

# Programmierung 1

im Wintersemester 2025/26

## Praktikumsaufgabe 1

### Einstieg in Java (Kap. 1)

Empfohlener Fertigstellungstermin: 28. Oktober 2025 (DFIW) und 29. Oktober 2025 (PI)

Mit diesen ersten Aufgaben steigen Sie in die Welt der Java-Programmierung ein: Gleich in **Aufgabe 1** schreiben Sie Ihre ersten Zeilen Java-Code und in **Aufgabe 2** führen Sie sogar Ihr erstes komplettes Java-Programm aus. In **Aufgabe 3** können Sie dann einen 15-minütigen Crashkurs zu den wichtigsten Grundlagen von Java absolvieren. In **Aufgabe 4** können Sie schließlich Ihre individuelle Entdeckungsreise starten.

Lassen Sie sich nicht von der Länge der Aufgaben abschrecken. Die Aufgaben sind so konzipiert, dass Sie sie in kleinen Schritten lösen können, insbesondere als Einsteiger:in ohne Vorkenntnisse in der Programmierung.<sup>1</sup> Um Ihnen den Einstieg zu erleichtern, sind die einzelnen Schritte ausführlich beschrieben. Dies gilt insbesondere für **Aufgabe 1**. Außerdem erhalten Sie mit **Aufgabe 2** eine Einführung in ein zentrales Entwicklungswerkzeug dieser Veranstaltung. Die **Aufgaben 3** und **4** entsprechen eher dem Detaillierungsgrad, den Sie in den folgenden Praktikumsaufgaben erwarten können.

Lösen Sie die Aufgaben nicht alleine, sondern mindestens zu zweit. Wenn Sie bereits eine Coding-Partner:in haben, dann arbeiten Sie mit dieser zusammen. Wenn nicht, dann tun Sie sich mit anderen Kommiliton:innen zusammen. Es ist wichtig, dass Sie die Aufgaben gemeinsam lösen. Warum? Weil man in der Gruppe mehr Spaß hat, sich gegenseitig unterstützt und dadurch viel besser lernt.

---

<sup>1</sup>Gegen Ende der Vorlesungszeit und damit in Richtung Klausur werden die Aufgabenbeschreibungen deutlich weniger detailliert ausfallen. Der Grund dafür ist, dass die Aufgabenstellungen in der Klausur wesentlich kürzer sein werden als die in dieser und den folgenden Praktikumsaufgaben. So können Sie sich spätestens dann auf die Art der Aufgabenstellung vorbereiten, die Sie in der Klausur erwartet.

Um Ihnen das Verstehen und Lernen zu erleichtern, finden Sie bei jeder Aufgabe die Lernziele, die Sie mit dieser Aufgabe erreichen sollen. Diese Lernziele helfen Ihnen, sich auf das Wesentliche zu fokussieren und so effektiv wie möglich zu lernen. Warum? Weil die Lernziele die Frage beantworten: “*Wozu* soll ich das tun und lernen?” Und da Sie in dieser Lernveranstaltung eigenverantwortlich lernen und machen, können Sie jedes Lernziel abhaken, sobald Sie denken, dass Sie es erreicht haben.

*Tipp:* Um effektiv zu lernen, machen Sie sich immer wieder klar, warum und wofür Sie etwas lernen! Das hilft Ihnen, sich zu motivieren und Ihren Fokus und Ihre Aufmerksamkeit aufrechtzuerhalten.

**Heads up!** Diese erste Praktikumsaufgabe zählt noch nicht zu den Prüfungsvorleistungen. Daher gibt es weder ein offizielles Review mit Tutor:in noch eine offizielle Einreichung Ihrer Reflexionen über das Feedback, das Sie im Rahmen der Reviews erhalten hätten. Die Aufgaben dieser ersten Praktikumsaufgabe sollen Ihnen den Einstieg in die Programmierung mit Java erleichtern und Sie mit dem Entwicklungswerkzeug vertraut machen, das Sie in dieser Veranstaltung immer wieder verwenden werden. Nutzen Sie die Präsenztermine des Praktikums am 21. Oktober (DFIW) bzw. am 22. Oktober (PI), um Ihre Fragen zu klären und sich bei der Lösung der Aufgaben unterstützen zu lassen.

**Wir als Ihre Lernbegleiter:innen unterstützen Sie gerne bei der Lösung der Aufgaben.**

# 1 Ihr erster Java-Code

## Lernziele

- ☐ Sie ändern und erweitern bestehenden Java-Code, um Ihren ersten eigenen Java-Code zu entwickeln.
- ☐ Sie verwenden den Java Playground, um Java-Code schnell und einfach auszuprobieren und auszuführen.
- ☐ Sie führen eine kurze Selbstreflexion durch, um Ihre Metakognition<sup>a</sup> zu verbessern und eine für Sie passende Lernstrategie zu entwickeln.

<sup>a</sup>Metakognition ist das “Denken über das Denken” und spielt eine wichtige Rolle für erfolgreiches Lernen. Es geht darum, das eigene Denken und Lernen bewusst wahrzunehmen, zu steuern und zu reflektieren. Sie lernen nicht nur, “wie” sie etwas tun, sondern auch “warum” und “wie gut”. Das erhöht die Effektivität des Lernens enorm.

Indem Sie ein paar Zeilen *Java-Code* schreiben und ausführen, machen Sie Ihre ersten Schritte als Java-Programmierer:in. Schnappen Sie sich Ihren Laptop, eine Tasse Kaffee oder Tee – und legen Sie los! Sie haben keinen eigenen Computer? Kein Problem, Sie können einen der Arbeitsplätze im STL benutzen. Nur Kaffee oder Tee darf man dort leider nicht trinken. Aber so schlimm ist das ja auch wieder nicht. . .

## 1.1 Den ersten Java-Code schreiben und ausführen

In diesem ersten Schritt passen Sie zwei bestehende Codezeilen an, um Ihren ersten Java-Code zu schreiben. Dazu verwenden Sie die Online-Plattform *Java Playground*.

Der Java Playground ermöglicht die einfache Ausführung von Java-Code, ohne dass ein vollständiges Java-Programm benötigt wird. So können Sie in kürzester Zeit Ihren ersten Java-Code schreiben und ausführen. Genau das werden Sie jetzt tun.

## 1.2 Den Java Playground im Browser öffnen

Öffnen Sie den Java Playground, indem Sie auf den folgenden Link klicken oder die Adresse in Ihrem Browser öffnen: <https://dev.java/playground/>

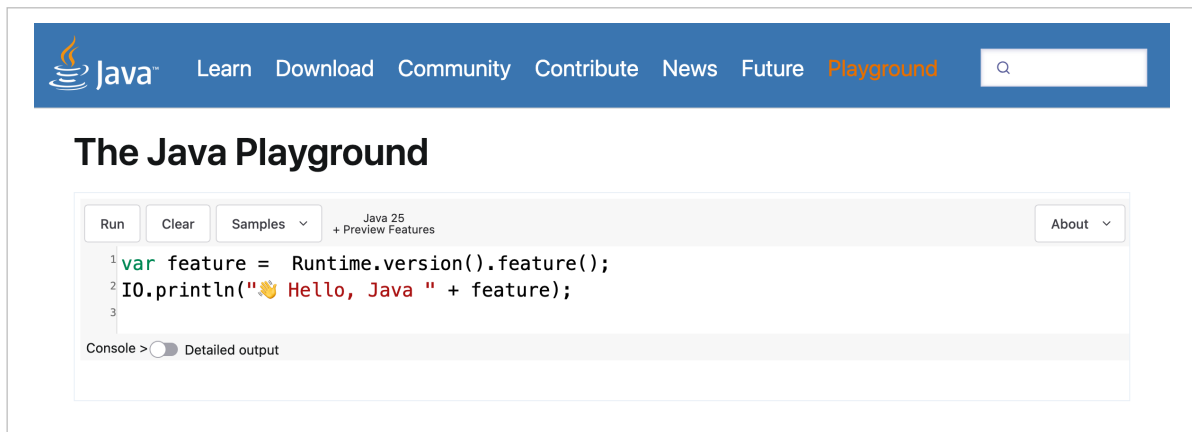


Abbildung 1: Der Java Playground nach dem Öffnen im Browser.

Dort sollten zwei Codezeilen zu sehen sein, die mit den beiden hier zu sehenden bis auf das Waving-Hand-Emoji identisch sind:

```
1 var feature = Runtime.version().feature();
2 IO.println("👋 Hello, Java " + feature);
```

**Abbildung 1** zeigt den Java Playground nach dem Öffnen im Browser. Dort sehen Sie den Java-Code, den Sie in dieser Aufgabe ändern, erweitern und ausführen sollen.

### 1.3 Den ersten Java-Code ausführen

Wenn Sie auf *Run* klicken, wird der Code ausgeführt und das Ergebnis angezeigt. Wie einfach ist das? Sie sollten in der Ausgabe das Emoji sehen, das im Code enthalten ist, gefolgt von dem Text *Hello, Java 25*. Und schon haben Sie Ihren ersten Java-Code ausgeführt. Und weil Ausprobieren und Experimentieren beim Lernen so wichtig sind, können Sie gleich weitermachen. Sie sind dran!

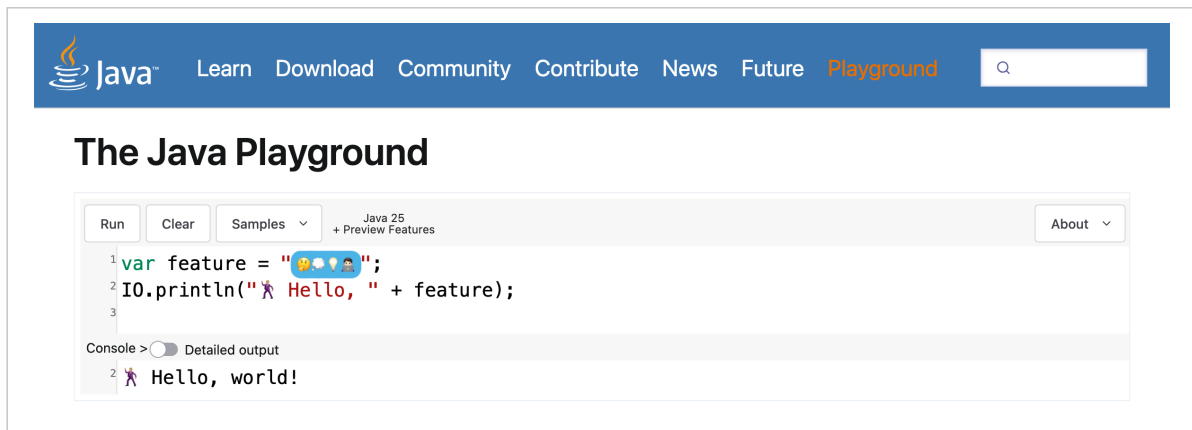


Abbildung 2: Der Java Playground nach dem Ausführen des geänderten Java-Codes.

## 1.4 Die ersten Änderungen an Java-Code machen

Wie könnten Sie den Code so anpassen, dass er stattdessen den Text `Hello, world!` ausgibt? Am besten mit dem Waving-Hand-Emoji!? Oder vielleicht mit einem anderen Emoji Ihrer Wahl? Probieren Sie verschiedene Möglichkeiten aus! Bleiben Sie neugierig und nutzen Sie die Möglichkeit des Java Playground, Ihren Code direkt auszuführen und das Ergebnis sofort zu sehen. So ist schnelles und effektives Lernen garantiert.

In **Abbildung 2** sehen Sie eine beispielhafte Lösung, die zumindest in Teilen zeigt, wie der Java-Code geändert werden kann, um ein anderes Emoji und den Text `Hello, world!` auszugeben.

*Tipp:* Sie können die gewünschte Ausgabe auch erreichen, indem Sie nur die zweite Zeile ändern. Die erste Zeile können Sie dabei unverändert lassen. Das haben Sie aber wahrscheinlich schon selbst durch Ausprobieren herausgefunden. Hurra!

## 1.5 Die ersten Lernerfolge feiern

Haben Sie den Java-Code entsprechend angepasst und ausgeführt? Erscheint die gewünschte Ausgabe im Browser? Herzlichen Glückwunsch! Sie haben Ihren ersten Java-Code geschrie-

ben und ausgeführt. Und das in kürzester Zeit. So macht Lernen Spaß! Und weil effektives Lernen Erfolge feiert: Klopfen Sie sich und Ihrer Coding-Partner:in auf die Schulter! Zelebrieren Sie kleine und große Lernerfolge! Und das Beste: Das ist noch nicht alles. Sie werden noch viele weitere Gelegenheiten haben, Lernerfolge zu feiern. Also: Weitermachen!

## 1.6 Die erste professionelle Verbesserung

Vielleicht stört es Sie jetzt, dass in Ihrem Code `feature` steht, obwohl der Code nichts mehr mit irgendeinem Feature zu tun hat. Als professionelle Programmierer:in sollte Sie dies jedenfalls wirklich stören. Ändern Sie also diesen *Variablennamen*<sup>2</sup> so ab, dass der Code besser lesbar und verständlicher wird! Kandidaten wären z. B. `recipient`, `name` und `greetee`.

Bleiben Sie neugierig und probieren Sie verschiedene Variablennamen aus! Welcher ist Ihr Favorit und warum? Denken Sie dabei daran, dass Sie als professionelle Programmierer:in auch für die Lesbarkeit und Verständlichkeit Ihres Codes durch andere verantwortlich sind. Computerprogramme entwickelt man nicht allein, sondern in Teams. Und Teams bestehen aus Menschen. Menschen, die Ihren Code lesen und verstehen wollen.

*Tipp:* Nun müssen beide Zeilen geändert werden. In beiden Zeilen wird das gleiche “Suchen und Ersetzen” durchgeführt.

## 1.7 Die nächsten Schritte machen

Warum probieren Sie nicht weiter aus? Ändern Sie den Code so ab, dass der Text `Salut le monde !` ausgegeben wird. Dabei sollte der Textteil `le monde` möglichst *variabel* sein. Huh, wie könnte *das* gehen? Überlegen und diskutieren. Dann ausprobieren! So können Sie die schnelle und einfache Ausführung von Java-Code weiter nutzen, um spielerisch zu lernen.

*Tipp:* Das `+`-Zeichen ist der *Operator* für die *Verkettung* von *Strings* in Java. Sie können also zwei Zeichenketten mit `+` zu einer neuen Zeichenkette verbinden.<sup>3</sup>

<sup>2</sup>Wir werden viel erklären, aber nicht alles. Und manches auch nicht sofort. Seien Sie neugierig und gehen Sie auf die Suche nach der **Bedeutung solcher Fachbegriffe!**

<sup>3</sup>Weitere Fachbegriffe, die ihnen bald fließend in den Mund gehen.

Wenn Sie den Code nicht so ändern können, dass der Text `Salut le monde !` auf diese Weise ausgegeben wird, ist das kein Problem. Schauen Sie sich den Crashkurs in **Aufgabe 3** an und kommen Sie dann zurück, um weiterzumachen.

## 1.8 Reflexion Ihrer Lernerfahrung

Nachdem Sie Ihre ersten Schritte in der Java-Programmierung gemacht haben, nehmen Sie sich bitte ein paar Minuten Zeit, um über Ihren Lernprozess nachzudenken. Beantworten Sie die folgenden Fragen schriftlich oder im Gespräch:

### 1. Was habe ich gelernt?

Beschreiben Sie in einigen Sätzen, welche neuen Konzepte oder Techniken Sie gelernt haben. Was hat Ihnen geholfen, diese Konzepte zu verstehen?

### 2. Welche Schwierigkeiten gab es?

Überlegen Sie, ob Sie bei bestimmten Schritten (z. B. beim Ändern oder Ausführen des Codes) auf Schwierigkeiten gestoßen sind. Wo genau sind Sie hängengeblieben? Was hat Sie davon abgehalten, weiterzumachen?

### 3. Wie habe ich diese Schwierigkeiten gelöst? Wenn nicht, wie könnte ich sie beim nächsten Mal lösen?

Überlegen Sie, wie Sie das Problem gelöst haben (z. B. durch Ausprobieren, Nachschlagen, mit Hilfe einer Kommiliton:in oder durch Fragen im **Forum**). Wenn Sie keine Lösung gefunden haben, was könnten Sie beim nächsten Mal anders machen?

#### 1.8.1 Beispielhafte Reflexion

Eine mögliche Reflexion könnte beispielsweise wie folgt aussehen:

1. Ich habe gelernt, dass sich der von einem Programm ausgegebene Text ganz einfach in den Anführungszeichen ändern lässt. Außerdem habe ich verstanden, dass eine Varia-

ble wie ein Behälter für Informationen ist und dass sie an allen Stellen, an denen sie verwendet wird, exakt gleich heißen muss.

2. Bei der Umbenennung der Variable `feature` hatte ich zunächst vergessen, sie auch in der zweiten Zeile (`println`) zu ändern. Das Programm lief dann nicht mehr und ich wusste kurz nicht, warum.
3. Ich habe meinen Code genau mit dem Originalcode aus der Anleitung verglichen und so den Fehler gefunden. Nächstes Mal achte ich darauf, eine Änderung an allen betroffenen Stellen durchzuführen.



## 2 Ihr erstes Java-Programm

### Lernziele

- ☐ Sie führen kleine, vollständige Java-Programme aus, um sich auf das Schreiben und Ausführen von umfangreicherem Code vorzubereiten.
- ☐ Sie verwenden den OneCompiler, um vollständige Java-Programme zu schreiben und diese über den Browser auszuführen.
- ☐ Sie verwenden den OneCompiler, um Java-Programme für die spätere Weiterentwicklung zu speichern.

Nachdem Sie in **Aufgabe 1** mit ein paar Zeilen Java-Code begonnen haben, ist es nun an der Zeit, Ihr erstes vollständiges *Java-Programm* zu schreiben und auszuführen. Bei dieser Gelegenheit lernen Sie auch gleich ein wichtiges Programmierwerkzeug kennen, das Sie in dieser Lernveranstaltung verwenden werden. Und das alles, ohne eine lokale Entwicklungsumgebung einrichten zu müssen. Los geht's!

### 2.1 Das erste Java-Programm schreiben und ausführen

Das Schreiben und Ausführen von Java-Code im Java Playground ist eine großartige Möglichkeit, schnell und einfach ein paar Zeilen Java-Code auszuprobieren. Aber was ist, wenn man ein *komplettes Java-Programm* schreiben und ausführen möchte? Dafür gibt es den *OneCompiler*.

Ein komplettes Java-Programm? Ja, genau! Ein vollständiges Java-Programm besteht aus mindestens einer *Klasse*, die eine *Methode* mit dem Namen `main` enthält. Diese Methode ist der Einstiegspunkt in das Java-Programm. Sie ist sozusagen der Ort, an dem das Java-Programm zu laufen beginnt. Man könnte auch sagen, dass Ihr Programm mit dieser Methode startet. Hier ist ein Beispiel für ein vollständiges Java-Programm:

```
1 public class Main { // The Main class is the "container" for this program.
2
3     public static void main(String[] args) { // The main method is the
        ↪ starting point for running the program.
4         System.out.println("Hello, World!"); // This code was previously run
        ↪ through the Java Playground.
5     }
```

```
6  
7 }
```

Details zu dieser “Anatomie einer Java-Klasse” finden Sie auf Seite 8 des Buches *Java von Kopf bis Fuß*. Machen Sie sich nicht die Mühe, alles im Detail zu verstehen. Diese Aufgabe soll Sie auf den Weg bringen. Wichtig ist an dieser Stelle, *wo* Sie Ihren Java-Code unterbringen, der ausgeführt werden soll: in der Methode `main`.

Der OneCompiler ist wie der Java Playground eine Online-Plattform, auf der Sie Java-Code schreiben und ausführen können. Der Unterschied besteht darin, dass Sie mit dem OneCompiler ein komplettes Java-Programm schreiben und ausführen können. Genau das werden Sie gleich tun.

In Java muss der Code, den Sie schreiben wollen, immer in einer Klasse und in einer Methode stehen, da Java objektorientiert ist. Einzelne Anweisungen wie

```
System.out.println("Hello, World!");
```

können nur funktionieren, wenn sie in den richtigen Kontext eingebettet sind. Der einfachste Kontext ist eine Klasse und die Methode `main`. Die Schaffung dieses Kontexts hat bisher der Java Playground übernommen. Jetzt sind Sie wieder dran!

## 2.2 Den OneCompiler im Browser öffnen

Sie können den OneCompiler öffnen, indem Sie auf den folgenden Link klicken oder die Adresse in Ihrem Browser öffnen: <https://onecompiler.com/java>. Auch hier können Sie den Java-Code auf Knopfdruck ausführen. Probieren Sie es aus! Danach sollten Sie den OneCompiler in Ihrem Browser sehen, wie in **Abbildung 3** dargestellt. Das Beste daran: Sie haben jetzt mehr als “nur” ein paar Zeilen Java-Code ausgeführt. Sie haben ein vollständiges Java-Programm ausgeführt. Zelebrieren Sie das!



Abbildung 3: Der OneCompiler nach Ausführen von “Hello, World!” im Browser.

## 2.3 Die eigenen Schritte machen

An dieser Stelle ist es im Interesse Ihres Erfolges angebracht, Sie weniger eng zu begleiten als bei **Aufgabe 1**. Wiederholen Sie die Schritte, die Sie dort mit dem Java Playground gemacht haben, mit dem OneCompiler:

**Ausführen:** Sie führen Ihren Java-Code aus. Oder besser gesagt, Ihr erstes vollständiges Java-Programm.

**Ändern:** Sie ändern Ihr Java-Programm so, dass der ausgegebene Text variabel ist. Sie wählen einen geeigneten Variablennamen.

**Erweitern:** Sie lassen Ihr Programm `Salut le monde !` ausgeben.

**Lernen:** Lassen Sie Ihrer Fantasie freien Lauf und probieren Sie weitere Änderungen aus. Was passiert, wenn Sie ein Semikolon vergessen? Oder wenn Sie den Variablennamen an einer Stelle falsch schreiben? **Machen** Sie und lernen Sie dabei!

*Tipp:* Viele dieser Schritte können Sie zumindest teilweise lösen, indem Sie Ihren Java-Code aus dem Java Playground in den OneCompiler kopieren. Auch wenn es Ihnen trivial erscheinen mag: Tun Sie es und übertragen Sie Ihr bisheriges Wissen und Können. Das ist ein wichtiger Lernschritt. Beachten Sie dabei, dass der Java Playground den Befehl `IO.println` verwendet, um Text auszugeben. Dieser Befehl ist jedoch nicht in allen Java-Umgebungen bzw. älteren Versionen von Java als der derzeit **neuesten Version 25** verfügbar. Im OneCompiler und in der Vorlesung verwenden wir den “Standardbefehl” `System.out.println`. Dieser Befehl ist in allen Java-Versionen verfügbar. Denken Sie also beim Übertragen Ihres Codes daran, diese Zeile ggf. entsprechend anzupassen!

## 2.4 Den Programm-Code speichern und laden

Wie Sie bereits bei der Entwicklung Ihres Java-Programms gesehen haben, können Sie Ihr Programm schreiben und ausführen, ohne ein Konto bei OneCompiler zu haben. Zum Speichern und Laden benötigen Sie jedoch ein Konto. Auf diese Weise können Sie Ihren Java-Code über mehrere Sitzungen hinweg speichern und laden.

Erstellen Sie ein solches Konto bei OneCompiler und speichern Sie Ihr Java-Programm. So können Sie später darauf zugreifen und weiter daran arbeiten. Dies ist auch ein wichtiger Schritt, um Ihre zukünftigen Lernerfolge festzuhalten und Ihren Code weiterzuentwickeln.

### 2.4.1 Automatisches Speichern

OneCompiler speichert Ihr Java-Programm automatisch, wenn Sie ihn ausführen. Das bedeutet, dass Ihr Code jedes Mal, wenn Sie auf *RUN* klicken, in Ihrem Konto gespeichert wird. Diese Funktion stellt sicher, dass Sie Ihre Arbeit nicht verlieren und später darauf zugreifen können.

*Hinweis:* Bitte beachten Sie, dass Ihr Code nicht gespeichert wird, wenn Sie die Seite neu laden oder schließen, ohne zuvor auf *RUN* geklickt zu haben.

### 2.4.2 Speicheroptionen und manuelles Speichern

Wenn Sie mehr Kontrolle darüber haben möchten, wie Ihr Programmcode gespeichert wird, z. B. dass er nicht öffentlich sichtbar ist, können Sie die Speicheroptionen öffnen und anpassen, indem Sie auf das Bleistiftsymbol klicken. Das Bleistiftsymbol befindet sich im oberen Teil des Compilerfensters. In den Optionen können Sie Ihren Code umbenennen, eine Beschreibung hinzufügen und festlegen, ob Ihr Code öffentlich, privat oder über einen Link zugänglich sein soll. Wenn Sie auf *SAVE* klicken, werden nicht nur die Speicheroptionen, sondern auch Ihr Code gespeichert.

*Hinweis:* Standardmäßig ist die Option *Public (Everyone)* ausgewählt, d. h. Ihr Code ist öffentlich sichtbar. Wenn Sie Ihren Code privat speichern möchten, müssen Sie die Option *Private (Only You)* auswählen. Mit der Option *Unlisted (People with the Link)* können Sie Ihren Code über einen Link teilen.

### 2.4.3 Zugriff auf gespeicherten Code

Sie können später auf Ihren gespeicherten Code zugreifen, indem Sie sich bei OneCompiler anmelden und zu Ihrem Konto gehen (*My account*). Dort finden Sie eine Liste Ihrer gespeicherten Codes (*CODES*).

## 2.5 Bonus: Begrüßen Sie namentlich!

Sie haben die Aufgaben schnell erledigt? Sie sind neugierig und möchten mehr erfahren und lernen? Dann haben wir eine Bonusaufgabe für Sie: Passen Sie Ihr Java-Programm so an, dass es einen Namen akzeptiert und Sie damit begrüßt.

Dies hat auch den Vorteil, dass Sie ein Programm kreieren, das von anderen genutzt werden kann. Viele Programme existieren nicht für sich allein, sondern in Interaktion mit Menschen. Und Menschen haben Namen. Warum also nicht gleich damit anfangen? Schließlich fühle ich mich als Anwender:in einfach wohler, wenn ich von einem Programm mit meinem Namen angesprochen werde.

Verschiedene Eingaben und passende Ausgaben für ein solches Programm könnten sein:

**Martin** Hallo Martin!

**Brian Kernighan** Hello, Brian Kernighan!

**Véronique** Salut Véronique !

*Tipp:* Unter dem eigentlichen OneCompiler finden Sie eine Anleitung zu diesem. Im dortigen Abschnitt *Taking inputs (stdin)* erfahren Sie, wie Sie Ihr Java-Programm dazu bringen, beliebige Eingaben und damit Namen zu akzeptieren.

## 2.6 Tipp: Tutorial, Cheatsheet und Coding Challenge

Neben der Möglichkeit, Java-Code schnell und ohne großen Aufwand im Browser zu schreiben und auszuführen, bietet OneCompiler auch eine **Einführung in Java** (Tutorial), einen **Java-Spickzettel** (Cheatsheet) und **Coding-Rätsel** (Beginners Coding Challenge).

Nutzen Sie diese vielfältigen Möglichkeiten, um z. B. mit Erfahrung in einer anderen Programmiersprache in Java einzusteigen (Tutorial), sich einen Überblick über die wichtigsten Elemente und Konzepte von Java zu verschaffen (Cheatsheet) oder Ihr Wissen mit kleinen Herausforderungen zu testen und zu erweitern (Beginners Coding Challenge).

## 3 Crashkurs

### Strecklernziele

- ☐ Sie verschaffen sich einen kurzen Überblick über die Programmiersprache Java.
- ☐ Sie erkennen Lücken in Ihrem Vorwissen und finden Wege, diese zu schließen.
- ☐ Sie beschäftigen sich mit ergänzenden und alternativen Lern- und Informationsquellen.

Mit dieser Aufgabe verfolgen Sie erstmals *Strecklernziele*, die auch als *Stretch-* oder *erweiterte Lernziele* bezeichnet werden. Diese erweiterten Lernziele gehen über die normalen Lernziele hinaus. Mit dem Erreichen dieser Lernziele können Sie z. B. Ihr Wissen über ein Thema vertiefen und erweitern oder es besser verstehen und lernen.

### 3.1 Learn Java in One Video - 15-minute Crash Course

Sehen Sie sich das Video [Learn Java in One Video - 15-minute Crash Course](#) an. Es gibt Ihnen einen schnellen Überblick über die Programmiersprache Java.

Es geht nicht darum, alles zu verstehen, sondern einen ersten Eindruck von der Sprache Java zu bekommen. Das erleichtert das spätere Verstehen. Aus der Hirnforschung ist bekannt, dass man am besten lernt, wenn neue Informationen mit Bekanntem verknüpft werden können. Daher auch das sogenannte *Priming* an dieser Stelle.

#### 3.1.1 Beobachtungsauftrag

Um den Lerneffekt (das Priming) zu maximieren, erteilen wir Ihnen einen konkreten Beobachtungsauftrag:

- Notieren Sie sich beim Schauen drei Begriffe oder Konzepte, die Ihnen neu oder besonders interessant erscheinen, zum Beispiel *Variable*, *Schleife* oder *Methode*.
- Bewerten Sie an dieser Stelle nicht, ob Sie sie verstehen. Es geht lediglich darum, sie für später im Gedächtnis zu verankern.

### 3.2 Selbsteinschätzung

Wenn Sie beim Anschauen des Videos 15 Minuten lang nur Bahnhof verstehen, könnte das ein Hinweis darauf sein, dass Sie sich intensiver mit den absoluten Grundlagen der Programmierung im Speziellen oder der Informatik im Allgemeinen beschäftigen sollten. Das ist nicht schlimm, sondern ein wichtiger Schritt für ein erfolgreiches Informatikstudium.

Überlegen Sie dann, wie Sie Ihr Vorwissen auffrischen oder erweitern können. Welche Ressourcen stehen Ihnen zur Verfügung? Das kann Material sein, aber auch eine Person, die Ihnen helfen kann. Welche Ressourcen fehlen Ihnen? Wie könnten Sie diese beschaffen?

### 3.3 Lern- und Informationsquellen

Im Moodle-Kurs finden Sie ergänzende und alternative Lern- und Informationsquellen. Insbesondere finden Sie dort eine Liste von Büchern, die Ihnen helfen können, die Grundlagen des Informatikstudiums zu verstehen: *Vorkurs Informatik für Dummies*, *Programmieren lernen für Dummies* und *Java für Dummies*. Auch wenn die Titel nicht gerade schmeichelhaft klingen, sind die Bücher sehr gut und verständlich geschrieben.

Schauen Sie sich die Liste an und überlegen Sie, ob Sie eines der Bücher lesen möchten. Was könnte Ihnen dadurch möglich werden?



## 4 Entdeckungsreise

### Strecklernziele

- ☐ Sie befassen sich mit grundlegenden Begriffen und Konzepten der Programmierung.
- ☐ Sie wenden grundlegende Konzepte der Programmierung in Ihrem eigenen Java-Code an.

Im Video zu **Aufgabe 3** sehen Sie viele wichtige Konzepte und Begriffe, die Sie für die Programmierung in Java benötigen und die wir in den kommenden Wochen ausführlich behandeln werden. Dazu gehören neben Variablen auch primitive Datentypen, Strings, Schleifen, Bedingungen und vieles mehr.

Vieles davon ist so grundlegend, dass es sich auf andere Programmiersprachen wie Python, C++ oder JavaScript übertragen lässt. Während des Studiums und im späteren Berufsleben werden Sie mit verschiedenen Programmiersprachen in Kontakt kommen. Diese Grundlagen gehören also zu Ihrem Handwerkszeug als Programmierer:in.

### 4.1 Reflektieren Sie Grundlagen der Programmierung

Diskutieren Sie im Coding-Team oder mit anderen Kommiliton:innen, was aus dem Video grundlegende Konzepte der Programmierung sein könnten. Die folgenden Fragen können dabei helfen:

- Welche Begriffe und Konzepte sind Ihnen im Video begegnet?
- Welche Begriffe und Konzepte sind Ihnen bereits bekannt?
- Welche Begriffe und Konzepte sind Ihnen neu?
- Welche Begriffe und Konzepte erscheinen Ihnen besonders wichtig?
- Wenn Sie aus der Sicht von Benutzer:innen darüber nachdenken, was Programme typischerweise tun, was könnten Anwendungsfälle für diese Begriffe und Konzepte sein?

## 4.2 Erweitern Sie Ihr Java-Programm

Erweitern Sie Ihr Java-Programm aus **Aufgabe 2** so, dass es mindestens zwei der Konzepte aus dem Video verwendet. Probieren Sie es aus! Wenn Sie sich unsicher sind, fragen Sie Ihre Kommiliton:innen oder Ihre Lernbegleiter:innen um Hilfe. Hier sind einige Vorschläge:

**for-Schleife** Verwenden Sie eine **for**-Schleife, um die Begrüßung mehrmals auszugeben.

**Alter statt Name** Fragen Sie nicht nach dem Namen, sondern nach dem Alter und geben Sie eine entsprechende Ausgabe aus.

**if-Bedingung** Verwenden Sie eine **if**-Bedingung, um eine Ausgabe zu erzeugen, die sich je nach Alter des Benutzers unterscheidet, aber dennoch schmeichelhaft ist.