

## Exemplarischer Unterrichtsverlauf zum Thema „Algorithmisches Problemlösen“ im Rahmen eines Pflichtfaches Informatik

– angelehnt an den veröffentlichten schulformübergreifenden Vorschlag für eine Themenauswahl und -reihenfolge für das Pflichtfach Informatik

### Vorbemerkungen

Im Folgenden wird exemplarisch ein möglicher Unterrichtsverlauf zur Vertiefung des Algorithmischen Problemlösens unter Verwendung der infsi-Materialien im Rahmen eines Pflichtfaches Informatik skizziert. Dieser orientiert sich am „*Schulformübergreifenden Vorschlag für eine Themenauswahl und -reihenfolge für das Pflichtfach Informatik*“. Insbesondere wird dabei davon ausgegangen, dass die Lernenden bereits erste Vorkenntnisse im Algorithmischen Problemlösen mit Scratch besitzen, wie sie beispielsweise im exemplarischen Unterrichtsverlauf zum Thema „Einstieg in das Algorithmische Problemlösen“ vermittelt wurden. Beim Algorithmischen Problemlösen in dieser Sequenz geht es jetzt darum, an die gemachten Vorerfahrungen anzuknüpfen. Der experimentelle Zugang wird weiterhin gefördert, erfolgt jedoch zunehmend zielgerichteter. Spätestens jetzt werden typische Bestandteile von algorithmischen Abläufen benannt und auch Variablen in einfachen Algorithmen verwendet.

Je nach Vorerfahrungen einer Lerngruppe müssen individuelle Anpassungen vorgenommen. Vorschläge hierzu werden im Dokument [L\\_Ueberblick\\_Materialien\\_gesamt](#) skizziert. Ziel der folgenden Überlegungen ist lediglich, beispielhaft einen möglichen Unterrichtsverlauf zu skizzieren.

### Ideen zur Sicherung und Vertiefung

Im exemplarischen Unterrichtsverlauf sind Anregungen für mögliche Stundeneinstiege und Erarbeitungsphasen enthalten. Die Arbeitsergebnisse können bei Bedarf während oder am Ende einer Unterrichtsstunde auf verschiedene Arten gesichert und vertieft werden. Ziel dabei sollte sein, eine Unterrichtskultur zu etablieren, in der Fehler als Lernchance begriffen werden. Daher bietet es sich beispielsweise an:

- Zwischenergebnisse zu präsentieren, die noch nicht ganz fertig sind und wo möglicherweise Fragen zur bisherigen Lösung besprochen werden können
- fehlerhafte Programme zu zeigen und gemeinsam nach Lösungen zu suchen

Beim Präsentieren fertiger Schüler:innenlösungen können allgemeine Strategien reflektiert oder die Interpretation von Programmen geübt werden:

- Wo gab es Schwierigkeiten bei der Implementierung? Wie seid ihr da vorgegangen/ auf eure Lösung gekommen?





- Wo müssen Änderungen am präsentierten Programm vorgenommen werden, um ... zu erreichen?

Möglicherweise haben Lernende in einer Unterrichtseinheit zuvor bereits Variablen in ihren Programmen verwendet, ohne dass diese konkret thematisiert wurden. Hier bietet es sich an, diese Programme erneut präsentieren zu lassen und die dortige Verwendung von Variablen gemeinsam zu untersuchen. Oder Schülerinnen und Schüler hatten damals bereits den Wunsch, ihre Programme so zu erweitern, dass Variablen nötig gewesen wären. Auch solche Beispiele bieten gute Gesprächsanlässe.

Ziel beim algorithmischen Problemlösen in der Sekundarstufe I ist weiterhin vorrangig die Förderung der Problemlösefähigkeiten und Stärkung der Selbstwirksamkeit. Häufig gibt es zu einer Problemstellung viele Lösungen auf unterschiedlichen algorithmischen Niveaus. Dabei können Lernende stolz sein, eine Lösung für ein Problem gefunden zu haben, unabhängig vom Anspruch des verwendeten Algorithmus. Hier ist geschicktes pädagogisches Handeln, beispielsweise bei der Festlegung der Reihenfolge von Präsentationen sowie beim individuellen Feedback, gefragt, um die Wertschätzung jeder individuellen Lösung zu verdeutlichen.



## Beispielhafter Unterrichtsverlauf Jahrgang 10 / Algorithmisches Problemlösen

Thema	Kompetenzen Die Schüler:innen...	Inhalte oder möglicher Ablauf	Materialien
<p>ca. 2 Doppelstunden: Entwicklung eines Gruppenprojektes – z.B. Implementierung eines interaktiven Vorfreudekalenders</p>	<ul style="list-style-type: none"> <li>entwickeln und implementieren einen Algorithmus in einer grafischen Programmiersprache auf experimentelle Weise</li> </ul>	<p>Zum Wieder-Einstieg in das algorithmische Problemlösen bieten sich ein arbeitsteiliges Projekt ähnlich dem „Interaktiven Adventskalender“ an, da die Implementierung der einzelnen, voneinander unabhängigen Objekte (beispielsweise der „Türchen“) offen und damit selbstdifferenzierend ist und die Entwicklung eines eigenen Algorithmus auf experimentelle Weise im Vordergrund stehen kann.</p> <p>Falls keine Realisierung in einer ganzen Lerngruppe gewünscht oder möglich ist, kann ein ähnliches Projekt auch in kleineren Gruppen durchgeführt werden. Ideen wären zum Beispiel die arbeitsteilige Vorstellung:</p> <ul style="list-style-type: none"> <li>verschiedener Berufe (je Beruf eine Figur auf der Bühne),</li> <li>verschiedener Sportarten</li> <li>verschiedener Musikinstrumente</li> <li>chemischer Elemente</li> <li>historischer Persönlichkeiten</li> <li>...</li> </ul>	<p>Interaktiver Adventskalender; bei alternativen Gruppenprojekten die im Lehrermaterial dazu enthaltenen Überlegungen zu Vorwissen und Export/Import von Objekten beachten</p>

		ggf. muss das Zusammenfügen aller Einzelprodukte zu einem Gesamtprojekt durch ausgewählte Lernende, die besonders schnell waren, oder durch die Lehrkraft erfolgen	
ca. 1 Doppelstunden: Präsentation des fertigen Gruppenprojektes und anschließende Zusammenfassung elementarer Kontrollstrukturen	<ul style="list-style-type: none"> <li>benennen Anweisung, Sequenz, Schleife und Verzweigung als elementare Kontrollstrukturen</li> </ul> <p>je nach Übung/Vertiefung u.U. zusätzlich eine der Kompetenzen:</p> <ul style="list-style-type: none"> <li>benennen und beschreiben Handlungsabläufe aus ihrer Lebenswelt</li> <li>beschreiben die Auswirkungen von Informatiksystemen auf die Gesellschaft</li> <li>benennen die Interessen, die bei der Ausgestaltung von Informatiksystemen eine Rolle spielen</li> </ul>	<p>z.B.</p> <ul style="list-style-type: none"> <li>Präsentation des Gruppenprojektes und Zusammenfassung der darin verwendeten Kontrollstrukturen;</li> </ul> <p>Mögliche Vertiefung/Übung z.B.</p> <ul style="list-style-type: none"> <li>Zeigen eines fertigen Scratch-Programms und Interpretation durch Lernende sowie Benennung der verwendeten Kontrollstrukturen</li> <li>AB Reflexion_Algorithmen</li> <li>eines der bisher nicht verwendeten kurzen Beispiele aus dem Vorschlag Jg. 9, z.B. <ul style="list-style-type: none"> <li>Farberkenner</li> <li>Vorlesetrainer</li> </ul> </li> </ul>	<p>Reflexion_Algorithmen oder Reflexion_Algorithmen_Variablen (deutlich komplexer, für stärkere SuS)</p> <p>Farberkenner</p> <p>Vorlesetrainer</p>
ca. 2 Doppelstunden: Werte in Variablen speichern	<ul style="list-style-type: none"> <li>verwenden Variablen und Wertzuweisungen in einfachen Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>Einstieg beispielsweise über fertiges Scratch-Programm mit Countdown (siehe Einstiegsbeispiel_Countdown), zu dem die Lernenden Vermutungen bezüglich der Funktionalität sammeln können</li> <li>Erarbeitung weiterer Eigenschaften von Variablen beispielsweise anhand des Arbeitsblattes „Einstiegsbeispiel_Countdown“, im UG sollte zuvor gezeigt werden, wie in</li> </ul>	<p>Einstiegsbeispiel_Countdown</p> <p>Hintergrundwissen_Variablen</p>

	<ul style="list-style-type: none"> <li>• beschreiben die Auswirkungen von Informatiksystemen auf die Gesellschaft</li> <li>• benennen die Interessen, die bei der Ausgestaltung von Informatiksystemen eine Rolle spielen</li> </ul>	<p>Scratch neue Variablen angelegt werden können.</p> <ul style="list-style-type: none"> <li>• Übung/Vertiefung der Verwendung von Variablen z.B. mithilfe                         <ul style="list-style-type: none"> <li>○ Quizmoderator</li> <li>○ Kassenautomat</li> <li>○ Captcha-Creator</li> </ul> </li> </ul>	<p>Quizmoderator                      Kassenautomat                      Captcha-Creator</p>
<p>ca. 3                      Doppelstunden:                      Dein eigenes                      Projekt</p>	<ul style="list-style-type: none"> <li>• entwerfen einen Algorithmus unter zielgerichteter Verwendung der elementaren Kontrollstrukturen</li> <li>• überprüfen, ob eine Implementierung die Problemstellung löst</li> </ul>	<p>Arbeit an eigenem Projekt.:</p> <ul style="list-style-type: none"> <li>• Festlegung eines gemeinsamen Oberthemas</li> <li>• Etablierung einer Routine für Projektarbeit (z.B. gegenseitige Unterstützung in der Lerngruppe, StandUp zu Beginn und Besprechung des aktuellen Arbeitsstandes, Sicherung von Zwischenergebnissen,...)</li> </ul>	

## Förderung von Kompetenzen aus dem Bereich Informatik und Gesellschaft

Viele Fragestellungen des dargestellten exemplarischen Unterrichtsverlaufs bieten die Chance, neben dem Algorithmischen Problemlösen auch Kompetenzen aus dem Bereich „Informatik und Gesellschaft“ zu fördern und Diskussionen beispielsweise zu Auswirkungen von Informatiksystemen auf die Gesellschaft oder die Interessen bei der Ausgestaltung von Informatiksystemen anzuregen. Im Unterrichtsverlauf sollte daher darauf geachtet werden, diese Chancen zu nutzen. Beispiele sind etwa der Farberkennung (Wo werten Computerprogramme automatisiert Farben aus? Welche Chancen und Risiken sind mit der automatisierten Bildanalyse verbunden? ...), der Vorlesetrainer (Kann man über die Lautstärke Personen voneinander unterscheiden oder die Stimmung einer Person analysieren? Wird eine solche Software in der Praxis tatsächlich eingesetzt? Was sind die Gründe dafür? Was sind mögliche Risiken?) oder der Captcha-Creator (Warum kann ein Mensch ein Captcha eher entziffern als eine Maschine? Gibt es auch Nachteile bei der Verwendung von Captchas?...). Auch eigene Projektideen der Lernenden bieten häufig die Gelegenheit zur Reflexion im Kontext „Informatik und Gesellschaft“.

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

Alle Abbildungen von Scratch-Bausteinen und -Objekten sind lizenziert unter einer [Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#).

Scratch wurde entwickelt von der Lifelong Kindergarten Group, MIT Media Lab,  
<http://scratch.mit.edu>

