

Java Leitprogramm

Kapitel 1 – Editieren, Compilieren, Ausführen

Übersicht

In diesem Kapitel wird erklärt, wie ein Programm entsteht. Du wirst lernen, wie man aus einem Text ein Programm machen kann.

Du wirst in diesem Kapitel noch nicht lernen, selbst ein Programm zu schreiben. Zunächst geht es nur darum aus einem vorgegebenen Text ein Programm zu machen, was man auf dem Computer laufen lassen kann.



Lernziel

Nachdem du dieses Kapitel durchgearbeitet hast, kannst du einen **vorgegebenen (Programm-)Text in ein Programm umwandeln** und dieses **Programm am Computer starten**.



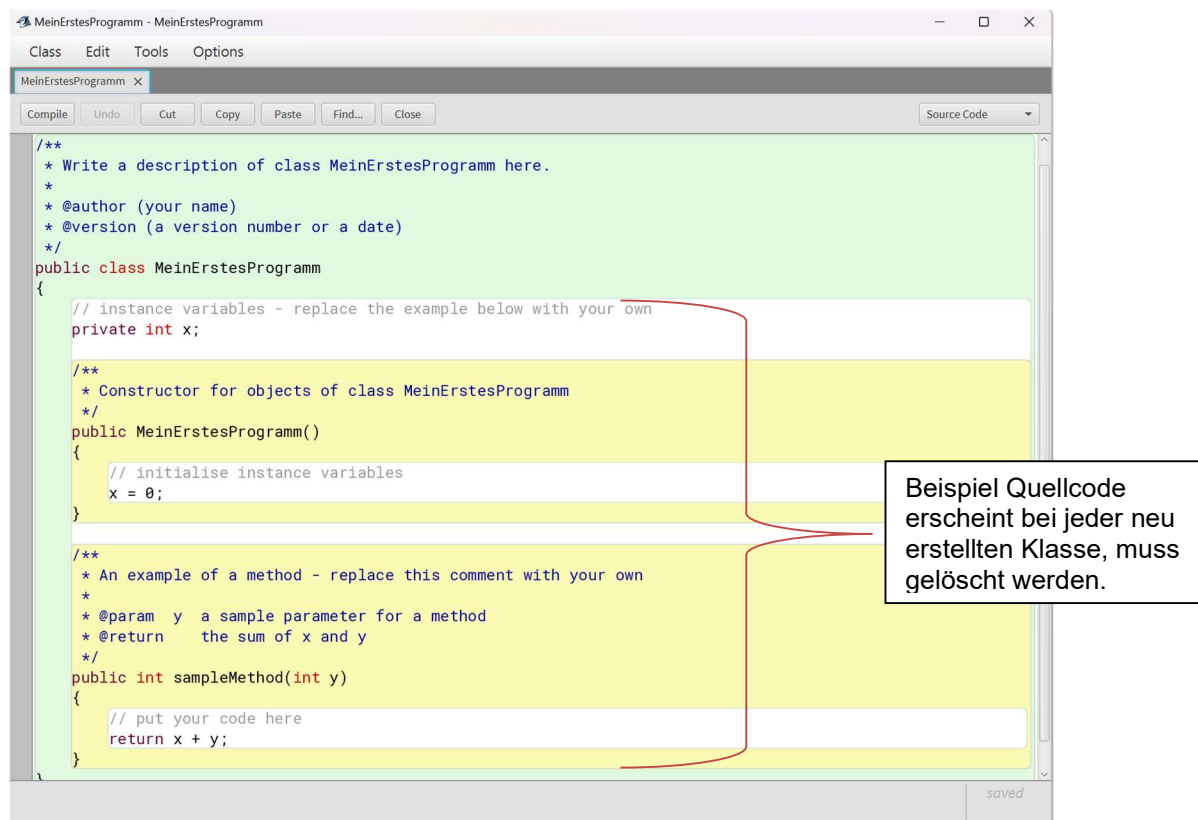
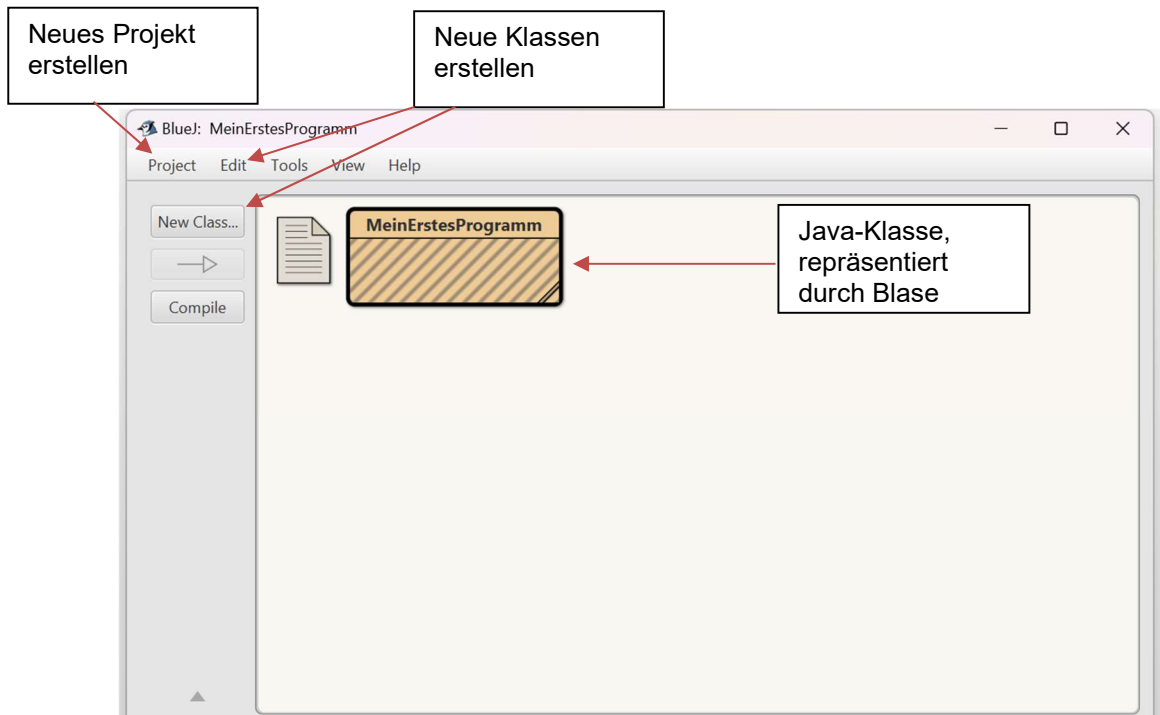
Aufgabe1 - Übungsaufgaben am Computer -

Wir kommen direkt zur Sache. Führe die nächsten Arbeitsaufträge aus! Wenn du unsicher bist, was mit einigen der Anweisungen gemeint sein könnte, dann hilft dir die Abbildung vielleicht weiter.

- Kopiere das Windows-Installer-Paket „BlueJ-windows-530“ aus dem Dateiordner des Kurses bei Teams in deinen persönlichen Ordner oder auf deinen Stick.

Wichtiger Hinweis: Der Pfad zum Installer-Paket darf keine Leerzeichen enthalten!

- Doppelklicke auf den Installer und folge den Anweisungen, um die Installation des Programms BlueJ durchzuführen. Wähle die Option „Install just for you“ aus.
- Öffne anschließend BlueJ 
- **Erstelle ein neues Projekt (Project -> new Project) und nenne es „MeinErstesProgramm“**
- **Erstelle eine neue Klasse mit dem Namen „MeinErstesProgramm“ (Edit -> new Class, oder New Class button)**
- **Doppelklick auf die Klasse „MeinErstesProgramm“,  um den Javaquelltext zu editieren**
- **Lösche den Beispielquellcode in der Klasse**




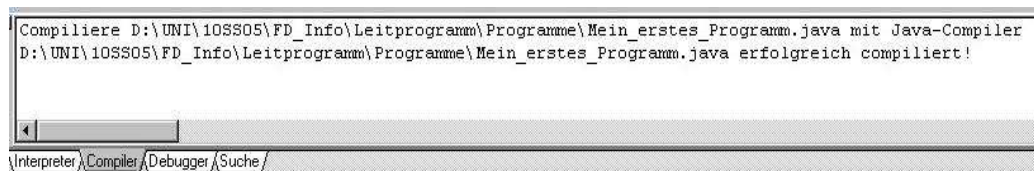
- Gib folgenden Javaprogrammcode ein!


```
public class MeinErstesProgramm {  
  
    public static void main(String[] arguments) {  
        System.out.print("Hallo Welt");  
    }  
  
}
```

- Speichere das Programm unter dem Namen:

Mein_erstes_Programm.java

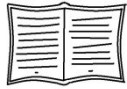
- Klicke auf das Symbol  ! Jetzt hast du das Programm vom Computer **compilieren** lassen. Was das bedeutet, klären wir im Theorieteil.
- Wenn du bis hierhin alles richtig gemacht hast, dann steht im unteren Feld so etwas wie:



- Wenn dieses nicht der Fall ist, dann taucht in dem Feld der Begriff „error“ (also Fehler) auf. Das bedeutet, dass du irgendeinen Teil des Textes nicht richtig übertragen hast. Das kann schnell passieren. Versuche, den Fehler zu finden und **compiliere** dann noch mal!
- Klicke auf das grüne Dreieck  ! Es geht ein schwarzes Fenster auf. In dem Fenster steht (unter anderem):

Hallo Welt!

Du hast das Programm **ausgeführt**. Was das bedeutet, klären wir auch im Theorieteil.



Theorie

Was hast du in der Aufgabe oben gemacht?

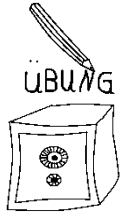
Du hast, vielleicht das erste Mal, ein Programm geschrieben. Dazu hast du einen vorgegebenen Text, den **Quelltext** deines Programms, mit Hilfe des Java-Editors geschrieben und gespeichert. Diesen Teil des Programmierens nennt man **editieren**. Deshalb nennt man Programme, mit denen man Quellcodes schreiben und speichern kann auch **Editoren**.

Im nächsten Schritt hast du das Programm compiliert. Weißt du eigentlich was das bedeutet? Ein Computer versteht zunächst einmal nur sehr einfache und sehr wenige Anweisungen. Wenn man mit diesen wenigen Anweisungen ein größeres Programm schreiben müsste, bräuhete man sehr lange und der Quelltext würde sehr unübersichtlich werden. Aus diesem Grund hat man **höhere Programmiersprachen** entwickelt, die viel mehr Anweisungen bieten. Der Quelltext, den du geschrieben hast, ist in der höheren Programmiersprache JAVA geschrieben. Mit höheren Programmiersprachen ist das Programmieren leichter und weniger aufwändig. JAVA ist die Programmiersprache, die wir in den nächsten Wochen lernen wollen.

Wie erfährt der Computer nun, welche der ihm bekannten, einfachen Anweisungen ausgeführt werden müssen, wenn er einen Befehl in einer höheren Programmiersprache erhält? Für diese Aufgabe gibt es **Compiler**. Sie übersetzen einen Quelltext, der in einer höheren Programmiersprache geschrieben ist, so, dass der Computer ihn versteht.

Nach dem Compilieren weiß der Computer also, was er tun soll. Dein nächster Schritt war in der obigen Aufgabe das **Ausführen des Programms**. Es öffnete sich ein schwarzes Fenster in dem „Hallo Welt“ stand. Dieses schwarze Fenster nennt man **Eingabeaufforderung**. Ein Programm ausführen bedeutet also, dass der Computer die Anweisungen befolgt, die im Quelltext des Programms formuliert sind. Du kannst dein Programm jetzt beliebig oft vom Computer ausführen lassen.

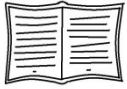
Eine Software, mit der man Quellcodes nicht nur editieren und speichern sondern auch compilieren und ausführen kann, heißt **Entwicklungsumgebung**. JavaEditor ist die Entwicklungsumgebung, mit der wir in Zukunft arbeiten wollen.



Aufgabe 2 - Übungsaufgabe im Heft und Sicherungsphase

Löse das Aufgabenblatt zu Kapitel 1 (am besten, ohne im Theorieteil nachzusehen).

Theorie



Bis hierhin haben wir erstmal nur gelernt, wie man ein vorgegebenes Programm editiert, compiliert und ausführt.

Aber was macht das Programm `Mein_erstes_Programm.java`?

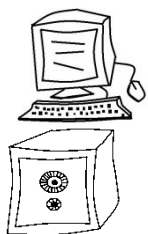
Nun wollen wir uns mit der Bedeutung, der sogenannten **Semantik**, des Quelltext befassen:

```
public class Mein_erstes_Programm {  
  
    public static void main(String[] arguments){  
        System.out.print("Hallo Welt");  
    }  
  
}
```

Die Ausführung dieses Programms bewirkt, dass die Textzeile

Hallo Welt

auf dem Bildschirm angezeigt wird. Man nennt das **Ausgabe** oder **Output**.



Aufgabe 3 - Übungsaufgabe am Computer

Welche Zeile des Programms könnte die Ausgabe bewirkt haben?

Versuche das Programm so umzuschreiben, dass es einen anderen Text als

Hallo Welt

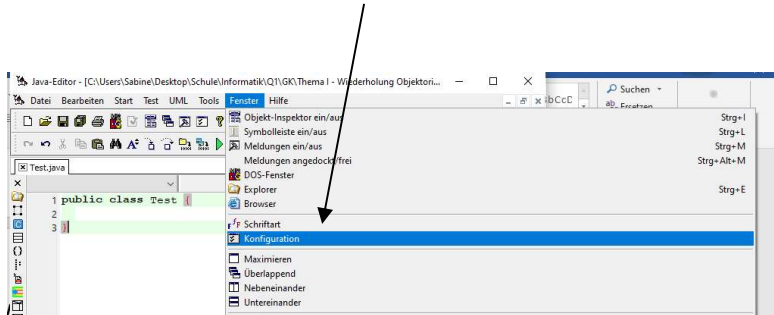
ausgibt!

Was, wenn das Compilieren nicht klappt???

Falls der Javaeditor nicht den richtigen Pfad zum JDK findet, gibt es ein Problem und dein Programm wird nicht compiliert.

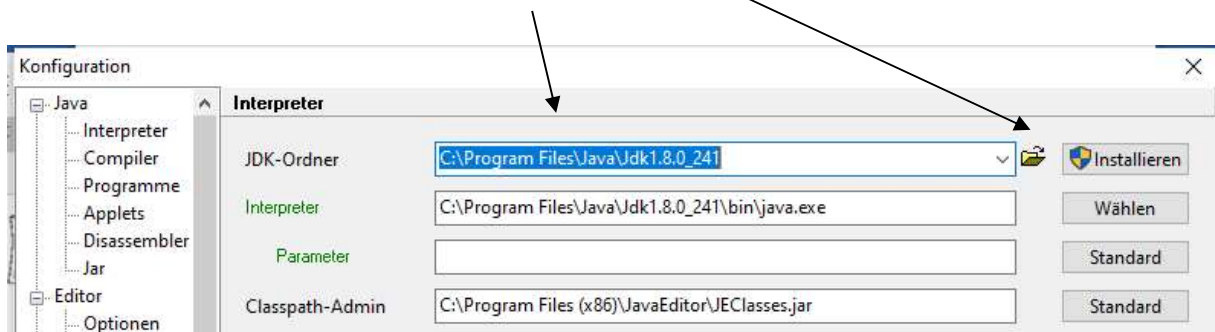
In diesem Fall musst du den Pfad zum JDK einstellen.

Öffne dazu das Konfigurationsfenster:



Sollte das JDK nicht gefunden werden sind die eingestellten Pfade nicht richtig und im Konfigurationsfenster rot unterlegt.

In diesem Fall musst du auf das Ordnersymbol klicken und das Verzeichnis, auf dem sich dein JDK befindet, auswählen, z.B.:



Falls du kein JDK auf deinem PC installiert hast, kannst du auch das JDK aus dem Software-Ordner auf deinem Stick angeben.

Falls der Pfad nun stimmt, verschwindet die Rotfärbung automatisch.

Wähle unten im Konfigurationsfenster den Button *Speichern* – dann sollte es eigentlich funktionieren! (Außer, dass du vielleicht noch Fehler im Programm hast!)