



ICT Berufsbildung
Formation professionnelle
Formazione professionale

Programmieren mit Java

Einführung in JetBrains IntelliJ IDEA



Inhalt

1	Auftrag.....	2
2	Ziel	2
3	Jetbrains IntelliJ Idea - Entwicklungsumgebung (IDE).....	2
3.1	Einführung	2
3.2	Installation	3
3.2.1	JDK – Java Development Kit.....	3
3.2.2	JAVA_HOME Umgebungsvariable	5
3.2.3	Deine Entwicklungsumgebung - IntelliJ IDEA.....	8
3.3	Erster Start und Konfiguration.....	10
3.4	Die Bestandteile der Entwicklungsumgebung	12
3.4.1	Aufbau IntelliJ.....	12
3.4.2	IntelliJ Code-Editor	13
4	"Hello World" - Dein erstes Projekt.....	14



1 Auftrag

Dein Auftrag ist es, die folgenden Kapitel durchzulesen. Du wirst erfahren, wie Du IntelliJ (<https://www.jetbrains.com/idea/>), die Entwicklungsumgebung für Java, auf Deinem Rechner installieren kannst und erhältst eine kurze Einführung in die wichtigsten Funktionen, die Dir IntelliJ bietet.

2 Ziel

Du lernst das mächtige und umfangreiche Entwicklungswerkzeug *IntelliJ Idea* von JetBrains kennen und erstellst dein erstes Java-Programm (HelloWorld-Applikation).

3 JetBrains IntelliJ Idea - Entwicklungsumgebung (IDE)

IntelliJ ist die optimale Entwicklungsumgebung für die Programmierung mit Java. Die Möglichkeiten, die dieses Tool bietet, sind derart umfangreich und mächtig, dass ein einziges Kapitel nicht ausreicht, um sie vollständig zu beschreiben. Daher soll es hier nur darum gehen, die wichtigsten Elemente und ihre Verwendung hervorzuheben. Weiterführende Informationen zu IntelliJ findest Du zuhauf im Internet, z.B. hier: <https://www.jetbrains.com/idea/documentation/>



Hast Du gewusst?

JetBrains entwickelt neben [IntelliJ](#) auch Entwicklungsumgebungen für andere Plattformen, wie z.B. [Webstorm](#) (Subset von IntelliJ), [PhpStorm](#) für die Entwicklung mit PHP oder [AppCode](#) für die Entwicklung von iOS Apps. Sogar eigene Programmiersprachen wurden von JetBrains entwickelt, z.B. [Kotlin](#). Ein weiteres Highlight ist die Buildumgebung [TeamCity](#), wo Du dein Programmiercode von einer Serverumgebung automatisiert bauen und testen lassen kannst.

3.1 Einführung

Die Anforderungen an moderne Applikationen steigen, ebenso die Möglichkeiten und auch die Fehlerquellen. Eine moderne IDE (Integrated Development Environment = Entwicklungsumgebung) hilft nicht nur bei der Codeeingabe, sondern stellt dem Entwickler eine Vielzahl von Werkzeugen und Möglichkeiten zur Verfügung, die ihm das (Entwickler-) Leben stark erleichtern.

IntelliJ ist aus zahlreichen Gründen die beste Wahl für die Anwendungsentwicklung unter Java:

- ✓ Intelligente Coding Unterstützung, wie z.B. Komplettierung deiner Eingabe, Navigation innerhalb deines Codes, Unterstützung bei der Umstrukturierung („Refactoring“) deines Codes sowie sofortige Fehleranalyse.
- ✓ Direkt integrierte Web- und Java Entwicklungstools wie [Maven](#), [Gradle](#) und [Grunt](#) oder bietet die Verwaltung von Datenbanken wie z.B. [Oracle](#), [SQL Server](#), [PostgreSQL](#) oder [MySQL](#) an. Zusätzlich wird das Software Configuration Management Tool [git](#) direkt integriert.



- ✓ Falls Du einmal Android Apps entwickeln möchtest, bist Du mit IntelliJ genau richtig. IntelliJ bietet einen eigenen Android UI-Designer an.
- ✓ Oder auch Web-Anwendungen kannst Du mit IntelliJ einfach entwickeln. HTML, Javascript und CSS werden komplett unterstützt.
- ✓ Zusätzlich verfügt IntelliJ über ein riesiges Plugin-System, mit welchem man seine Entwicklungsumgebung nach den eigenen Wünschen konfigurieren kann.
- ✓ Die IntelliJ Community Edition ist kostenlos.
- ✓ Exceptions (Fehler) werden detailliert angezeigt und der Debugger unterstützt dich während der Fehlersuche.
- ✓ Das sogenannte Codecompletion ist eine grosse Hilfe bei der Codeeingabe: es bietet automatische Wortvervollständigung sowie die Anzeige von Dokumentationen als ToolTips.
- ✓ Das Syntax-Highlighting bewirkt, dass bekannte Datentypen farblich hervorgehoben werden.
- ✓ Viele weitere integrierte Tools wie beispielsweise „Code Snippets“ zum Einfügen kleiner Codebestandteile oder auch Refactoring-Tools erleichtern die Arbeit des Programmierers.

3.2 Installation

In diesem Kapitel installieren wir IntelliJ IDEA sowie die dazu benötigten Tools. Die Installationsdateien findest Du im Internet auf der Webseite von JetBrains: <https://www.jetbrains.com/idea/download/>

Lade die Community-Edition auf Deinen Rechner herunter und warte aber noch mit der Installation. Bevor Du die Installation startest, musst Du nämlich noch das Java Development Kit (JDK) installieren. Dies machen wir im nächsten Kapitel.

3.2.1 JDK – Java Development Kit

Der Java Development Kit (JDK) ermöglicht es Dir auf deinem lokalen Computer Java zu programmieren. Deshalb muss dieser Kit vor dem eigentlichen IntelliJ installiert werden.

Hast Du gewusst?

Der Java Development Kit (JDK) besteht aus den folgenden Kern-Elementen:

- Java Compiler *javac* – er übersetzt deinen Code in ein ausführbares Programm.
- Java Debugger *jdb* – er hilft Dir deinen Code Schritt für Schritt zu analysieren.
- Java Dokumentationstool *javadoc* – er generiert aus deinen Code-Kommentaren eine Dokumentation.
- Java Archiver *jar* - er fasst deinen kompilierten Java-Code in verteilbare Einheiten zusammen.



ICT Berufsbildung
Formation professionnelle
Formazione professionale

Hole Dir das aktuelle und auf Deinen Rechner passende **Java SE Development Kit (JDK)** von der Webseite von Oracle: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Klicke nun auf `jdk-XXXXX-windows-x64.exe`. Es öffnet sich ein Installationsablauf. Klicke Dich durch den Installationsablauf der in etwa wie folgt abläuft:

1. Der Start-Screen des Installationsprogrammes. Bitte überprüfe die Version.

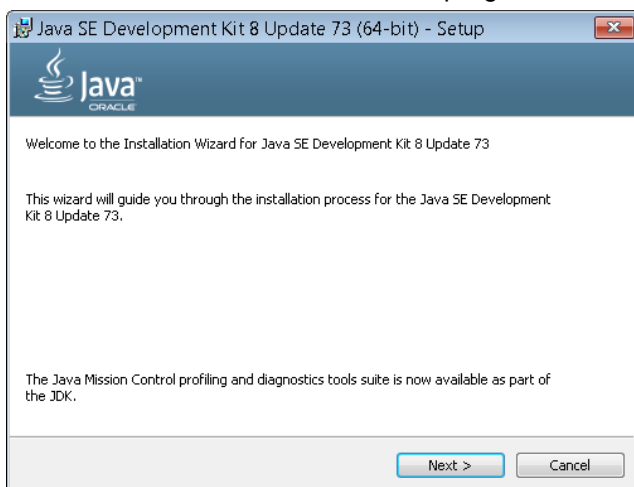


Abbildung 1: Installation Java JDK - Schritt 1

2. Wähle die entsprechenden Komponenten aus.

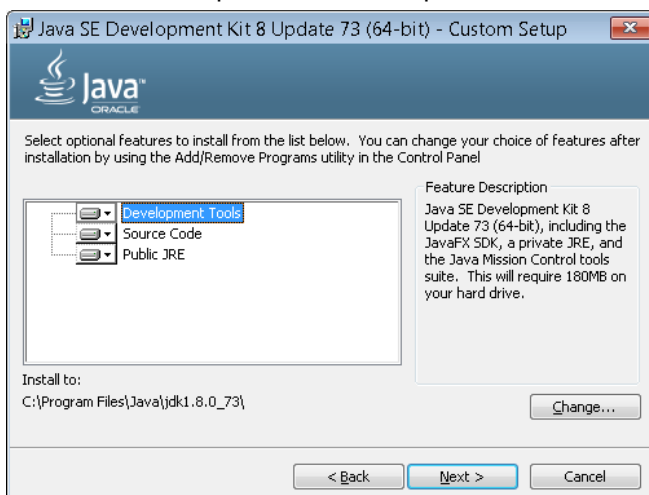


Abbildung 2: Installation Java JDK - Schritt 2



3. Wähle den Zielordner

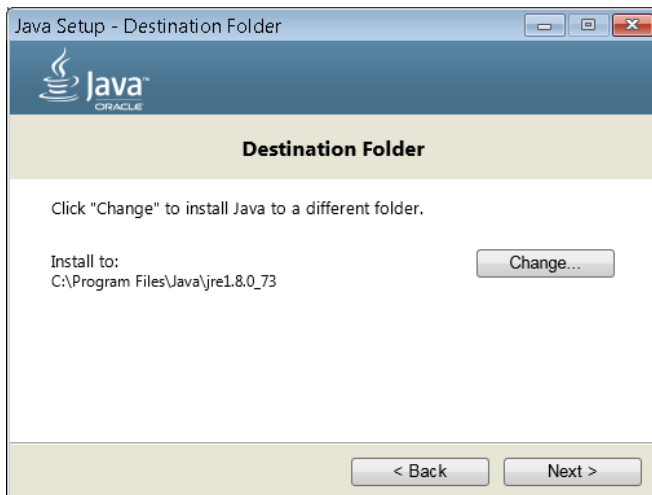


Abbildung 3: Installation Java JDK - Schritt 3

4. Die Installation wird durchgeführt.



Abbildung 4: Installation Java JDK - Schritt 4

Nun sollte das Java Development Kit installiert sein. Um dies zu überprüfen, kannst Du unter folgendem Pfad schauen, ob das Java Development Kit installiert ist: `C:\Program Files\Java\jdk1.8.0_73`

3.2.2 JAVA_HOME Umgebungsvariable

Wichtig: Auf Deinem Rechner sollte nun auch die Umgebungsvariable JAVA_HOME eingerichtet sein. Überprüfe dies, indem Du wie folgt vorgehst:

1. Systemsteuerung / System öffnen
2. Erweiterte Systemeinstellungen öffnen

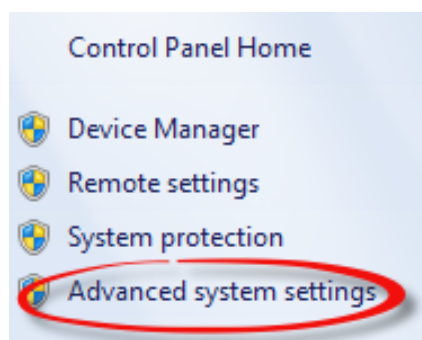


Abbildung 5: Umgebungsvariablen überprüfen - Schritt 1

3. Umgebungsvariablen öffnen

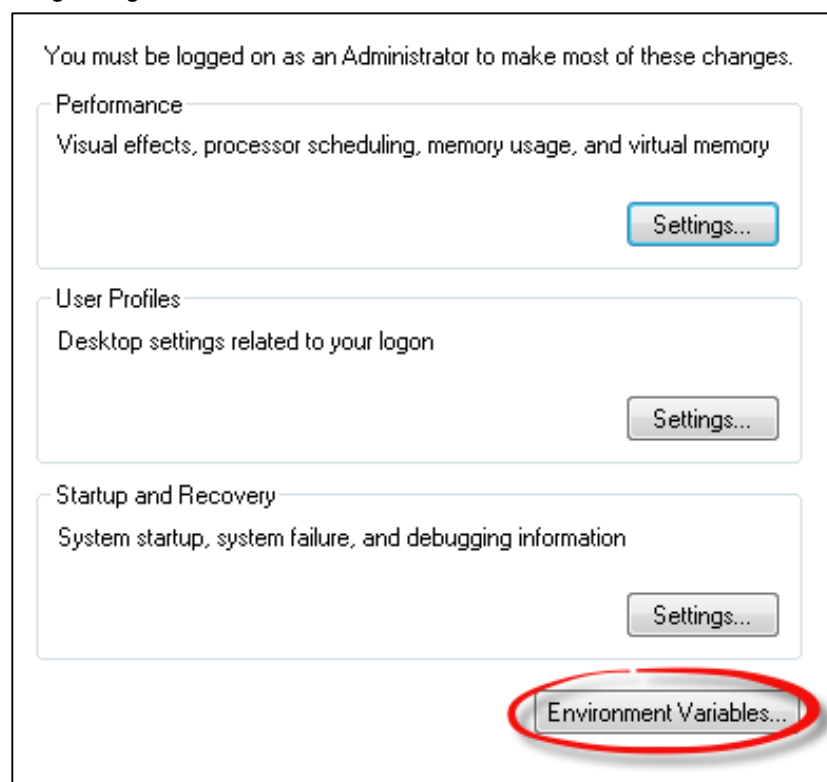


Abbildung 6: Umgebungsvariablen überprüfen - Schritt 2



4. Überprüfen, ob die Systemvariable JAVA_HOME angelegt ist. Falls nicht, diese über den Knopf "Neu..." anlegen.

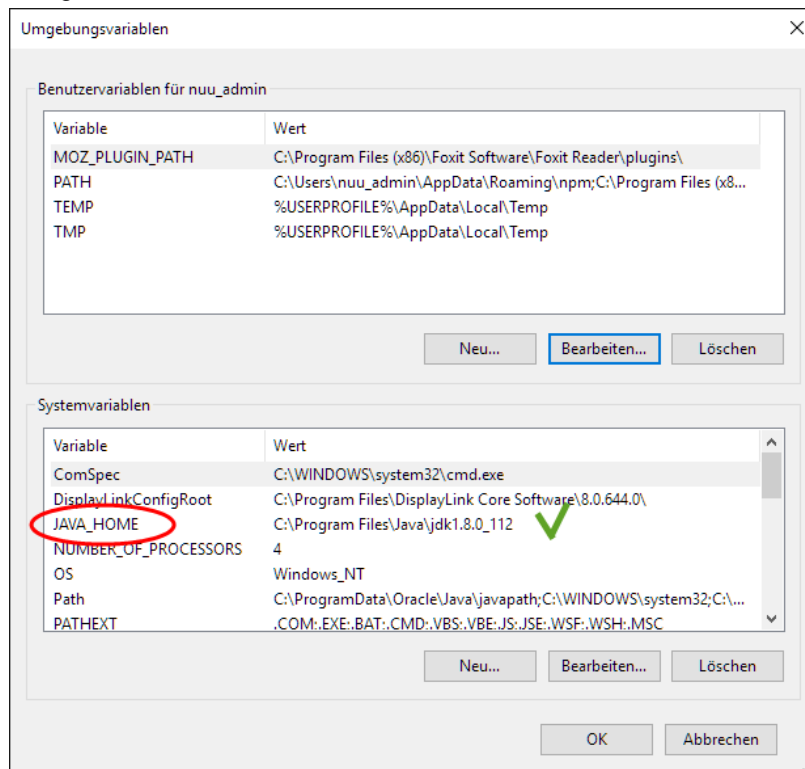


Abbildung 7: Umgebungsvariablen überprüfen - Schritt 3

Eine letzte Überprüfung kannst Du über die Konsole (Start/Ausführen/cmd) starten. Führe Dazu den folgenden Befehl in der Kommandozeile aus:

```
java -version
```

Du solltest nun die eben installierte Java-Version erhalten. Falls etwas bei diesen Punkten nicht stimmt, melde dich bei deinem Ausbilder.

Hast Du gewusst?

Du kannst bereits jetzt Java-Code kompilieren und ausführen (ohne eine „Entwicklungsumgebung“).

1. Hol Dir den Beispiel Code von hier: <https://gist.github.com/paroos/903a1fcecc1b767a019c>
2. Kopiere den Code in eine Textdatei und speichere die Datei unter dem Namen HelloWorld.java ab.
3. Öffne die Konsole und navigiere in Dein Java-Verzeichnis, z.B. c:\Program Files\Java\jdk1.8.0_65\bin
4. Kompiliere deinen Code mit dem Java-Compiler

```
javac "<Pfad zu deiner Java-Datei>\HelloWorld.java"
```

5. Lasse deinen Code laufen mit dem Befehl

```
java -cp "<Pfad zu deiner Java-Datei>" HelloWorld
```

Verstehst Du was hier abgeht?



3.2.3 Deine Entwicklungsumgebung - IntelliJ IDEA

Los geht's mit deiner eigenen Entwicklungsumgebung! Öffne die Datei idealC-15.0.4.exe. Folge dabei den Installationsschritten wie die folgenden Screenshots es Dir zeigen.

1. Startbildschirm des Installationsprogrammes



Abbildung 8: Installation IntelliJ Idea - Schritt 1

2. Auswahl des Ziel-Ordners

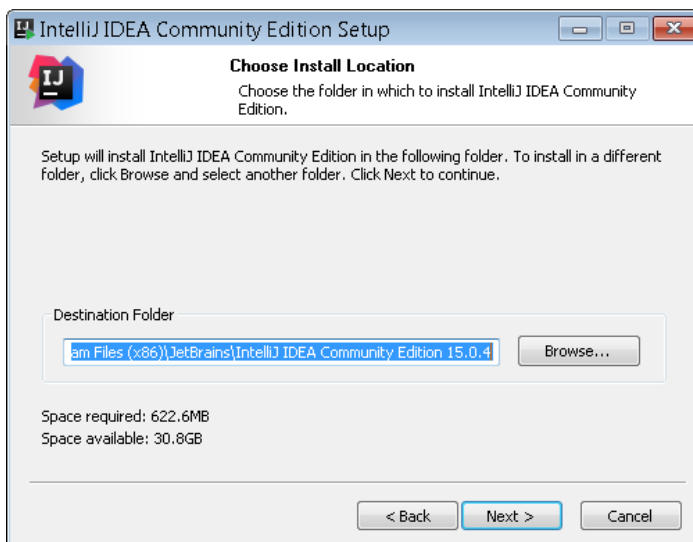


Abbildung 9: Installation IntelliJ Idea - Schritt 2



3. Datei Verknüpfungen

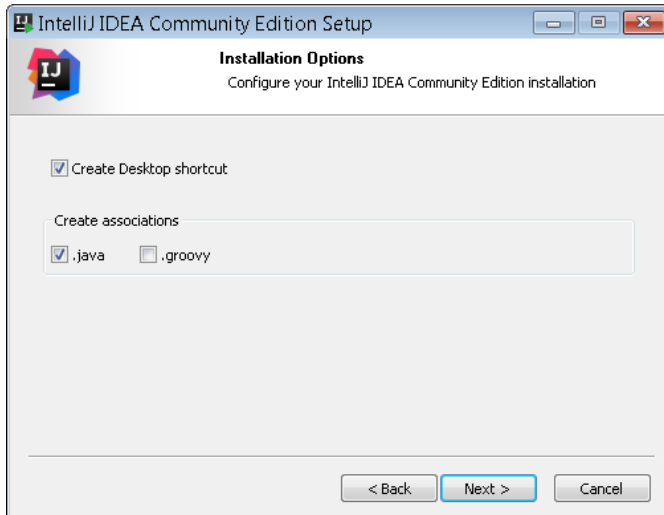


Abbildung 10: Installation IntelliJ Idea – Schritt 3

4. Verknüpfungen im Startmenü

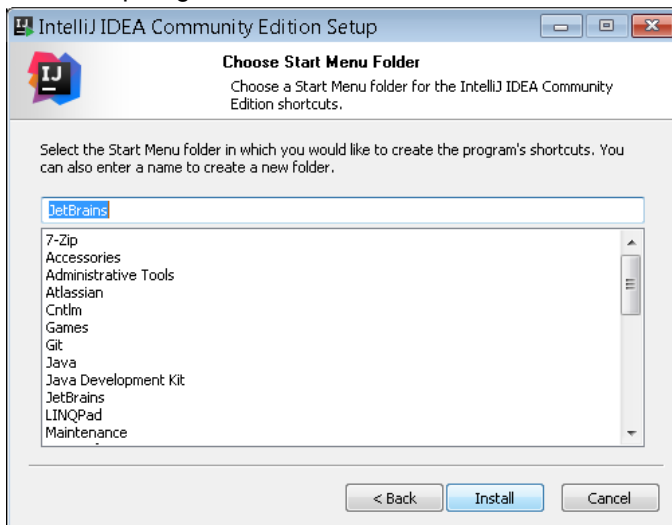


Abbildung 11: Installation IntelliJ Idea - Schritt 4



ICT Berufsbildung
Formation professionnelle
Formazione professionale

5. Die Installation läuft.

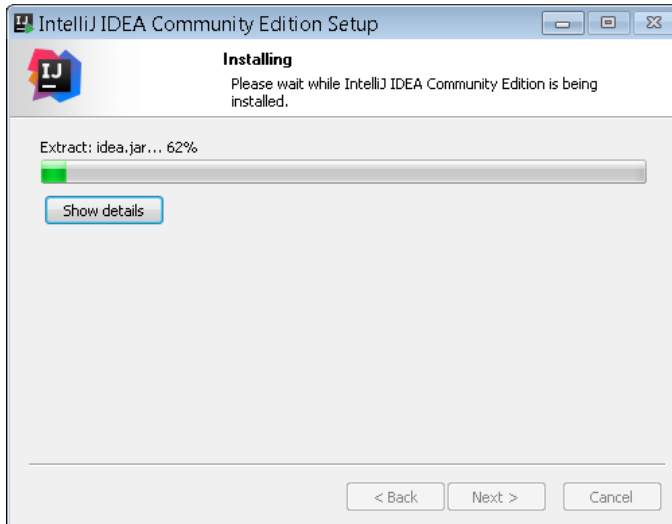


Abbildung 12: Installation IntelliJ Idea - Schritt 5

3.3 Erster Start und Konfiguration

IntelliJ sollte nun installiert sein. Starte die Entwicklungsumgebung um sie initial zu konfigurieren. Um IntelliJ zu starten, kannst Du auf die entsprechende Verknüpfung klicken, die Du auf dem Desktop und/oder im Startmenu deines Rechners findest.



Abbildung 13: IntelliJ Idea Desktop Icon



ICT Berufsbildung
Formation professionnelle
Formazione professionale

1. Wähle dein Theme aus – Du kannst zwischen Blau und Schwarz wählen.

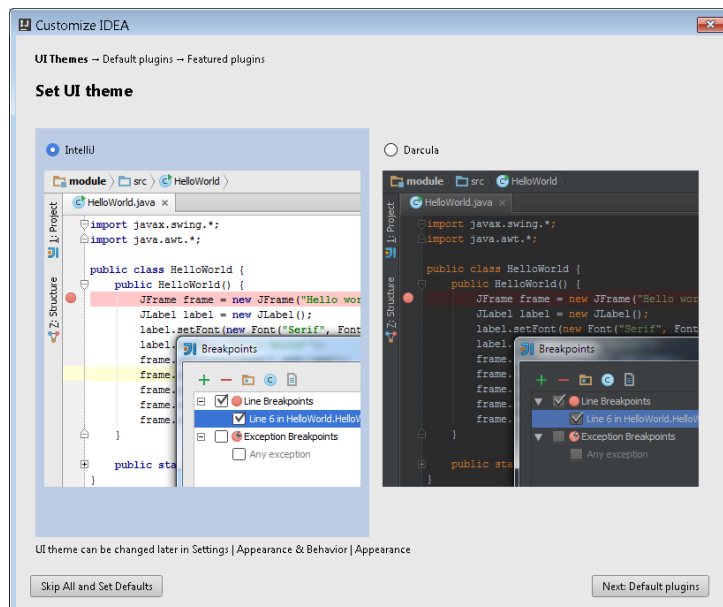


Abbildung 14: Konfiguration IntelliJ Idea - Schritt 1

Klicke anschliessend auf Next: Default Plugins.

2. Stelle sicher, dass der nächste Dialog wie folgt aussieht.

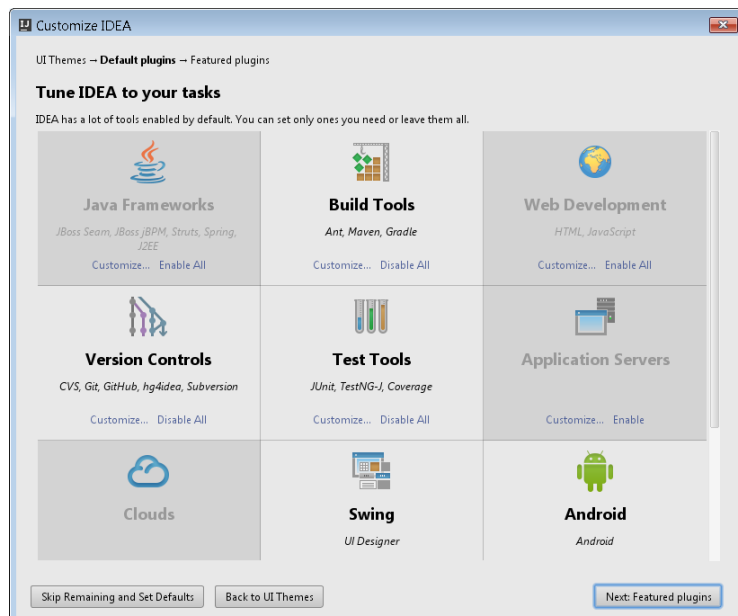


Abbildung 15: Konfiguration IntelliJ Idea - Schritt 2



Klick anschliessend auf Next: Featured Plugins. Dein IntelliJ sollte nun fertig konfiguriert sein.

Hast Du gewusst?

Mit IntelliJ und Java kannst Du auch eigene Android-Apps programmieren. Lerne heute Java und erstelle morgen deine eigenen [Android-Apps](#)!

Dein erstes Java-Projekt wirst Du weiter unten im Kapitel 4 erstellen. Zunächst werden Dir noch kurz die Bestandteile der IntelliJ-Entwicklungsumgebung erklärt.

3.4 Die Bestandteile der Entwicklungsumgebung

Dieses Kapitel beschreibt die verschiedenen Bestandteile der Entwicklungsumgebung IntelliJ Idea.

3.4.1 Aufbau IntelliJ

Der folgende Screenshot zeigt die wichtigsten Bestandteile der Entwicklungsumgebung.

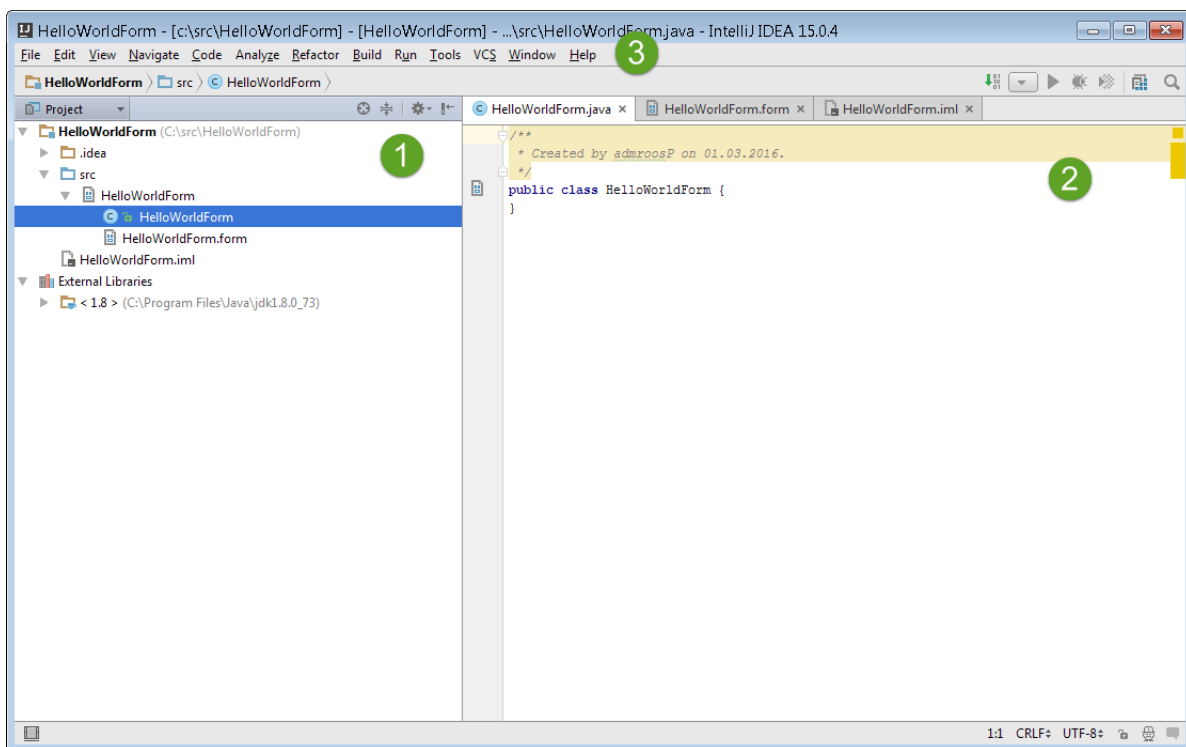


Abbildung 16: Aufbau IntelliJ



In der folgenden Tabelle findest Du die Erklärungen zu den einzelnen Bestandteile Deiner Entwicklungsumgebung.

Nummer	Beschreibung
1	Dies ist die sogn. <i>Project View</i> . In der Projektansicht findest Du dein gesamtes Projekt, d.h. alle Dateien, welche für dich relevant sind. Hier kannst Du durch deine sämtlichen Dateien navigieren, neue Dateien zu deinem Projekt hinzufügen oder entfernen.
2	Das ist der <i>Code-Editor</i> . Hier schreibst Du deinen Code.
3	Dies ist die <i>Symbolleiste</i> . Über diese erreichst Du sämtliche Operationen, die Du während dem Programmieren benötigst.

3.4.2 IntelliJ Code-Editor

Die folgende Abbildung zeigt den IntelliJ Code-Editor.



Abbildung 17: IntelliJ Code-Editor

Der Code-Editor besteht grundsätzlich aus den folgenden Teilen:

Nummer	Beschreibung
1	Dies ist der eigentliche Editor. Hier schreibst Du deinen Code. Der Editor zeigt Dir Fehler direkt an.
2	Die linke Leiste gibt Dir Kontext-Informationen. Falls Du ein GUI programmierst, zeigt Dir diese Leiste die Verknüpfung zwischen Code- und UI an.
3	Diese Leiste zeigt Dir wo Spezialitäten in deinem Code sind.



4 "Hello World" - Dein erstes Projekt

Das erste Programm, das Du nun mit IntelliJ erstellen wirst, ist eine Konsolenanwendung. Die Anwendung wird den Text "Hallo Welt" in der Konsole ausgeben.

Starte IntelliJ, rufe das Menu → **File** > **New** > **Project...** auf und folge den untenstehenden Schritten.

1. Wähle aus den Projektvorlagen den Eintrag "Java" aus
2. Klicke auf "Next"

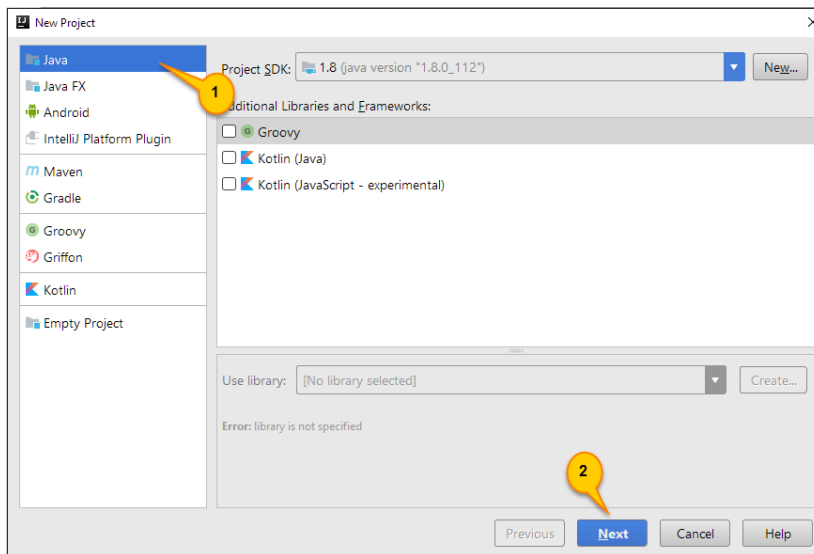


Abbildung 18: Konsolenapplikation erstellen, Schritte 1 und 2

3. Aktiviere die Option "Create project from template" und wähle den Eintrag "Command Line App" aus
4. Klicke auf "Next"

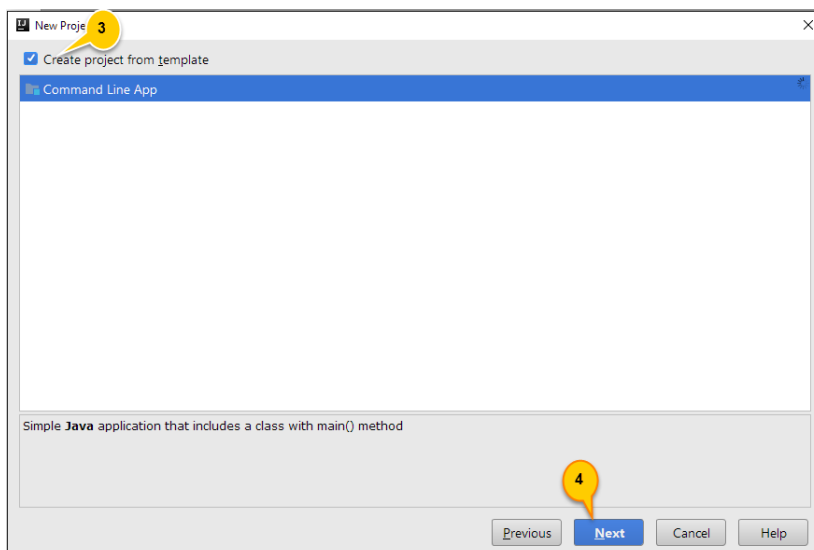


Abbildung 19: Konsolenapplikation erstellen, Schritte 3 und 4



5. Gib Deinem Projekt einen sprechenden Namen, z.B. HelloWorld
6. Klicke auf "Finish"

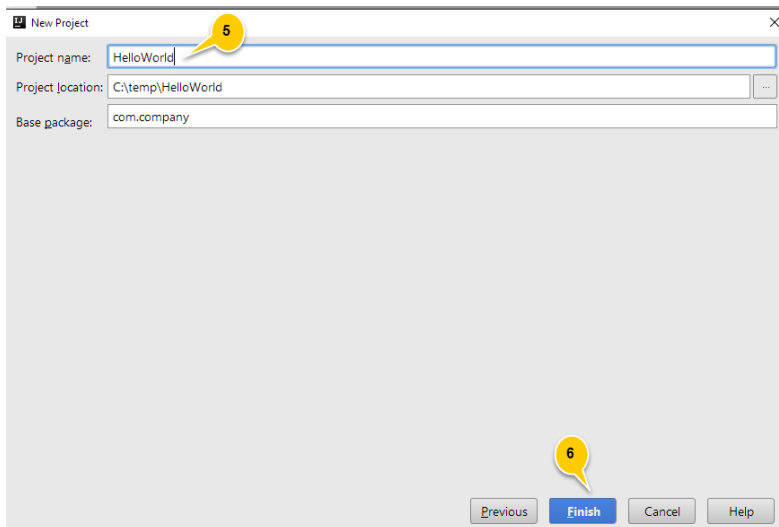


Abbildung 20: Konsolenapplikation erstellen, Schritte 5 und 6

Dein Code-Fenster sollte nun mit folgendem, von IntelliJ automatisch generierten Code gefüllt sein.



Abbildung 21: Konsolenapplikation, die von IntelliJ generierte main() Methode

7. Passe nun die main()-Methode so an, dass beim Start der Applikation der Text "Hello World" in der Konsole ausgegeben wird. Du kannst den Code dazu von hier kopieren:

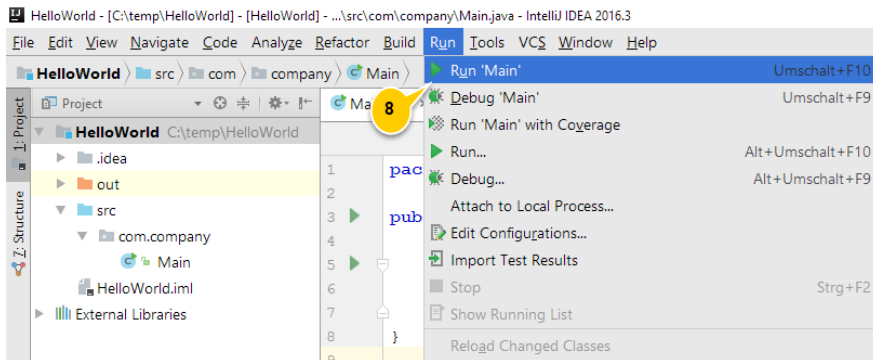
```
package com.company;

public class Main {

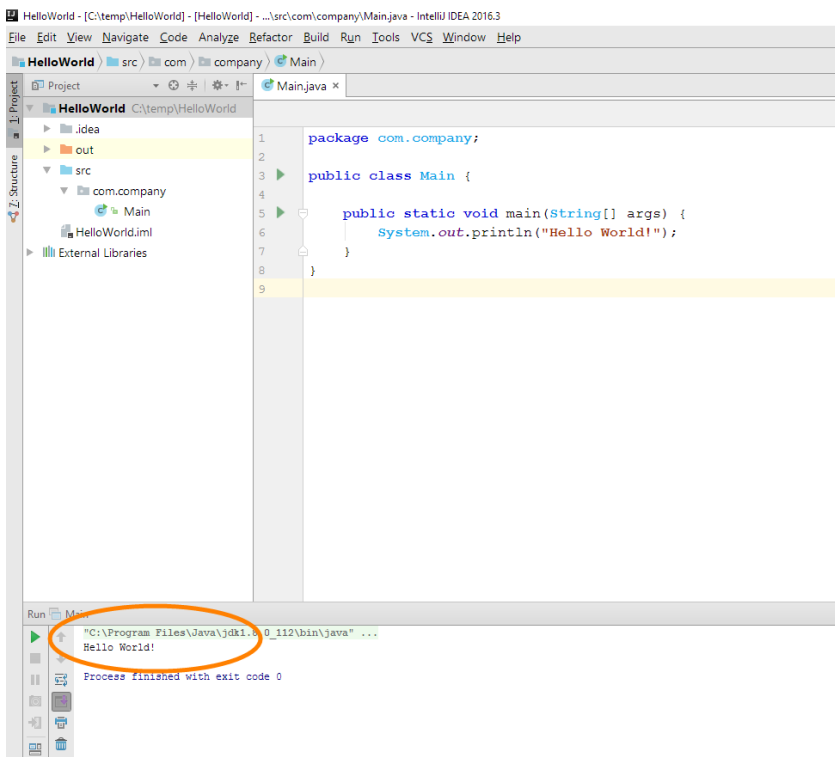
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```



8. Starte die Applikation indem über den Menu-Befehl Befehl *Run > Run 'Main'*



Stelle sicher, dass der gewünschte Text in der Konsole (siehe Bild) ausgegeben wird.



Gratuliere! Du hast soeben Dein erstes Java-Programm mit IntelliJ erstellt.