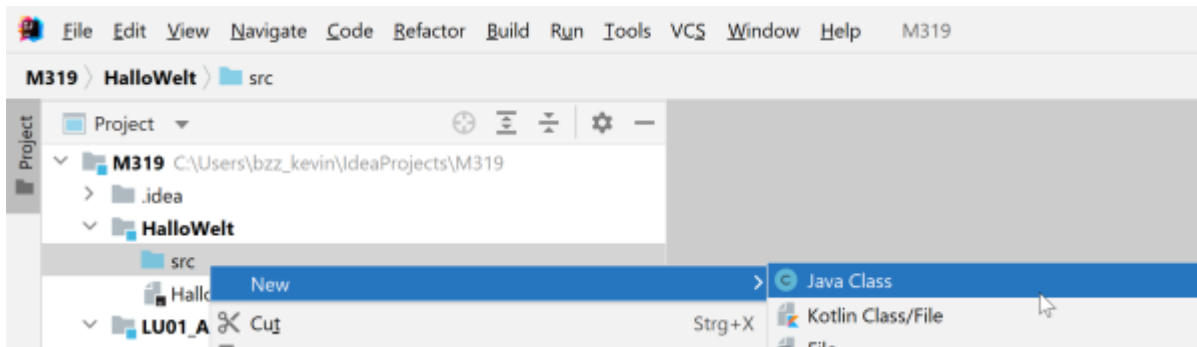


# LU01b - IntelliJ - Hallo Welt!

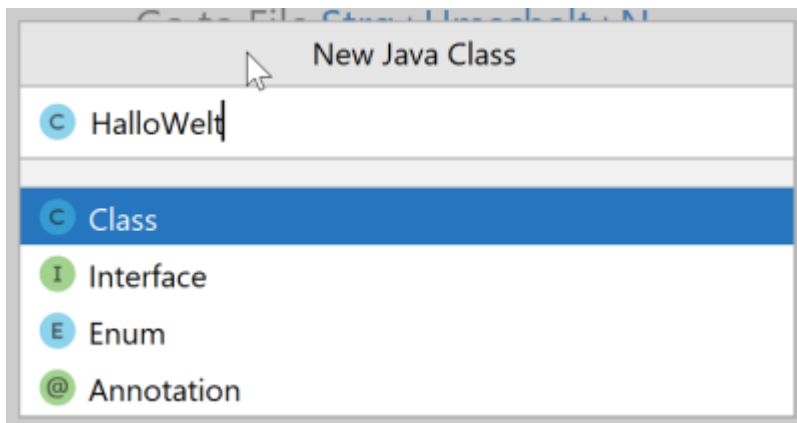
Sie erstellen nun in IntelliJ ihr erstes Java Programm.

## Schritt 1:

Fügen Sie eine neue Java-Klasse in den `src` Ordner im passenden Modul ein:



Benennen Sie die Klasse Beispielsweise `HalloWelt`

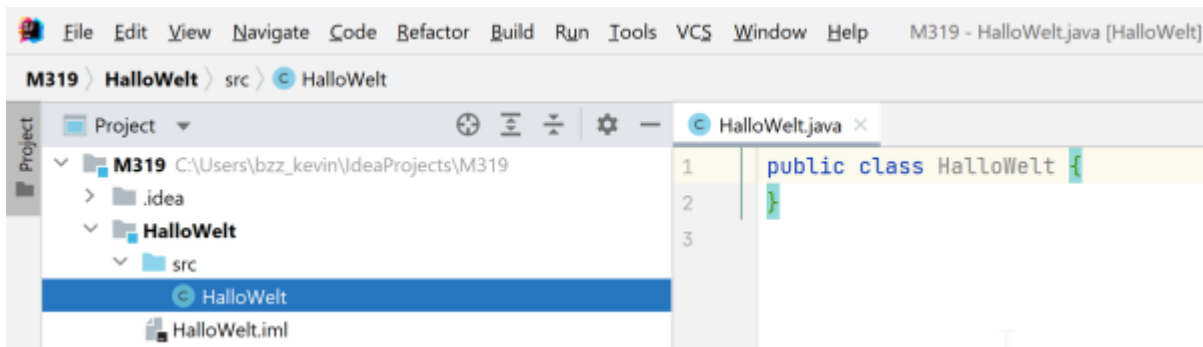


Für Klassenbenennungen gibt es Konventionen:



Klassennamen sollten Nomen sein, in gemischter Großschreibung, wobei der erste Buchstabe jedes inneren Wortes groß geschrieben werden sollte (CamelCase). Versuchen Sie, Ihre Klassennamen einfach und beschreibend zu halten. Verwenden Sie ganze Wörter - vermeiden Sie Akronyme und Abkürzungen (es sei denn, die Abkürzung ist viel verbreiteter als die Langform, z. B. URL oder HTML).

Sie haben nun Ihr `HalloWelt.java` im `src` Ordner des Moduls.



## Schritt 2:

In jeder Programmiersprache gibt es verschiedene Vorschriften zum Aufbau des Sourcecodes.

Für Java müssen Sie die folgenden Regeln beachten:

- Das Programm wird mit der Zeile `public class KLASSENNAME` eingeleitet.
- Der Klassenname muss mit dem Dateinamen (ohne die Endung `.java`) übereinstimmen. Zum Beispiel:
  - Datei: „Universe.java“ / Klassenname: „Universe“
  - Datei: „HalloWelt.java“ / Klassenname: „HalloWelt“
- Das ganze Programm wird mit geschweiften Klammern eingefasst.
- Jedes Programm muss eine Methode `public static void main(String[] args)` haben. Beim Ausführen des Programmes startet das Betriebssystem diese Methode.
- Methoden (z.B. `main`) werden mit geschweiften Klammern eingefasst.

Wir verwenden für unsere Programme im Modul 319 immer eine Programmiervorlage. Kopieren Sie diese nun in ihr File „HalloWelt.java“

```
/**  
 * Prints "Hallo Welt!"  
 *  
 * @author Ibid umm  
 * @since yyyy-mm-dd  
 * @version 1.0  
 */  
public class HalloWelt{  
  
    /**  
     * starts the execution  
     *  
     * @param args command line arguments  
     *  
     */  
    public static void main(String[] args) {  
        HalloWelt program = new HalloWelt();  
        program.run();  
    }  
}
```

```
/**
 * runs the program
 */
private void run() {

    // TODO realize the program logic --

}
```

## Schritt 3:

Vielleicht ist das Programmieren noch völlig neu für Sie oder haben Sie schon ein wenig Erfahrung? In beiden Fällen lohnt es sich, zuerst zu denken und danach zu tippen.

Um unsere Befehle einzufügen, müssen wir jeweils drei Fragen klären:

Was soll der Befehl machen? Wo gehört der Befehl hin? Wie schreibt man diesen Befehl?

### Auftrag

Das Programm soll den Text „Hallo Welt“ ausgeben.

### Was?

Das „Was“ ergibt sich hier aus dem Auftrag: Der neue Befehl soll einen Text ausgeben.

### Wo?

Wenn Sie das Programmgerüst betrachten, so finden Sie einen Kommentar

```
// TODO realize the program logic
```

Ihre Befehle folgen direkt auf diesen Kommentar. Somit wäre auch das Wo geklärt.

Fügen Sie an der richtigen Stelle einen neuen Kommentar in den Sourcecode ein:

```
// TODO realize the program logic
// TODO Ausgabe: "Hallo Welt"
```

### Wie?

Sie wissen nicht, wie man eine Ausgabe in Java macht? Dann lesen Sie einfach nach, wie das geht.

Suchen Sie im Internet (Java Text ausgeben) oder einem Fachbuch den richtigen Befehl.

Schon nach kurzer Zeit finden Sie ein Beispiel für eine Ausgabe:

```
System.out.println("Dieser Text wird angezeigt");
```

Bevor Sie das Beispiel einfach kopieren, sollten Sie sich diesen Befehl kurz anschauen:

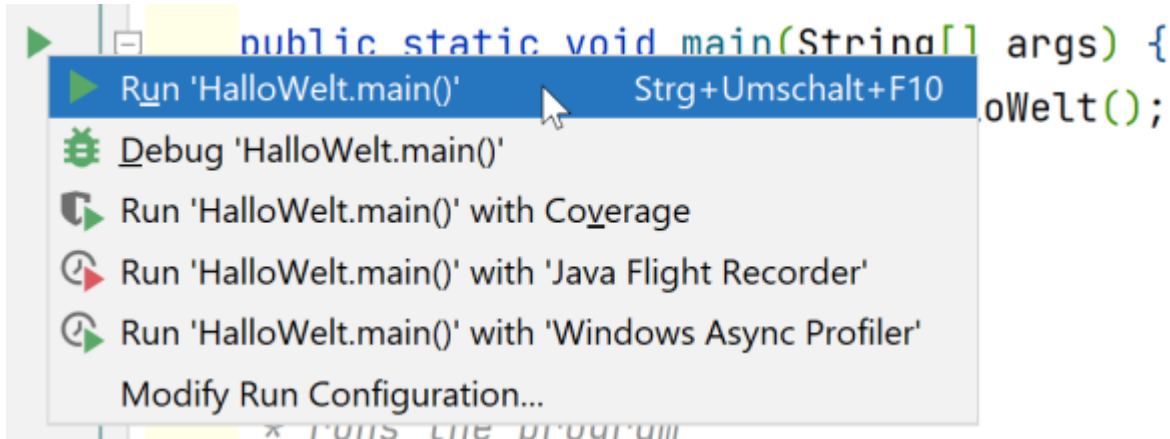
- Der Befehl für eine Ausgabe lautet: `System.out.println(...)`.
- Der auszugebende Text wird zwischen Anführungs-/Schlusszeichen (engl. double quotes) geschrieben.
- Der Befehl wird mit einem Strichpunkt (engl. semi colon) beendet.

Nachdem Sie den Befehl (Syntax) recherchiert haben, ersetzen Sie Ihren Kommentar mit dem Befehl.

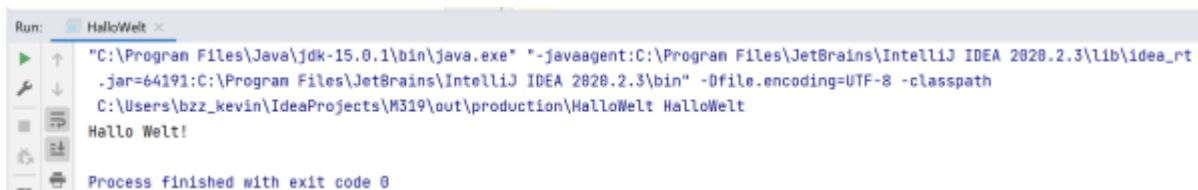
```
HalloWelt.java x
1  /**
2      * Gibt "Hallo Welt!" aus.
3      *
4      * @author Kevin Maurizi
5      * @since 29.02.2040
6      * @version 1.0
7  */
8  public class HalloWelt{
9
10     /**
11      * starts the execution
12      *
13      * @param args command line arguments
14      *
15     */
16     public static void main(String[] args) {
17         HalloWelt program = new HalloWelt();
18         program.run();
19     }
20
21     /**
22      * runs the program
23     */
24     private void run() {
25
26         // TODO realize the program logic --
27         System.out.println("Hallo Welt!");
28     }
29 }
```

## Schritt 4

Als nächstes müssen wir unsere Programm kompilieren und ausführen. Ihre IDE macht das für Sie, die einzige Aufgabe für Sie ist, dass Sie der IDE sagen müssen welches Programm ausgeführt werden soll. Drücken Sie dazu auf den grünen Pfeil neben der main-Methode.



IntelliJ kompiliert nun Ihr File „HaloWelt.java“ und führt es anschliessend aus:



Gratulation, Sie haben nun ihr erstes Java-Programm in IntelliJ geschrieben und gestartet.



© Kevin Maurizi, Marcel Suter

[IntelliJ, IDE](#)

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu01/lu01b-helloworld>

Last update: **2025/06/23 07:45**

