DIE SÄULEN DER OBJEKTORIENTIERTEN PROGRAMMIERUNG TEIL 1/2

VOM 05/11/2024 BIS 22/11/2024

VORSTELUNGSRUNDE

Name

Background

C++?

JAVA?

Programmierkentnisse

Ziel

Python

KURSPLAN

Timeline

• 13 Tage

Themen

• Java Einführung
• Java-Grundbausteine
• Kontrollstrukturen
• Java-Grundbausteine
• Methoden
• Strings und Wrapper-Klassen
• Arrays und Enums
• Collections-Framework – Grundlagen
• Ausnahmebehandlung mit Exceptions

INTRODUCTION TO JAVA

BY BETI

WHAT IS JAVA

Java is a popular programming language, created in 1995.
It is owned by Oracle, and more than **3 billion** devices run Java.
It is used for:

Java is used for

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

WHY USE JAVA?

Why use Java

Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)

It is one of the most popular programming languages in the world

It has a large demand in the current job market

It is easy to learn and simple to use

It is open-source and free

It is secure, fast and powerful

It has huge community support (tens of millions of developers)

Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs

As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa
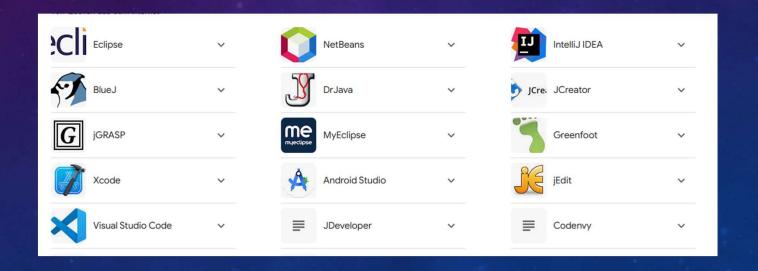
JAVA IDES

An IDE (Integrated Development Environment) allows you to quickly create applications,
by combining a source-code editor with the ability to compile and run your code, as well as integration with build,
test and debug tools, version control systems, and so on. Finally, an IDE will let you search and navigate your codebase
in ways your file system won't.
One of the most widely used integrated development environments (IDEs) for Java is IntelliJ IDEA.
Its user-friendly interface,
rich feature set, and vast ecosystem make it an ideal environment for beginners to learn and grow as developers.

INTELLIJ IDEA

IntelliJ IDEA is a cross-platform IDE that provides consistent experience on the Windows, macOS, and Linux operating systems.
IntelliJ IDEA is available in the following editions:

IntelliJ Editions

Community Edition is free and open-source, licensed under Apache 2.0. It provides all the basic features for JVM and Android development.
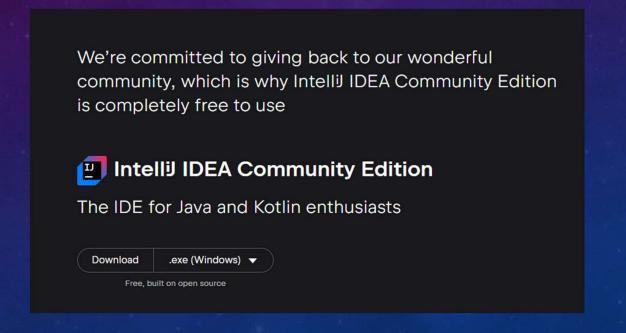
IntelliJ IDEA Ultimate is commercial, distributed with a 30-day trial period. It provides additional tools and features for web and enterprise development. For pricing information, refer to Subscription options & Pricing.

INSTALLATION

In order to install IntelliJ IDEA, you will first need to navigate to jetbrains.com/idea. From here, you can click on the "Download" button, and you'll be redirected to the download page.
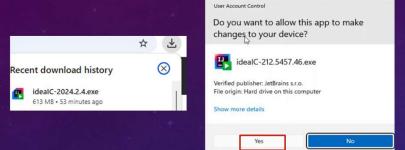
You'll be provided with two available editions to download: The free Community edition and the paid Ultimate edition. We'll be installing the Community edition.
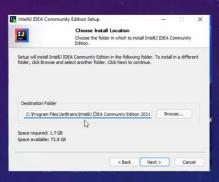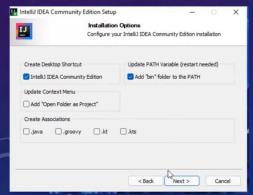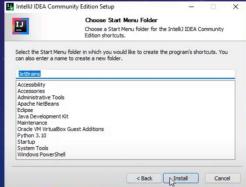Just scroll down and click download.

INSTALLATION

Open the .exe file   Allow the security message
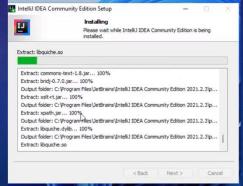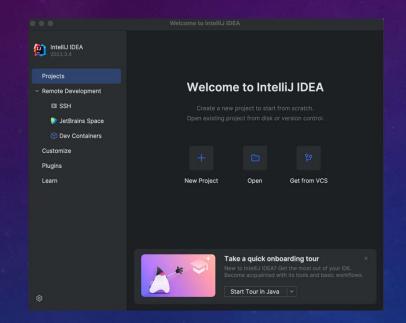
INSTALL INTELLIJ IDEA

When you launch IntelliJ IDEA for the first time, you'll see the **Welcome** screen.
From here, you create a new project, open an existing project, or get a project from a version control system (like GitHub).

To start working with Java, you will need to install a JDK.
You can do this yourself, or you can do so in IntelliJ IDEA when creating a new project,
without having to switch from your IDE and other tools (such as your browser, file system, etc.)
 to download and configure a JDK.
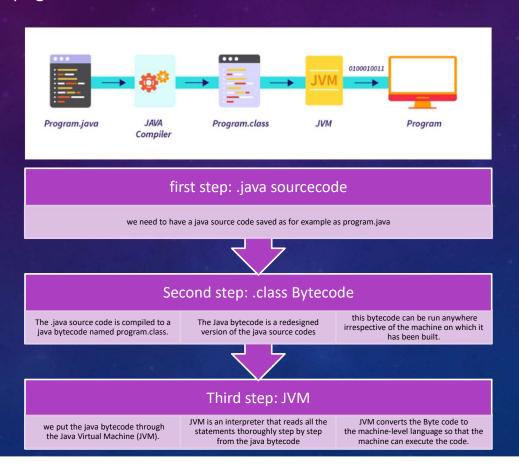
HOW JAVA PROGRAM WORKS?

Java is a high-end programming language loved by many programmers for its interesting features. We cant run Java programs on a normal machine this is because high-level programming languages need to be translated into machine code first, due to this we run them on a Java compiler.

The execution of a Java program consists of five steps which are	Creation of a Java Program
	Compiling a Java program,
	Loading the program into the memory by Java Virtual Machine,
	Java Virtual Machine verification for bytecode,
	Java Program execution

HOW DOES JAVA PROGRAMMING LANGUAGE WORK?

The working of the Java programming language can be summed up in three steps.
Let's go through the steps given below:

first step: .java sourcecode
we need to have a java source code saved as for example as program.java

Second step: .class Bytecode		
The .java source code is compiled to a java bytecode named program.class.	The Java bytecode is a redesigned version of the java source codes	this bytecode can be run anywhere irrespective of the machine on which it has been built.

Third step: JVM		
we put the java bytecode through the Java Virtual Machine (JVM).	JVM is an interpreter that reads all the statements thoroughly step by step from the java bytecode	JVM converts the Byte code to the machine-level language so that the machine can execute the code.

JAVA ARCHITECTURE AND COMPONENTS

Java Architecture is a collection of components, namely
- JVM (Java Virtual Machine),
- JRE (Java Runtime Environment),
- JDK (Java Development Kit).
- It orchestrates the process of both interpretation and compilation,
- delineating all processes involved in creating a Java program.

WHAT ARE THE COMPONENTS OF JDK IN JAVA?

java	It is responsible for loading the class files and interpreting the source code compiled initially by javac.
javac	It correctly specifies the java compiler for translating the source code into bytecode.
javadoc	The comments that are added in the source code are documented by the javadoc.
jar	It assists the archives with adequately managing the jar files in the library package.
jps	JPS stands for Java Virtual Machine Process Status Tool, which is responsible for managing the active JVMs for the ongoing execution of the Java programs or a Java command.
appletviewer	It can run and debug Java applets without requiring a connection with an internet browser.
Idlj	This compiler is responsible for generating Java bindings from a given Java IDL file.
javap	It behaves as a file disassembler.
JConsole	It simply behaves as a Java Management and Monitoring unit.
javah	It is both a stub-generator and a C-Header, which are helpful in writing native methods.
javaws	The javaws works as the Web Start launcher for JNLP applications.
jhat	It is a heap analysis tool.
jmc	The full form of JMC is Java Mission Control.

JAVA EINFÜHRUNG

Let's Create our first JAVA project ☺