

Didaktische Hinweise

Einstieg in die Programmierung mit Processing¹

Zielgruppe

Der Leitfaden zum Einstieg in Processing richtet sich vor allem an Schüler*innen in der Einführungsphase, die Einblick in das textbasierte Programmieren erhalten möchten.

Ebenso kann der Leitfaden für Lernende in der Sekundarstufe I eingesetzt werden, die neben der grafischen Programmierung Einblick in eine textbasierte Programmiersprache erhalten möchten.

Voraussetzungen

Wenn die Schüler*innen bereits Vorerfahrungen im Programmieren mit einer grafischen Programmiersprache haben, sollten sie mit dem Leitfaden relativ selbstständig arbeiten können.

Handelt es sich um Anfangsunterricht müssen die verschiedenen algorithmischen Konzepte, insbesondere das Variablenkonzept, begleitend im Unterricht thematisiert werden.

Lernziele

Mithilfe des Leitfadens zum Einstieg in die Programmierung mit Processing sollen die Schüler*innen einen ersten Überblick erhalten, wie sie algorithmische Konzepte, die sie bereits aus den grafischen Programmiersprachen kennen, in Processing umsetzen können. Dazu gehört die Verwendung der Kontrollstrukturen Schleife und Sequenz sowie die Reaktion auf Maus- und Tastaturereignisse.

Didaktische Hinweise

Die einfachen Möglichkeiten zur Erzeugung von grafischen Ausgaben bietet einen relativ angenehmen, anschaulichen Einstieg in die textbasierte Programmierung. Durch die grafischen Ausgaben erhalten die Schüler*innen eine direkte visuelle Rückmeldung, ob ihr Algorithmus korrekt arbeitet. Ein Processing Programm ist dabei deutlich übersichtlicher aufgebaut als beispielsweise ein Java-Applet, das vergleichbare visuelle Ausgaben erzeugt.

Mithilfe von Processing haben die Schüler*innen daher die Möglichkeit, sich den Einblick in die textbasierte Programmierung an anschaulichen Beispielen zu erarbeiten. Außerdem erlernen sie die Syntax von Java ohne sich gleich mit allen Konzepten, die beispielsweise ein Java-Applet beinhaltet, auseinandersetzen zu müssen.

Es ist auch denkbar diesen Leitfaden parallel zu den Materialien für den Einstieg in die Programmierung mit *Snap!* einzusetzen, um Vorlieben und unterschiedliche Wissensstände der Schüler*innen zu berücksichtigen. Dabei wird transparent, dass die algorithmischen Konzepte imperativen Programmierens unabhängig von der Programmiersprache gültig sind.

Je nachdem wie sehr die (textbasierte) Programmierung im Unterricht vertieft werden soll, können Teile des Leitfadens, insbesondere zur Interaktion mit dem Anwender, auch ausgelassen bzw. explizit den besonders leistungsstarken Schüler*innen angeboten werden. Verzichtet werden kann beispiels-

¹ Die Programmierumgebung Processing wurde 2001 von Ben Fry und Casey Reas initiiert. Nähere Informationen finden Sie unter <https://processing.org/>

weise auf die Simulation von Buttons. Die entsprechende Aufgabe 14 ist daher mit einem Stern gekennzeichnet.

Da sich Processing besonders für die Programmierung von Algorithmen zur Manipulation von Bildern eignet, bietet sich Processing als Werkzeug besonders dann an, wenn die Verarbeitung von Bildern im weiteren Verlauf des Unterrichts noch eine Rolle spielen soll.

Die Aufgaben

Der Leitfaden zum Einstieg in Processing ist so aufgebaut, dass verschiedene algorithmische Konzepte zunächst an kleinen Beispielen und Übungsaufgaben erlernt werden. Diese sollen die Lernenden anschließend in einem größeren Projekt kombinieren, in das sie auch eigene Ideen einfließen lassen können und sollen. Dabei können die prozessbezogenen Kompetenzen aus dem Bereich Kreatives Schaffen und Problemlösen besonders gefördert werden². Es werden drei Vorschläge für mögliche Projekte gemacht, die von den Schüler*innen individuell ausgestaltet werden können. Alternativ sind aber natürlich auch gänzlich eigene Ideen der Schüler*innen möglich. Um die Projekte der anschließend objektiv vergleichen und bewerten zu können, bietet es sich an, vorher einen Kriterienkatalog zu erstellen. Dieser gibt den Schüler*innen einen Überblick, welche algorithmischen Kompetenzen sie im Rahmen ihres Projektes nachweisen sollen. Der Katalog kann zum Beispiel folgende Kompetenzen enthalten:

- zielgerichteter Einsatz von Verzweigungen und Schleifen
- zielgerichtete Verwendung von Variablen
- Verwendung von Zufallszahlen
- Reaktion auf Ereignisse, die von der Maus ausgelöst werden
- Simulation eines Buttons
- Reaktion auf Tastendruck auf der Tastatur
- *alternativ zu den letzten drei Punkten allgemein:* Interaktion mit dem Anwender

Zu den Aufgaben gibt es Lösungsvorschläge. Wobei beachtet werden sollte, dass es sich dabei immer nur um eine mögliche Lösung handelt und die Schüler*innen natürlich auch auf andere gleichwertige Lösungen kommen können.

Für die korrekte Ausführbarkeit wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte für die Musterlösungen oder Beispiele des Leitfadens oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

² vgl. Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck