

Übung 1

Aufgabe 1: Hello World

1. Download der IDE **IntelliJ IDEA CE** von JetBrains
2. Schreiben Sie ein Programm **Hello**, welches den Text **"Hello World"** ausgibt.
(Google: java hello world)
3. Führen Sie dieses Programm via IDE aus
4. Kompilieren Sie dieses Programm von der Konsole und führen Sie es dort aus.

Verständnisziele:

- Was ist eine IDE ist und wie wird diese benutzt?
- Wie ist der Grundaufbau eines jeden Java-Programmes?
 - Wie hängen Dateiname und **"class <name>"** zusammen?
- Was macht der Befehl **javac** und wie wird er benutzt?
- Was macht der Befehl **java** und wie wird er benutzt?

Aufgabe 2: Testen von Hello World mit Kommandozeilen-Argumenten

1. Schreiben Sie ein Programm **HelloArgs**, welches ein Kommandozeilen-Argument verarbeiten kann und entsprechend **"Hello <Ihr Name>"** ausgibt.
2. Führen Sie dieses Programm via IDE und CMD* aus
3. Geben Sie Ihren Vor- und Nachnamen aus.
4. Schreiben Sie einen Test, welches unser Programm aufruft:
 1. Mit einem Argument
 2. Mit mehreren Argumenten
 3. Mit keinem Argument
 4. Mit einem leeren Argument

* CMD = Kommandozeile, Terminal, Command Line Interpreter

Tipps

- Mit **"System.out.println(Arrays.toString(args))"** können Sie alle Argumente ausgeben.
- IntelliJ bietet mit Strg+Shift+A eine Befehlssuche, Tests lassen sich damit erzeugen, wenn nach "create Test" gesucht wird.

Verständnisziele

- Was sind Kommandozeilen-Argument?
- Wie unterscheiden sich Argumente mit und ohne Anführungszeichen?
- Wieso ist ein Test sinnvoll?
 - Wieso sollte ein Test zunächst Fehler anzeigen?
- Was ist ein Debugger und wie hilft dieser bei der Fehlerfindung?

Aufgabe 3: CMD-Taschenrechner

1. Schreiben Sie ein Programm **Calc**, welches zwei ganze Zahlen akzeptiert und diese verrechnet. Das Programm soll mit **"java Calc 1 - 2"** aufrufbar sein.
2. Erweitern Sie ihr Programm um die anderen Rechenoperationen: +, *, / und %
3. Extrahieren Sie die Rechenoperationen in eine eigene Funktion, welche auf Ein- und Ausgabe getestet werden soll (EVA).
Die Funktion soll einen String zurück geben mit dem Ergebnis oder einer Fehlermeldung, z.B. Operator nicht gefunden.
4. Testen Sie auf Division durch 0.

Verständnisziele

- Wie hilft die IDE beim Beheben von Fehlern?
- Wie werden Strings in ints umgewandelt?
 - Wozu ist Casten notwendig?
- Wie wird eine eigene Funktion deklariert und aufgerufen?
 - Was ist Refactoring?
- Wie werden mehrere boolesche Ausdrücke verrechnet und ausgewertet?

Aufgabe 4:

Schreiben Sie ein Programm **Counter**, welches abwechselnd . und ! ausgibt bis zu einem festen Limit.