

Didaktische Hinweise

Einstieg in die Programmierung mit Processing

Zielgruppe

Der Leitfaden zum Einstieg in Processing¹ richtet sich vor allem an Schülerinnen und Schüler in der Einführungsphase, die einen Einblick in das textbasierte Programmieren erhalten möchten.

Ebenso kann der Leitfaden für besonders interessierte Lernende in der Sekundarstufe I eingesetzt werden, die über die blockbasierte Programmierung im Pflichtfach Informatik hinaus Einblick in eine textbasierte Programmiersprache erhalten möchten.

Voraussetzungen

Es ist hilfreich, wenn die Schülerinnen und Schüler bereits erste Erfahrungen im Umgang mit algorithmischen Kontrollstrukturen und Variablen gesammelt haben, beispielsweise im Rahmen einer Einheit zum algorithmischen Problemlösen mit Scratch² im Pflichtfachunterricht Informatik in Jg.9/10. Je nach Vorkenntnissen der Schülerinnen und Schüler müssen Fachbegriffe gegebenenfalls über den Leitfaden hinaus vertiefend erläutert werden.

Lernziele

Mithilfe des Leitfadens zum Einstieg in die Programmierung mit Processing erhalten die Schülerinnen und Schüler einen ersten Überblick, wie sie algorithmische Konzepte, die sie ggf. bereits aus blockbasierten Programmiersprachen kennen, in Processing umsetzen können. Dazu gehört die Verwendung von algorithmischen Kontrollstrukturen und Variablen sowie die Reaktion auf Maus- und Tastaturereignisse. Je nachdem, welche Programmiersprachen die Lernenden bereits kennengelernt haben, ist es sinnvoll, auf Gemeinsamkeiten hinzuweisen, damit entsprechende Strukturen als Konzepte und nicht als werkzeugspezifisch wahrgenommen werden.

Processing kann auch in der Qualifikationsphase bis hin zum Abitur genutzt werden. Da Processing eine auf Java basierende Programmierumgebung ist, können die hier erlernte Syntax ebenso wie die Konzepte aber auch dann als Grundlage für die Qualifikationsphase dienen, wenn später ein Umstieg von Processing auf eine vollwertige Java-Entwicklungsumgebung erfolgen soll.

Hinweise zum Leitfaden

Der Leitfaden zum Einstieg in Processing ist so aufgebaut, dass die Aufgaben zu Beginn eines Kapitels häufig zum Erkunden neuer Konzepte einladen. Auch die Beispiele können von den Lernenden ausprobiert und erläutert werden. Es bietet sich daher ggf. an, den Schülerinnen und Schülern die entsprechenden Aufgaben oder Beispiele zunächst einzeln zur Verfügung zu stellen und diese nach einer Erkundungsphase mit der Lerngruppe zu besprechen. Der Leitfaden enthält dann weitere Aufgaben zum Üben und Festigen der verschiedenen algorithmischen Konzepte.

Der Leitfaden zum Einstieg in das Programmieren mit Processing teilt sich in zwei Teile. In Teil 1 werden am Beispiel des Erzeugens von Grafiken und Animationen grundlegende algorithmische

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

² Scratch ist ein Projekt der Scratch Foundation in Zusammenarbeit mit der Lifelong Kindergarten Group des MIT Media Lab. Es ist kostenlos unter <https://scratch.mit.edu> erhältlich.

Konzepte und ihre Umsetzung in Processing erarbeitet, während es in Teil 2 um die Nutzerinteraktion geht.

Teil 1 umfasst im Einzelnen folgende Aspekte:

Erste Grafiken können mithilfe von Sequenzen aus Zeichenbefehlen erzeugt werden. Dabei ist zu beachten, dass das Koordinatensystem in Processing der Ursprung in der oberen linken Ecke des Programmfensters hat und die y-Achse von dort in die positive Richtung nach unten verläuft.

Im Rahmen des Erzeugens von Animationen werden das Aufteilen eines Programms in die `setup`- und die `draw`-Methode sowie der Umgang mit Variablen und Verzweigungen thematisiert. Mit der Einführung der Methoden `setup` und `draw`, müssen die Lernenden einen Perspektivwechsel im Umgang mit Methoden vornehmen. Bis dahin haben sie vordefinierte Methoden, vor allem zum Zeichnen von geometrischen Figuren, aufgerufen. Für die Methoden `setup` und `draw` ist hingegen im Hintergrund festgelegt, wann diese ausgeführt werden. Die Lernenden legen hingegen den Inhalt der Methoden fest. Dieses Konzept wird auch später bei der Nutzerinteraktion fortgeführt.

Die algorithmische Kontrollstruktur Schleife wird schließlich im Zusammenhang mit Zufallsgrafiken und Mustern eingeführt. Schleifen werden hier vor allem benötigt, um Figuren wiederholt zu zeichnen, bis eine bestimmte Anzahl erreicht ist. Dazu eignen sich sowohl eine *while* als auch eine *for*-Schleife. Beide Varianten werden thematisiert, so dass die Lernenden entscheiden können, welche Variante sie bevorzugen.

Abschließend werden in Teil 1 Figuren abwechselnd mit unterschiedlichen Eigenschaften gezeichnet. Dazu werden mithilfe des Modulo-Operators gerade und ungerade Werte der Zählvariablen unterschieden. Dieses Konzept kann optional von leistungstärkeren Schülerinnen und Schülern erarbeitet werden.

In Teil 2 werden die Processing-Programme um die Möglichkeit der Nutzerinteraktion erweitert. Dazu werden zum einen Systemvariablen, in denen Processing beispielsweise die aktuelle Position des Mauszeigers oder die zuletzt gedrückte Taste zur Verfügung stellt, genutzt. Zum anderen werden neben `setup` und `draw` weitere vordefinierte Methoden im Processing-Programm ergänzt, um auf Ereignisse, die von der Maus oder der Tastatur ausgelöst werden, zu reagieren. Sowohl Systemvariablen als auch ein ähnliches Konzept der Ereignissteuerung kennen die Schülerinnen und Schüler ggf. bereits aus Scratch. Die Systemvariablen entsprechen in Scratch den impliziten Variablen.

Für die Maus werden verschiedene Ereignisse wie beispielsweise Geklickt, Gezogen oder Losgelassen unterschieden. Für die Tastatur ist im Wesentlichen das Ereignis, dass eine Taste gedrückt wurde, relevant. Hier wird anhand der verwendeten Methode unterschieden, ob auf jeden Tastendruck reagiert wird oder Steuerungstasten wie z. B. die Shift-Taste ignoriert werden. Das Ereignis des Loslassens einer Taste wird im Leitfaden nicht thematisiert.

Das Reagieren auf ein Mausereignis durch Definition einer entsprechenden Ereignismethode wird anhand verschiedener Aufgaben geübt. Da das Konzept auf Tastaturereignisse übertragbar ist, stehen hier weniger Übungsaufgaben zur Verfügung. Teil 2 schließt mit einigen Projektvorschlägen ab. Hier können die Lernenden die bislang erlernten Konzepte in einem etwas komplexeren Programm kombinieren und je nach Erforderlichkeit vertiefen. Dabei kann optional auch auf das am Ende des Leitfadens angebotene Anzeigen eines Dialogfensters für Nutzereingaben zurückgegriffen

werden. Dieses ist hier aber optional zu sehen und kann auch später im Zusammenhang mit der Zeichenkettenverarbeitung thematisiert werden.

Im Rahmen der Projekte können die prozessbezogenen Kompetenzen aus dem Bereich Kreatives Schaffen und Problemlösen besonders gefördert werden³ (s. [2]). Es werden drei Vorschläge für mögliche Projekte gemacht, die von den Schülerinnen und Schülern individuell ausgestaltet werden können. Alternativ sind auch gänzlich eigene Ideen der Schülerinnen und Schüler möglich. Hier sollte den Lernenden frühzeitig eine Einschätzung gegeben werden, ob die Idee mit den erlernten Konzepten umsetzbar ist. Um die Projekte anschließend objektiv vergleichen und bewerten zu können, bietet es sich an, vorher einen Kriterienkatalog zu erstellen. Dieser gibt den Schülerinnen und Schülern einen Überblick, welche algorithmischen Kompetenzen sie im Rahmen ihres Projektes nachweisen sollen. Der Katalog kann zum Beispiel folgende Kriterien enthalten:

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

Zu den Aufgaben stehen für Lehrkräfte Lösungsvorschläge zur Verfügung. Dabei ist zu beachten, dass es sich immer nur um eine mögliche Lösung handelt und die Lernenden in der Regel auch andere gleichwertige Lösungen entwickeln werden.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#).

Die Quelltexte und Leitfäden sind lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#).

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

³ [1] Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck