

Exemplarischer Unterrichtsverlauf zum Thema „Einstieg in das Algorithmische Problemlösen“ im Rahmen eines Pflichtfaches Informatik

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

Im Folgenden wird exemplarisch ein möglicher Unterrichtsverlauf zum Einstieg in das Algorithmische Problemlösen 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 erstmalig Informatikunterricht erhalten und noch keine Vorkenntnisse vorliegen.

Je nach Vorerfahrungen einer Lerngruppe müssen individuelle Anpassungen vorgenommen werden. 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?

Ziel beim algorithmischen Problemlösen in der Sekundarstufe I ist 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.

| | | | |
|--|---|--|---|
| <p>ca. 2 Doppelstunden: Algorithmische Grundstrukturen</p> | <p>entwickeln einen Algorithmus in Scratch auf experimentelle Weise, indem sie die Funktionalität von Schleifen, Ereignissen und Verzweigungen an Beispielen erkunden (ggf. je nach Anpassung des Leittextes ergänzt um weitere Aspekte)</p> | <ul style="list-style-type: none"> • Erarbeitung wesentlicher Grundstrukturen mithilfe eines Leittextes in individuellem Lerntempo • komplette Besprechung im Plenum nicht unbedingt nötig, eher Thematisierung von beobachteten Fehlvorstellungen, Schwierigkeiten etc. • ggf. „Programmieren im Team“: dabei auf regelmäßiges Wechseln der Rollen (wer implementiert und wer hilft) achten | <p>an Lerngruppe angepasster Leittext O2_Programmieren_in_Scratch (vgl. Vorschläge zu individuellen Anpassungen in L1_Ueberblick_Materialien)</p> |
| <p>ca. 3 Doppelstunden (Variante mit 6 DSt siehe nächste Zeile): je nach Lerngruppe Arbeit an eigenem Projekt oder Unterteilung in mehrere, voneinander unabhängige Unterrichtsstunden mit verschiedenen Schwerpunkten</p> | <ul style="list-style-type: none"> • entwickeln und implementieren einen Algorithmus in Scratch auf experimentelle Weise • beschreiben einen gegebenen Algorithmus in ihren eigenen Worten • überprüfen, ob eine Implementierung die Problemstellung löst <p>je nach Beispiel (z.B. ScratchTranslator, Zugangskontrolle und Co) zusätzlich:</p> <ul style="list-style-type: none"> • beschreiben die Auswirkungen von Informatiksystemen auf die Gesellschaft • benennen die Interessen, die bei der Ausgestaltung von | <p>Arbeit an eigenem Projekt:</p> <ul style="list-style-type: none"> • Festlegung eines gemeinsamen Oberthemas • Etablierung einer Routine für Projektarbeit (z.B. gegenseitige Unterstützung in der Lerngruppe, StandUp zu Beginn und Besprechung des aktuellen Arbeitsstandes, Sicherung von Zwischenergebnissen,...) <p>Ideen für Schwerpunkte bei Unterteilung in mehrere Unterrichtsstunden:</p> <ul style="list-style-type: none"> • Vertiefe deine Programmierkenntnisse und wähle: Interaktionsgeschichte/ Bienenschwarm/ ScratchTranslator/... (ca. 1 Doppelstunde) • Zugangskontrolle und Co beim Smartphone (ca. 1 Doppelstunde); • Simulation Rasenmäher (ca. 1 Doppelstunde) | <p>Interaktionsgeschichte Bienenschwarm Scratch-Translator PinNachbauen Simulation Rasenmäher</p> |

| | Informatiksystemen eine Rolle spielen | | |
|---|---|--|---|
| <p>Variante, falls die 3 DSt des Vorschlags zum Calliope entfallen und stattdessen dem Alg. Problemlösen mit Scratch zugeordnet werden: 6 DSt</p> | <ul style="list-style-type: none"> entwickeln und implementieren einen Algorithmus in Scratch auf experimentelle Weise beschreiben einen gegebenen Algorithmus in ihren eigenen Worten überprüfen, ob eine Implementierung die Problemstellung löst <p>je nach Beispiel (z.B. Farberkenner, Vorlesetrainer, ScratchTranslator, Zugangskontrolle und Co) zusätzlich:</p> <ul style="list-style-type: none"> beschreiben die Auswirkungen von Informatiksystemen auf die Gesellschaft benennen die Interessen, die bei der Ausgestaltung von Informatiksystemen eine Rolle spielen | <p>zunächst ein oder mehrere Unterrichtsstunden mit versch. Schwerpunkten, z.B.</p> <ul style="list-style-type: none"> Farberkenner (ca. 1 Doppelstunde) Vorlesetrainer (ca. 1 Doppelstunde) Zugangskontrolle und Co beim Smartphone (ca. 1 Doppelstunde); <p>Gut wäre, wenn hier an mindestens einem Beispiel Erfahrungen gesammelt werden können, dass Eingaben nicht nur Texteingaben sein müssen, sondern auch Sensorwerte (wie die Lautstärke oder der „Farbsensorwert“) sein können</p> <p>Anschließend beispielsweise Arbeit an eigenem Projekt, s.o.:</p> <ul style="list-style-type: none"> Festlegung eines gemeinsamen Oberthemas Etablierung einer Routine für Projektarbeit (z.B. gegenseitige Unterstützung in der Lerngruppe, StandUp zu Beginn und Besprechung des aktuellen Arbeitsstandes, Sicherung von Zwischenergebnissen,...) <p>oder weitere Unterteilung in mehrere, voneinander unabhängige Unterrichtsstunden, s.o.</p> | <p>Farberkenner Vorlesetrainer PinNachbauen</p> <p>s.o.: Interaktionsgeschichte, Bienenschwarm, Scratch-Translator, Simulation Rasenmäher</p> |

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? ...) oder 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?). 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>