

JAVA Programmierung

ECLIPSE & JAVA 8

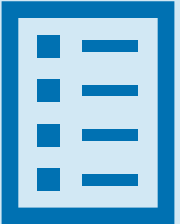


Themenübersicht

01

Einführung in Java

1. Vorwort
2. Grundkonzepte
3. Warum Java8
4. Erste Aufgabe
5. Erste Grundlagen





Java

01 Einführung in Java

1. *Vorwort*
2. *Grundkonzepte*
3. *Warum Java8*
4. *Erste Aufgabe*
5. *Erste Grundlagen*



1.1

Einführung in Java

Vorwort



1.1 Vorwort

Java ist eine objektorientierte Sprache. Sie wurde von 1991 bis 1992 von Sun Microsystems entwickelt.

=> <https://de.wikipedia.org/wiki/Java-Technologie#Geschichte>

Sie werden sich in diesem Kurs die objektorientierte Programmierung und verschiedene Java Technologien kennenlernen.

Für nicht Alles werden sie direkt Verwendung finden. Vielmehr geht es darum zu erfahren, was Java kann und wo die richtigen Informationen hierzu gefunden werden können, wenn sie gebraucht werden.

Der Name „Java“ beruht auf der Begebenheit, dass das örtliche Straßencafe in dem die meisten Programmierer ihren Kaffee tranken, „Java City“ hieß.

1.2

Einführung in Java

Grundkonzepte



1.2 Grundkonzept

- **Generelles**

=> Java ist ...

- **Objektorientiert**

Alles ist ein Objekt in Java. Selbst die Datentypen sind Objekt. Was ein Objekt ist werden wir später klären.

- **Sicher**

Ein Java-Programm läuft nicht direkt auf dem Betriebssystem. Der Java Quellcode wird zuerst vom Compiler in Bytecode übersetzt, dieser wird von einer im JDK mitgelieferten VM in Maschinsprache übersetzt.

- **Architekturneutral**

Java läuft auf allen Betriebssystemen.

- **Dynamisch**

Alles in Java basiert auf Bibliotheken (Packages), diese können sich zur Laufzeit ändern.

1.2 Grundkonzept

- Sicherheit / Architekturneutral / Dynamisch**

Programmiersprache		Java => Quelltext aus der IDE (Eclipse) in den .java Dateien
JDK		Übersetzt den Quelltext in ByteCode mittels Java-Compiler. Die JRE ist teil der JDK. Diese enthält zusätzliche Dateien auf welche wir hier nicht weiter eingehen wollen.
	JRE	<p><u>Sicherheit</u></p> <p>Mittels eine JVM (Java Virtuellen Maschine) wird der Bytecode in Maschinen Sprache übersetzt. Somit wird nicht direkt auf das Betriebssystem zugegriffen.</p> <p><u>Dynamisch</u></p> <p>Die Kompilierung läuft Just in Time. Somit können Bibliotheken zur Laufzeit geändert werden</p>
Betriebssystem		<p><u>Architekturneutral</u></p> <p>Windows, Linux, ...</p>

1.2 Grundkonzept

- **Objektorientiert => Was ist ein Objekt?**

Objekte in der OOP können Objekte aus der realen Welt abbilden. Um das zu tun, verfügen Objekte in der OOP über Eigenschaften und Methoden. Nehmen wir zum Beispiel einen Wasserkocher:

Ein Wasserkocher hat:

- Ein Volumen
- Eine Farbe
- Ein Sichtfenster
- ...

Das alles sind Eigenschaften des Wasserkochers. In der OOP werden diese Eigenschaften auch Attribute genannt. Wir können die Attribute mit Werten versorgen und diese Werte zur Laufzeit des Programms ändern.

1.2 Grundkonzept

- **Objektorientiert => Was ist ein Objekt?**

Alles, was ein real existierendes Objekt tun kann, wird in der OOP als Methode bezeichnet.

Ein Wasserkocher kann:

- Wasser kochen
- Wasser auf eine bestimmte Zieltemperatur bringen
- Ein akustisches Signal geben, wenn der Zielzustand erreicht ist
-

Ziel ist es, diese Methoden so zu programmieren, dass sie das gewünschte Verhalten an den Tag legen.

1.2 Grundkonzept

- **Objektorientiert => Was ist ein Objekt?**

Die in der OOP abgebildeten Objekte können von beliebiger Komplexität sein. Diese hängt zum Einen davon ab, wie komplex das betrachtete reale Objekt generell ist

Komplexität aufsteigend

Brotmesser -> Wasserkocher -> Mikrowelle -> Hightech-Herd

und zum Anderen davon, wie weit wir bei der Abbildung des realen Objekts in die Tiefe gehen wollen (oder müssen), was wiederum davon abhängt, was der Sinn und Zweck unseres Programms sein soll.

Beispiel: Mensch

Mitarbeiterverwaltung: Nachname, Vorname, Personalnummer, Geburtsdatum, Abteilung, ...

Biologie-Software: Skelett, Muskeln, Sehnen, Blutbahnen, innere Organe, ...

1.3 Einführung in Java

Warum Java8



1.3 Warum Java8

Java 8 ist die letzte lizenzfreie Version. Sie erschien 2014 und bis einschließlich 2019 gab es kostenlose und lizenzfreie Updates.

Danach musste ein Oracle Account erstellt werden und für weitere Updates wurden Lizenzgebühren erhoben.

Da in den meisten Firmen aus Lizenzgründen mit Java8 entwickelt wird, werden wir in diesem Kurs ebenfalls mit Java8 arbeiten.

1.4 Aufgaben

Aufgabe



1.4 Aufgabe

Lassen Sie uns nun gemeinsam ein erstes Projekt erstellen. Dies soll Folgendes beinhalten:

- **Eine Klasse**
- **Eine main Methode**

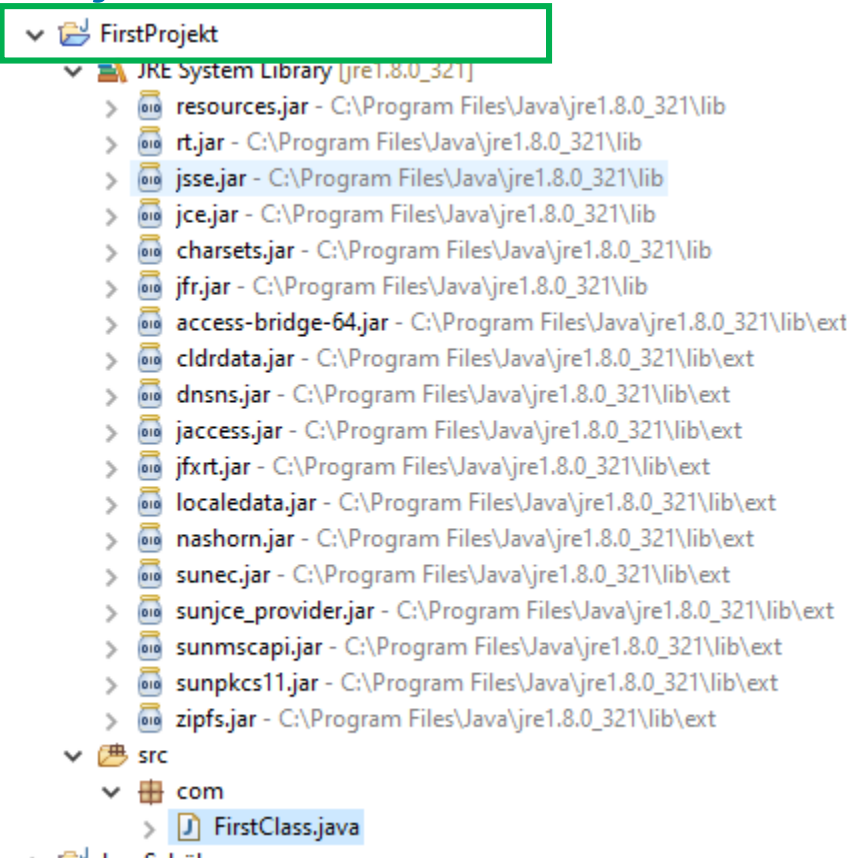
1.5 Einführung in Java

Erste Grundlagen



1.5 Erste Grundlagen

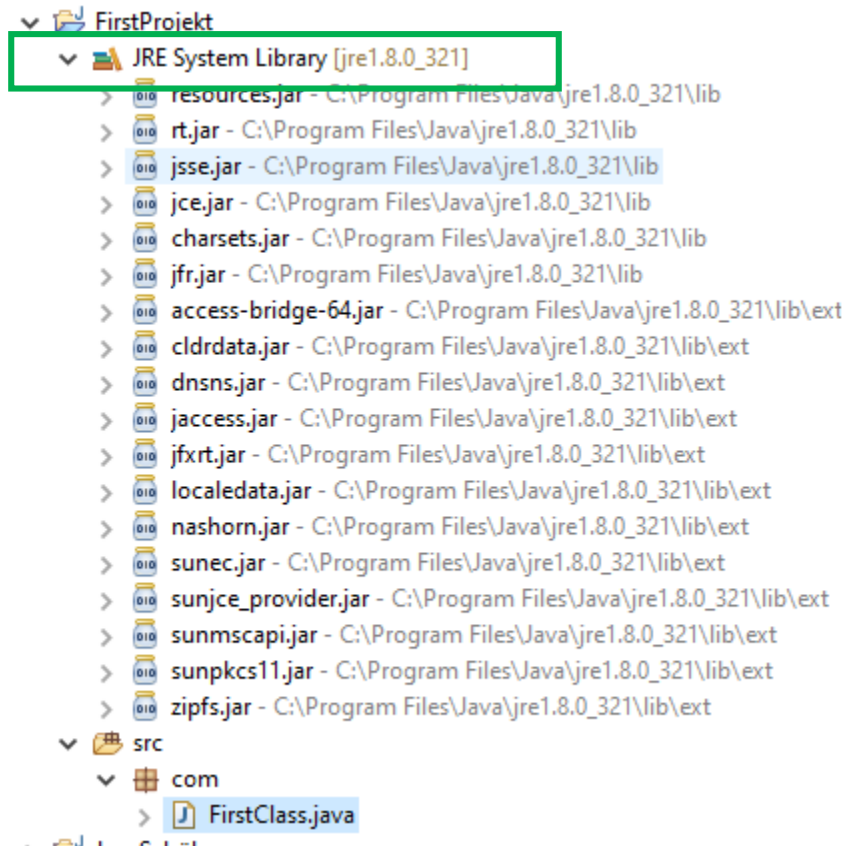
Projekte => Übersicht und Informationen



Der Root Ordner (oberster Knoten im Projekt) wird mit einem kleinen „J“ markiert, was uns den Projekt Typen Verrät. In unserem Fall ist das Java. Es gibt noch weitere.

1.5 Erste Grundlagen

Projekte => Übersicht und Informationen



Innerhalb von jedem Projekt liegt die

⇒ JRE System Library

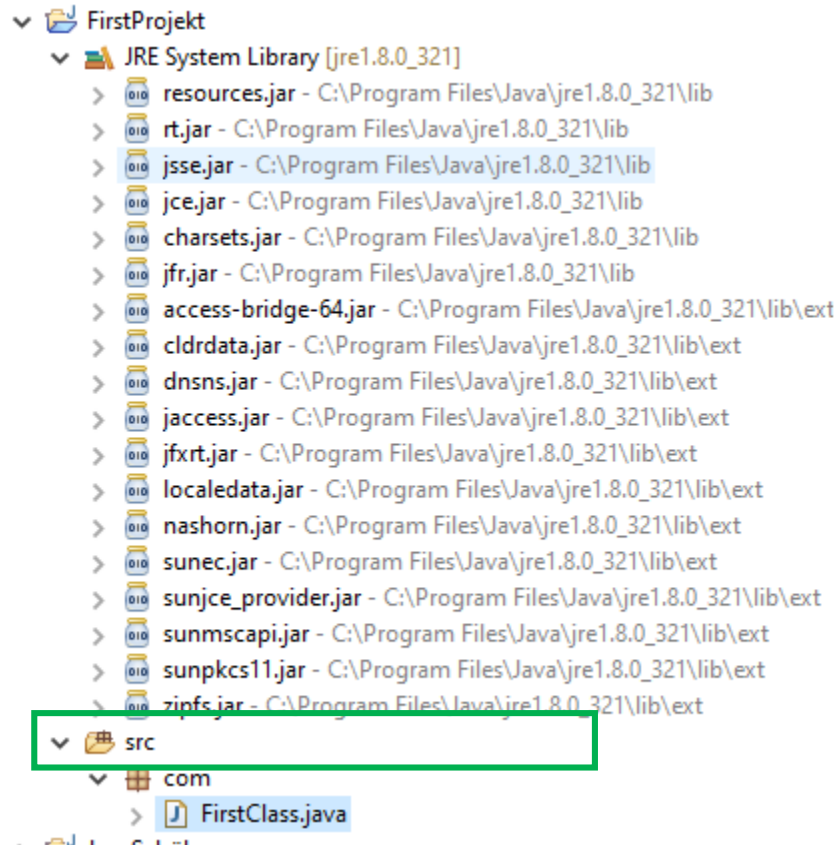
In Klammern dahinter folgt die Version

⇒ jre1.8.0_321

Hier werden, wie der Name schon sagt, Bibliotheken von Java eingebunden.

1.5 Erste Grundlagen

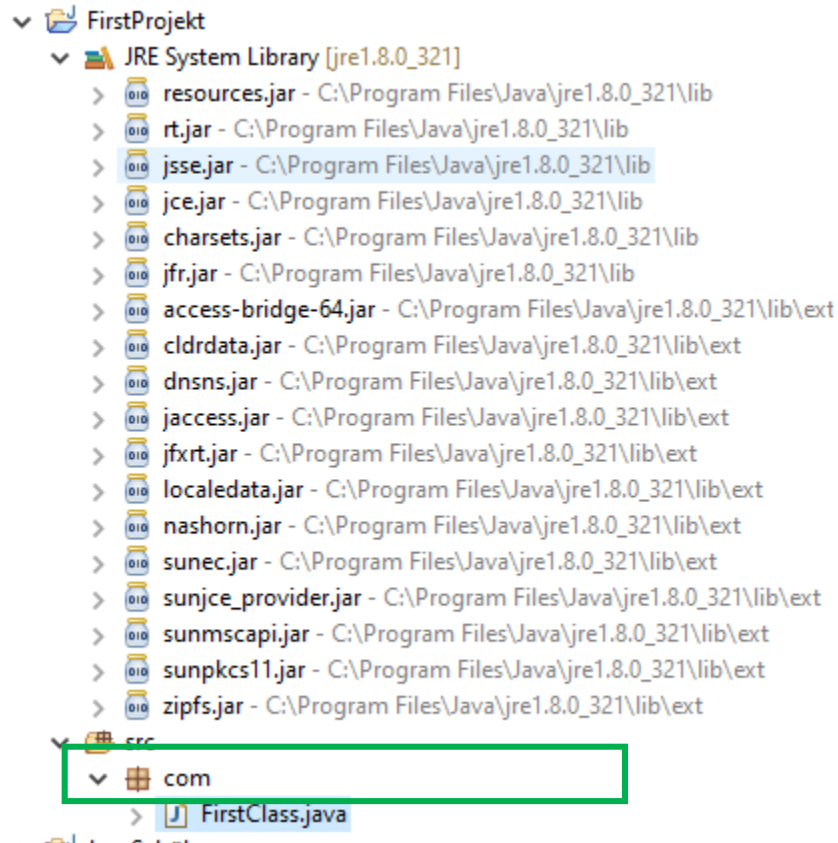
Projekte => Übersicht und Informationen



Unterhalb des Root Ordners (Verzeichnis) folgt der Ordner „src“. Dieser wird miterstellt, wenn wir dies bei der Projekt Anlage explizit auswählen. Das kann bei unterschiedlichen Projekt Typen abweichen.

1.5 Erste Grundlagen

Projekte => Übersicht und Informationen



Packages

Dies sind, wie Sie schon am Symbol erkennen können => Pakete. Sie werden „default“ mäßig immer erstellt, sofern Sie nicht explizit eins miterstellen oder auswählen. Packages sind vergleichbar mit Bücherregalen innerhalb einer Bibliothek => dem Projekt. Hier werden Sie ihre Java Dateien ablegen welche das Buch innerhalb der

⇒ Bibliothek

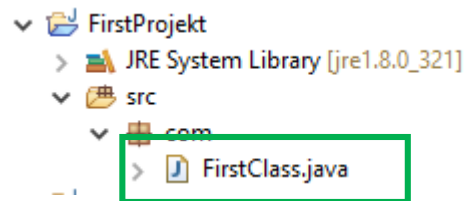
innerhalb des

⇒ Bücherregals

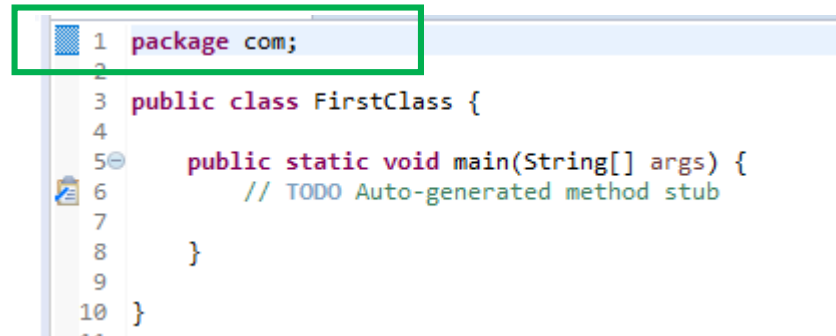
darstellen.

1.5 Erste Grundlagen

Pakete => packages



In der ersten Zeile finden wir den Verweis auf unser Package (Paket).

A screenshot of the Eclipse IDE's code editor. The first line of the code is 'package com;', which is highlighted with a green box. The rest of the code shows a public class 'FirstClass' with a 'main' method that has a TODO comment.

```
1 package com;  
2  
3 public class FirstClass {  
4  
5     public static void main(String[] args) {  
6         // TODO Auto-generated method stub  
7     }  
8  
9  
10 }
```

1.5 Erste Grundlagen

Import => package

```
2
3 import java.util.*;
4 import java.util.List;
5 import java.util.Optional;
6 import java.util.regex.Pattern;
7
```

Auch wir können Pakete einbinden.
Hierzu wird das Schlüsselwort „import“
(speziell für diesen Zweck reserviert)
verwendet. Als Analogie für den Aufbau eines
solchen Imports beziehen wir uns wieder auf
eine Bibliothek.

- import => Es wird ein Buch benötigt
- java => in welchem Genre
- util => welcher Autor
- List => welches konkrete Buch

1.5 Erste Syntax

Import => package

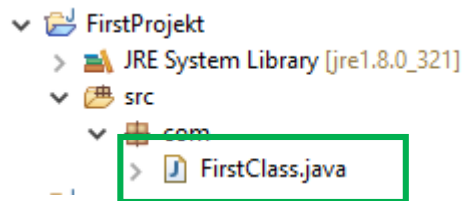
```
1
2
3 import java.util.*;
4 import java.util.List;
5 import java.util.Optional;
6 import java.util.regex.Pattern;
7
```

- import => Es wird ein Buch benötigt
- java => in welcher Genre
- util => welcher Autor
- * => **Wildcard**

Das Sternchen ist eine Besonderheit. An dieser Stelle ist uns das konkrete Buch egal. Wir benötigen alle Bücher des jeweiligen Autors.

1.5 Erste Grundlagen

Klassen => und der „Auto-generated“ Code

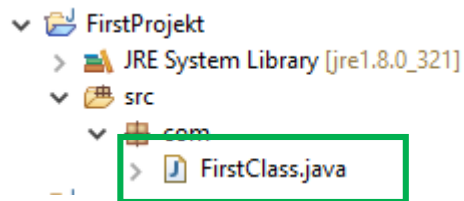


```
1 package com;  
2  
3 public class FirstClass {  
4  
5     public static void main(String[] args) {  
6         // TODO Auto-generated method stub  
7  
8     }  
9  
10 }
```

Alles in den objektorientierten Sprachen basiert auf der Klassen-Technologie. Eine Klasse wird immer beim Erstellen einer Java-Datei mit erzeugt. Der Namen ist identisch mit dem Dateinamen. Gekennzeichnet wird die Klasse mit dem Begriff „class“.

1.5 Erste Grundlagen

der „Auto-generated“ Code



```
1 package com;
2
3 public class FirstClass {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7     }
8
9 }
10 }
```

Jedes Java Projekt besitzt einen einzigen Einstiegspunkt! Sollte dieser nicht den korrekten Aufbau haben, wird das Programm nicht starten.

Der Einstiegspunkt ist vergleichbar mit dem Haupteingang einer Bibliothek.
=> Es gibt nur einen und der befindet sich an einem bestimmten Platz im Gebäude.



VIELEN DANK!

