) || !a;  
f = ((d == e) || (d != e)) == true;  
g = 5==7;
```

Datentypen und Schleifen

A08

Warum wird zwischen der impliziten und expliziten Typumwandlung unterschieden? Was könnte schiefgehen?

A09

Arbeiten Sie die Besonderheiten der unterschiedlichen Schleifenvarianten in Java heraus und formulieren Sie einen Ratgeber mit Beispielen, wann welche Variante am besten geeignet ist.

Einfache Java-Programme

A10

Gehen Sie die einzelnen Schritte aus Abschnitt 2.3.2 durch und bringen Sie das Programm ProgrammEins zum Laufen.

```
public class ProgramEins {  
    public static void main(String[] args) {  
        System.out.println("Endlich ist es soweit! Mein erstes Programm läuft...");  
    }  
}
```

A11

Geben Sie ein Programm in Java an, das folgende Formeln in jeweils separaten Funktionen berechnet.

(a) $f_1(x) = x$

(b) $f_2(x) = x^2/2 + 17 \cdot 2$

(c) $f_3(x) = ((x-1)^3 - 14)/2$

Programmieren mit Schleifen

A12

Schreiben Sie ein Programm, das für $i = 1, 2, \dots, 20$ die Fakultätsfunktion berechnet und die Funktionswerte zeilenweise ausgibt. Die Fakultätsfunktion ist wie folgt definiert:

$$fakultaet(n) = 1 \cdot 2 \cdot \dots \cdot n = \prod_{i=1}^n i = n!$$

Beispiel: $fakultaet(4) = 1 \cdot 2 \cdot 3 \cdot 4 = 24$

A13

Geben Sie für die folgenden Summen entsprechende for-Schleifen an:

$$(a) \sum_{i=0}^{28} (i-1)^2 \quad (b) \sum_{i=1}^{100} \frac{i \cdot (i+1)}{2} \quad (c) \sum_{i=1}^{25} \frac{(i+1)}{i}$$

Programmieren mit Schleifen

A14

Überführen Sie die folgenden Schleifen in for- bzw. while-Schleifen.

```
for (int x=7; x<12; x++)  
    <Anweisung>
```

```
for (int y=0, x=10; x>y; y++, x--)  
    <Anweisung>
```

```
int a = 1024;  
while (a>2) {  
    <Anweisung>  
    a=a/2;  
}
```

Komplexaufgabe

A15

Schreiben Sie ein Java-Programm, dass nach Eingabe von zwei Ganzzahlen auf der Konsole diese multipliziert und dabei überprüft, ob der Zahlenbereich des Datentyps Integer über- oder unterschritten wird. In diesem Fall soll eine Warnung auf der Konsole zusätzlich zum Berechnungsergebnis ausgegeben werden.

Hinweis: Rechts sehen Sie, wie Nutzereingaben mit der Klasse Scanner vorgenommen und in Integer-Werte konvertiert werden können.

```
import java.util.Scanner;

public class Overflow {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter first operand: ");
        String op1Input = sc.nextLine();
        int op1 = Integer.parseInt(op1Input);

        // TODO: Eingabe zweiter Operand
        // und Multiplikation mit Überprüfung auf
        // Einhaltung des Zahlenbereichs!
    }
}
```