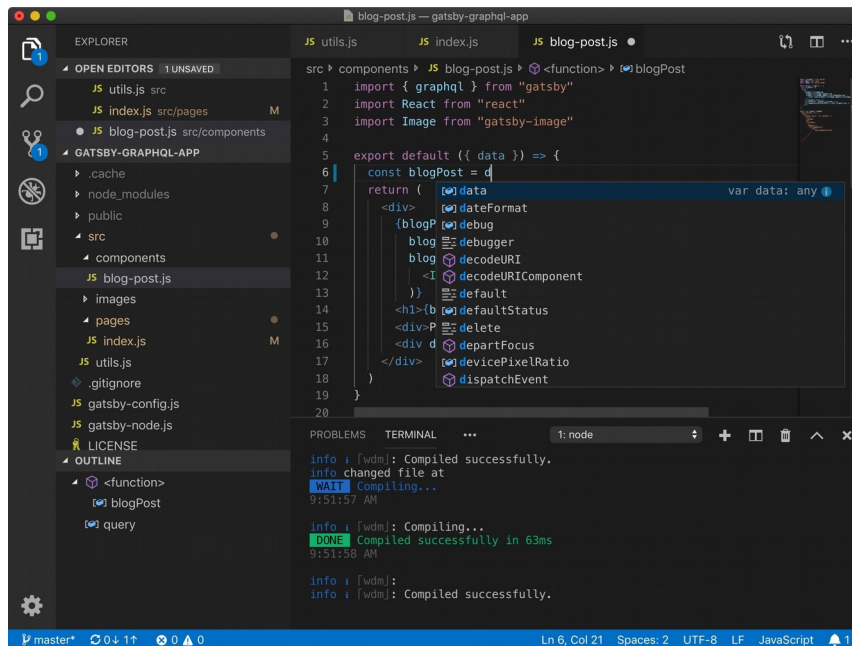


Visual Studio Code für den Unterricht



Einleitung

Dieses Dokument beschreibt den Umgang mit Visual Studio Code für den Unterricht. Visual Studio Code (kurz VS Code) ist ein Quelltext-Editor mit folgenden Merkmalen:

- Plattformübergreifend (Windows, macOS, Linux)
- Hybrid zwischen Editor und IDE
- Unterstützt eine Vielzahl an Programmiersprachen
- Erweiterbar durch Plugins
- Offener Quellcode
- Kostenlos
- Freie Lizenz (Sourcecode → MIT-Lizenz)

Dieses Dokument bezieht sich auf die neueste Version von Visual Studio Code.

Nutzungsdatensammlung

Beachten Sie, dass die Binary-Varianten von VS Code unter einer proprietären Lizenz (Microsoft: <https://code.visualstudio.com/license>) stehen. Dies ist durch die Platzierung von Markenkennzeichen und Sammlung von Nutzerdaten begründet. Möchte man die Sammlung von Nutzerdaten umgehen, kann man einer der folgenden Schritte durchführen:

- Telemetrie in VS Code ausschalten: <https://code.visualstudio.com/docs/getstarted/telemetry>
- VS Code aus dem Quellcode erstellen und diese Variante verwenden.
- Build-Prozess von VS Codium verwenden und diese Variante verwenden: <https://github.com/VSCodium/vscodium>

Bezug zwischen Visual Studio Code und Visual Studio

Microsoft bietet neben Visual Studio Code auch das Programm Visual Studio an. Visual Studio ist eine proprietäre IDE für gewisse Hochsprachen. Es wird oft fälschlicherweise angenommen, dass diese Programme eine Verbindung oder Ähnlichkeit aufweisen. Lediglich der Name, das Logo und einzelne Funktionen sind gleich, der Rest unterscheidet sich zwischen den Programmen. **Dieses Dokument bezieht sich nur auf Visual Studio Code.** Auch kann Visual Studio gewisse Unterrichtsbedingungen nicht erfüllen (z.B. wird die Sprache Java in Visual Studio nicht unterstützt).

Installation VS Code

VS Code kann von der Hauptseite heruntergeladen und installiert werden: <https://code.visualstudio.com/>

Beachten Sie auch, dass für gewisse Sprachen vorgefertigte Installationsprogramme existieren (z.B. Java für das Betriebssystem Windows). Lesen Sie sich dabei in die entsprechenden Kapitel des Dokumentes ein.

Java-Programmierung

Installation Java in VS Code

Windows

Für das Betriebssystem Windows existiert eine speziell erstellte VS Code-Installation für die Sprache Java. Diese kann unter folgender Seite, durch anklicken des Knopfes *Download Visual Studio Code Java Pack Installer*, bezogen werden:

<https://code.visualstudio.com/docs/languages/java>

Dabei wird VS Code, das Java-Plugin für VS Code sowie die Java-Umgebung (JDK) installiert.

macOS und Linux

Für die Betriebssysteme macOS und Linux kann die Grundinstallation, wie im Abschnitt *Installation VS Code* beschrieben, durchgeführt werden. Als nächstes muss das Plugin für Java installiert werden:

1. VS Code starten
2. Auf dem Bereich *Extensions* klicken
3. Im Suchfeld *Java* eingeben
4. *Java Extension Pack* auswählen und installieren
5. VS Code neustarten

Zudem muss, falls nicht bereits vorhanden, eine Java-Umgebung (JDK, z.B. openJDK) installiert werden. Dies kann entweder durch die Paket-Manager der Betriebssysteme (empfohlen) oder mit dem Programm VS Code durchgeführt werden. Für den Bezug der Umgebung mit dem Programm VS Code können folgende Schritte durchgeführt werden:

1. VS Code starten
2. Taste *F1* drücken
3. Im Eingabefeld *Java* eingeben und das Feld *Java: Configure Java Runtime* mit Pfeiltasten und Enter auswählen
4. Im Bereich *Install* auf Download klicken und den Inhalt an einem bestimmten Ort extrahieren
5. Taste *F1* drücken
6. Im Eingabefeld *settings* eingeben und das Feld *Preferences: Open Settings (JSON)* mit Pfeiltasten und Enter auswählen

7. Im JSON-Dokument um folgenden Eintrag erweitern:
`"java.home": "<Pfad zu JDK>"`
Wobei *<Pfad zur JDK>* mit dem absolutem Pfad des extrahierten Ordners ersetzt werden soll.
8. VS Code neustarten

Umgang mit Java-Projekte vereinfacht

Einrichtung

VS Code erlaubt es einfache Java-Projekte anzulegen (*standalone java file support*). Diese Variante wird vor allem für die ersten Schritte in der Java-Programmierung empfohlen. Zudem können so auch kleinere Projekte realisiert werden. Dazu wird lediglich ein Ordner benötigt:

1. Legen Sie einen Ordner für das Projekt an
2. Im Programm VS Code drücken Sie auf *File* → *Open Folder* und wählen den Projektordner aus

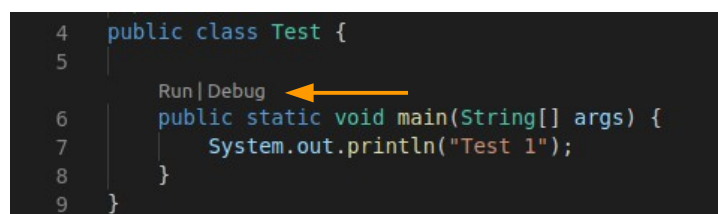
Wenn Sie eine Java-Datei in diesem Ordner anlegen, betrachtet VS Code diesen Ordner als Java-Projekt.

Beachten Sie: Java-Dateien müssen immer die Endung `.java` aufweisen. Zudem muss der Dateiname gleich der Klassenname sein. Wenn Sie also die Klasse *Datenstrukturen* (*public class Datenstrukturen{...}*) erstellen wollen, muss diese in der Datei *Datenstrukturen.java* abgelegt sein!

Ausführung (Debug)

Um das Programm auszuführen, können Sie in VS Code die Java-Datei mit der gewünschten Main-Methode öffnen und die Taste *F5* drücken. Beim erstmaligen Ausführen erstellt Ihnen VS Code den versteckten Ordner `.vscode` und die darin befindende Einstellungs-Datei *launch.json*. Sollte dies der Fall sein, können Sie wieder zur Ansicht der Datei mit der Main-Methode wechseln und wiederholt *F5* drücken. Nun sollte das Programm im Debug-Modus gestartet werden.

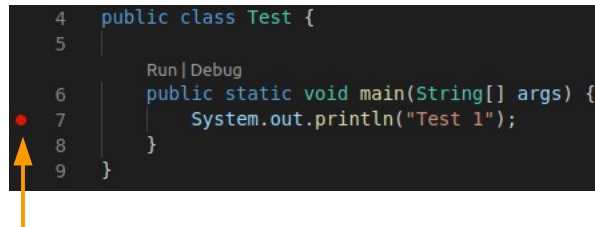
Eine Weitere Möglichkeit zur Ausführung des Programmes besteht durch das Anklicken der *debug*-Schaltfläche im Quellcode. Dazu öffnen Sie wieder die gewünschte Datei mit der Main-Methode. Über der Main-Methode sollten nun die zwei Schaltflächen *run* und *debug* stehen:



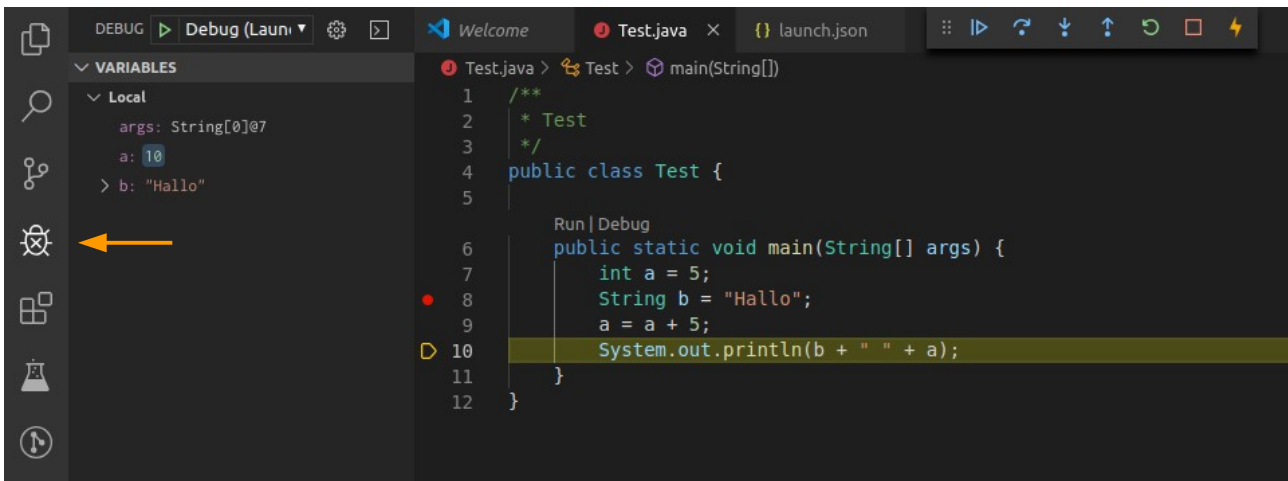
Drücken Sie auf *debug*, damit sich das Programm im Debug-Modus startet. Diese Variante ist vor allem Hilfreich, wenn Ihr Projekt aus mehreren Programmen und somit aus mehreren Main-Methoden besteht.

Debugging

In VS Code können Sie Break-Points durch Doppelklick neben der Zeilennummer setzen:



Beim Ausführen des Programmes im Debug-Modus wird das Programm an dieser Stelle dann Anhalten. Der Editor sollte dabei in die Debug-Kategorie wechseln. Sollte dies nicht der Fall sein, können Sie durch anklicken der Debug-Kategorie in diese wechseln:

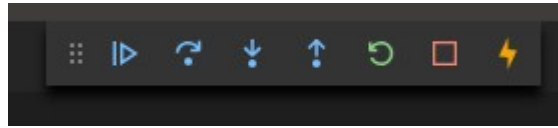


In dieser Kategorie sehen Sie 4 Unterteilungen:

- *VARIABLES*
 - Aktuelle Werte der Variablen im aktuellen Scope
- *WATCH*
 - Aktuelle Werte von Variablen welche speziell für die Überwachung markiert wurden
- *CALL STACK*
 - Ablaufstruktur bezogen auf Ausführung und Threads (Hilfreich bei parallelen Programmabschnitte und Mehrfachausführungen)

- **BREAKPOINTS**
 - Auflistung aller gesetzten Breakpoints. Erlaubt das aktivieren oder deaktivieren von einzelnen oder allen Breakpoints.

Zudem erhalten Sie im Debug-Modus das entsprechende Steuerelement im oberen Bereich:

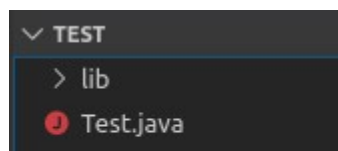


Externe Jar-Dateien

In VS Code können Sie externe Jar-Dateien in dieser vereinfachten Projektstruktur über den Ordner *lib* hinzufügen. VS Code bindet alle in diesem Ordner enthaltenen Jar-Dateien dann automatisch ein. Gehen Sie dazu wie folgt vor:

- Mit VS Code Projekt öffnen
- Im Projektordner einen Ordner namens *lib* erstellen
- Gewünschte Jar-Dateien in diesem Ordner ablegen.
- VS Code neustarten

Ein Beispiel der Ordnerstruktur sieht dabei wie folgt aus:



Problembehebungen

Language Server Workspace leeren

VS Code Java verwendet einen versteckten Workspace (*language server workspace*) auf dem System für die kompilierten Dateien der einzelnen Programmen. In manchen Situationen kann es durch Änderungen zu Fehlern kommen (z.B. gewisse Elemente können nicht im Projekt vom Programm gefunden werden, obwohl diese existieren.). Allgemein ist es daher empfehlenswert bei solchen Fehlern den Workspace zuerst zu löschen. Dabei gehen keine wichtigen Dateien verloren. Die Löschung können Sie mit folgenden Schritten durchführen:

- In VS Code die Taste *F1* drücken
- Im Eingabefeld *java clean* eingeben und das Feld *Java: Clean the Java language server workspace* mit Pfeiltasten und Enter auswählen

- Im erschienenen Dialog unten rechts auf *Restart and delete* klicken

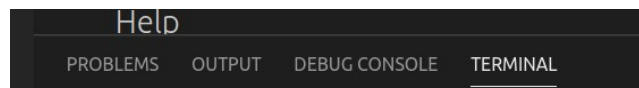
Programme exportieren

In diesem Kapitel finden Sie Anleitungen zum Exportieren von Projekten, so, dass andere Benutzer auf anderen Systemen Ihr Programm ausführen können, ohne dazu VS Code installieren zu müssen. Beachten Sie jedoch, dass diese Varianten für die «vereinfachten» Projekte gedacht sind. Projekte mit anderen Strukturen (z.B. Buildsysteme wie Maven) können Probleme beim Durchführen der Anleitungen verursachen.

Jar-Datei Konsolenvariante

Über die Konsole können Sie einfache Projekte in eine lauffähige Jar-Datei umwandeln. Beachten Sie dabei, dass diese Anleitung von Projekten ohne Unterverzeichnissen ausgeht. Diese Jar-Dateien können bei den meisten Endbenutzer mit installierter Java-Laufzeitumgebung mit einem Doppelklick ausgeführt werden. Der Endbenutzer kann dazu die openJDK-Umgebung installieren (<https://adoptopenjdk.net/>).

Öffnen Sie mit VS Code wie gewohnt ihr Projekt, welches Sie exportieren möchten. Wählen Sie dann den Terminal aus:



Im Terminal können Sie mit folgenden Befehl zuerst alle Java-Dateien in Class-Dateien kompilieren:

```
javac *.java
```

Nach dem Kompilieren können Sie mit folgendem Befehl die erstellten Class-Dateien in eine Jar-Datei zusammenfügen:

```
jar cf MyJar.jar ./
```

Danach sollte sich in Ihrem Projekt-Verzeichnis die erstellte Jar-Datei *MyJar.jar* befinden.