

Strukturmodell *Programmieren*

Stand: 26.11.2020

Allgemeine Hinweise zum Umgang mit den Strukturmodellen

Die Strukturmodelle dienen als Anregung und Hilfe für die Unterrichtsvorbereitung, indem sie Phasen eines möglichen Unterrichtsverlaufs darstellen. Alle Strukturmodelle enthalten fünf **übergeordnete Phasen**. Jede dieser fünf Phasen enthält eine oder mehrere **Teilphasen**, die sich in den verschiedenen Strukturmodellen unterscheiden und die Unterschiedlichkeit des Unterrichtsaufbaus in den Fächern und/oder Lernbereichen hervorheben. Die beschriebenen Teilphasen stellen eine Auswahl dar. Welche Teilphasen im Unterricht umgesetzt werden, entscheidet die Lehrkraft.

Die erste Seite des Dokuments gibt einen Überblick über den fünfschrittigen Unterrichtsaufbau inklusive der enthaltenen Teilphasen.

Für jede der fünf Unterrichtsphasen werden auf den folgenden Seiten **didaktische und pädagogische Hinweise** gegeben sowie **Ideen zur Umsetzung** aufgezeigt, die sich in besonderer Weise für die aufgezeigte (Teil-)Phase eignen.

Am Ende des Dokuments wird der mögliche Verlauf einer gesamten Unterrichtsstunde zu einem Unterrichtsthema dargestellt. Dabei liegt das Augenmerk nicht auf den verwendeten Methoden, sondern auf dem Aufbau der gesamten Unterrichtsstunde im entsprechenden Fach bzw. Lernbereich.

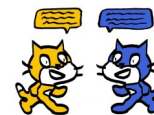
Hinweise zum Umgang mit dem vorliegenden Strukturmodell

Im Strukturmodell **Programmieren** werden Anregungen zu folgenden Unterrichtsthemen gegeben:

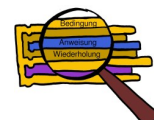
- **Coding mit Scratch 3.0**



- **Einfache Dialoge mit Scratch 3.0 programmieren**



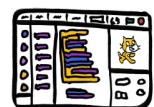
- **Übung: Vertiefung und Übung der algorithmischen Grundstrukturen mit Scratch 3.0**



- **Programmieren auf der Plattform Code.org**



- **Kennenlernen der Oberfläche von Scratch 3.0**

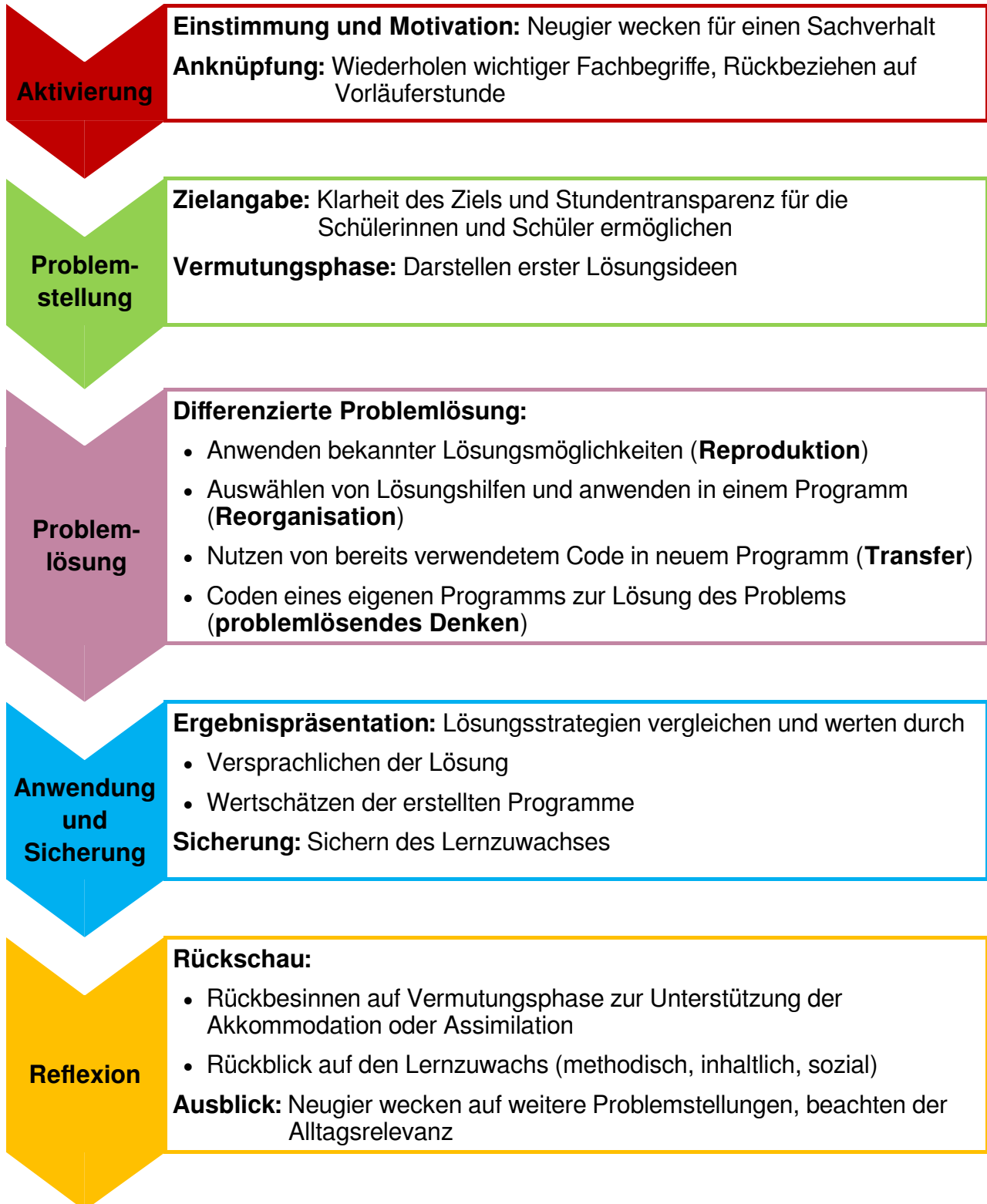


- **Wir entwickeln unseren eigenen Algorithmus**



Strukturmodell *Programmieren*

Stand: 26.11.2020



Aktivierung

Allgemeine Hinweise:

Einstimmung und Motivation

Interesse und Neugier wecken durch

- visuellen Impuls: Video auf einer Medienplattform (z. B. Mebis) oder eine Grafik zeigen
- das Vorstellen eines Lernspiels oder eines Scratch-Programms
- das Zeigen eines programmierbaren Roboters, z. B. Bee-Bots

Anknüpfung

- Wiederholen von Fachbegriffen, die für die Unterrichtsstunde wichtig sind (z. B. auch visualisiert mit grafischem Material)
- Inhalte der vorherigen Stunde wiederholen, um davon ausgehend weiterarbeiten zu können

Ideen zur Umsetzung:

Einstimmung und Motivation

- Die Schülerinnen und Schüler sehen ein von der Lehrkraft mit [Scratch 3.0](#) programmiertes Programm.
- Sie haben nur in die Ausgabe des Programms Einblick, nicht aber in den Programmcode.
- Die Lehrkraft stellt einen Fehler im generierten Programm dar und lässt diesen von den Schülerinnen und Schüler analysieren.

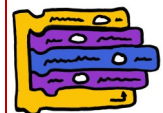
Anknüpfung

- Mögliche Beispiele:
 - Sammeln von Vorwissen mit Hilfe eines Feedback-Tools (z. B. oncoo, Mentimeter) *oder*
 - Kurze Abfrage mit Plickers oder LearningsApps *oder*
 - Analoge Fragerunde: wöchentlich erstellte Karteikarten mit Fragen und Antworten, die von den Schülerinnen und Schülern erstellt und laufend ergänzt werden *oder*
 - Sammeln von Wortkarten an der Tafel



Einstimmung und Motivation

- Die Lehrkraft zeigt den Schülerinnen und Schülern das Bild eines Muffins.
 - ➔ Assoziationen der Schülerinnen und Schülern unkommentiert sammeln (auch visuell)
- Impuls der Lehrkraft: „Du kannst mir sicher sagen, was ich bei einem Rezept beachten muss, damit ich am Ende einen leckeren Muffin bekomme.“
 - ➔ Schülerinnen und Schülern erschließen, dass das Rezept Schritt für Schritt abgearbeitet werden muss.



Problemstellung

Allgemeine Hinweise:

Zielangabe

- Wenn möglich: Formulieren eines Ziels durch die Schülerinnen und Schüler zur Klärung der Fragestellung und Zielfokussierung
- Visualisieren der Zielangabe, z. B. an der Tafel (Sichtbarkeit der Zielangabe über die gesamte Unterrichtsstunde gewährleisten)

Vermutungsphase

- Möglichkeit für Schülerinnen und Schüler, ihr Vorwissen einzubringen
- Fördern des kreativen Denkens (thinking outside the box)
- Alle Vermutungen zulassen und schriftlich (analog oder digital) fixieren, um in der Reflexionsphase darauf zurück zu kommen

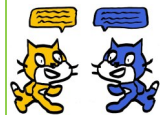
Ideen zur Umsetzung:

Zielangabe

- „Wir schreiben eine Geschichte mit Scratch weiter“

Vermutungsphase

- Die Schülerinnen und Schüler stellen Vermutungen über den Fortgang der Geschichte auf.
- Mit Hilfe des Tools [Mentimeter](#) werden zunächst in Kleingruppen und später im Plenum Ideen gesammelt, zusammengefasst und für die Sicherungsphase abgespeichert.
- Im Anschluss stellen die Schülerinnen und Schüler ihre Vermutungen im Plenum vor.
- Die Lehrkraft moderiert diesen Prozess und achtet darauf, dass alle Schülerinnen und Schüler ihre Vermutungen äußern können.

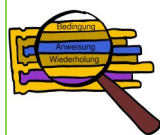


Zielangabe

- „Wir üben Programmieren mit dem Programm Scratch“

Vermutungsphase

- Die Lehrkraft zeigt den Schülerinnen und Schülern ein fertiges Programm, in dem die Bausteine „Anweisung“, „Sequenz“ und „Wiederholung“ verwendet werden.
- Die Lehrkraft hat einen Fehler bei der Wiederholung der Sequenz eingebaut.
- In einer Flüsterphase (2 Minuten) überlegen die Schülerinnen und Schüler zunächst, welches Problem vorliegen könnte.
- In einem zweiten Schritt werden die Vermutungen an der Tafel festgehalten (digital oder analog mit Wortkarten).
- Auf die Vermutungen wird am Ende der Stunde nochmals eingegangen.



Problemlösung

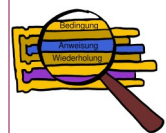
Allgemeine Hinweise:

Differenzierte Problemlösung

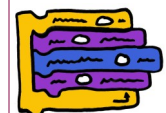
- **Reproduktion:** Den Schülerinnen und Schülern ihre Selbstwirksamkeit erlebbar machen durch
 - das Bereitstellen von 1:1 Anleitungen (auch digital, z. B. durch Lernvideos)
 - das Ermöglichen, bereits erlernte Lösungsmöglichkeiten anzuwenden
- **Reorganisation:** Mehrere Unterstützungsszenarien (analog und digital) anbieten, um die Schülerinnen und Schüler zur selbständigen Auswahl eines Hilfsmittels und der damit verbundenen (weitestgehend eigenständigen) Lösung einer Programmierung zu befähigen.
- **Transfer:** Einsatz eines Unterstützungsnetzwerks (Lehrkraft, Mitschülerinnen und Mitschüler), um bereits Teile verwendeten Codes in einem neuem Programm zu nutzen.
- **Problemlösendes Denken:** Im Team (auch peer-to-peer-teaching möglich) zur Lösung kommen und dabei auf das bereits erworbene Wissen sowie die bereits erworbenen Kompetenzen zurückgreifen (pair coding – immer zwei Schülerinnen und Schüler programmieren gemeinsam).

Ideen zur Umsetzung:

- **Reproduktion**
 - Die Schülerinnen und Schüler programmieren unter Zuhilfenahme eines [Lernvideos](#) mit [Scratch 3.0](#) ein Programm.
- **Reorganisation**
 - Die Schülerinnen und Schüler nutzen Hilfskarten zur Bewältigung eines Problems in [Scratch 3.0](#). Sie wenden darauf beschriebene Lösungsideen und -hilfen bei der Problemlösung an.
 - Sie setzen Handlungsanweisungen unter Verwendung vorgefertigter Befehlsblöcke zusammen und verändern diese so, dass auch eine andere Zusammenstellung zum Ziel führt.
- **Transfer**
 - Die Schülerinnen und Schüler übertragen ihr Wissen vom analogen Programm (im Rollenspiel zwei Menschroboter interagieren lassen) in ein digitales Programm ([Scratch 3.0](#)).
- **Problemlösendes Denken**
 - Die Schülerinnen und Schüler unterstützen ihre Mitschülerinnen und Mitschüler im Sinne des peer-to-peer teachings, um entstandene Probleme bei dem programmierten Dialog zwischen zwei Scratch Figures zu lösen (Einsatz von Befehlsblöcken, Pseudocode, Hilfskarten, etc.).
 - Die Lehrkraft kann unterstützend eingreifen und Lösungsmöglichkeiten moderierend einbringen.



- **Reorganisation**
 - Die Lehrkraft zeigt den Schülerinnen und Schülern ein auf dem Boden aufgeklebtes 5x5 Gitternetz (Seitenlänge der Quadrante ca. 40 cm).
 - Die Schülerinnen und Schülern überlegen in Kleingruppen einen Programmcode, mit dem sie ihre Lehrkraft programmieren und von einem Startfeld zu einem Zielfeld schicken:
 - ➔ Kurze Vorstellung der Aufgabe
 - ➔ „Schreiben“ eines Codes mit Befehlskärtchen
 - ➔ Testung, Analyse und Korrektur des Programms
 - ➔ Die Lehrkraft moderiert und unterstützt bei Schwierigkeiten



Anwendung und Sicherung

Allgemeine Hinweise:

Ergebnispräsentation

- Lösungsstrategien vergleichen und dabei die entsprechenden Fachbegriffe einfordern (Fachbegriffe auf Plakat)
- Gemeinsam Kriterien der Präsentation erarbeiten bzw. wiederholen
- Den eigenen Zugewinn bei der Arbeit im Team feststellen

Sicherung

- Begriffsbilden der informatischen Grundbegriffe durch Zuordnen von Wortkarten zu den entsprechenden Icons
- Festhalten der neuen Begriffe in einer (digitalen) Begriffskartei
 - um diese in einer Hausaufgabe eigenständig zu wiederholen und zu vertiefen
 - die als Freiarbeitsmaterial zum selbstständigen (und gezielten) Üben benutzt werden kann

Ideen zur Umsetzung:

Ergebnispräsentation

- Die Lehrkraft zeigt einige ausgewählte, von den Schülerinnen und Schülern bearbeitete Übungen von Code.org an der Tafel.
- Die Kleingruppen stellen ihre Lösung und ihren Lösungsweg genau vor.
- Die Mitschülerinnen und Mitschüler ergänzen alternative Lösungsvorschläge.

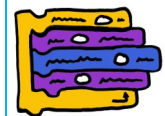
Sicherung

- Die Schülerinnen und Schüler führen ein von der Lehrkraft erstelltes, individuelles Lernspiel auf [LearningApps](#) aus.
- Neue Begriffe der Stunde werden vom Begriffskartei-Dienst am Ende der Stunde vorgelesen und mit Hilfe der anderen Schülerinnen und Schüler ergänzt.



Ergebnispräsentation

- Die Kleingruppen stellen ihre Programmierung bzw. ihr Programm vor und geben sich gegenseitig Rückmeldung.



Reflexion

Allgemeine Hinweise:

Rückschau

- Rückbesinnen auf Vermutungsphase, indem die eigene Vermutung überprüft wird und das neue Wissen mit dem neu erworbenen verknüpft wird (durch Akkomodation und/oder Assimilation)
- Versprachlichen der eigenen Stärken und Entwicklungsfelder und diese bewusst wahrnehmen, z. B. durch
 - Zielscheibe zur Visualisierung der eigenen Programmierkompetenzen, z. B. digital auf [oncoo](#)
 - Wortwolke (Wordcloud), z. B. digital auf [Mentimeter](#)
- Den Schülerinnen und Schülern den Umgang mit konstruktiver Kritik vermitteln und den Fehler als notwendigen Bestandteil des Programmierens und eines Programmierprozesses verinnerlichen und verstehen

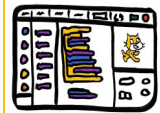
Ausblick

- Neugier wecken auf weitere Problemstellungen
- Beachten der Alltagsrelevanz

Ideen zur Umsetzung:

Rückschau

- Die Schülerinnen und Schüler notieren in einer *Wordcloud* auf [Mentimeter](#) welche weiteren Oberflächen sie von Scratch kennen. Die Lehrkraft macht zunächst ein Beispiel.
- Die Lehrkraft stellt abschließend eine *Skalierungsfrage* auf einer Skala von 1-4: „Wie gut hat die Programmierung im Team funktioniert?“ Die Schülerinnen und Schüler geben digital Rückmeldung.



Ausblick

- Die Lehrkraft zeigt zum Abschluss eine PowerPoint-Präsentation mit Screenshots eines Programms mit der neuen Problemstellung.

Rückschau

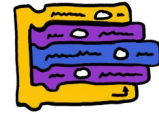
- Die SuS nennen Begriffe aus der Begriffskartei, die für sie heute besonders wichtig waren und begründen ihre Auswahl mit einem kurzen Satz. Dafür wird die Begriffskartei zwischen den Schülerinnen und Schülern weitergegeben.



Ausblick

- Die Lehrkraft gibt den Schülerinnen und Schülern einen ersten Einblick in die nächste Lektion des Übungskurses auf [code.org](#). Zusätzlich gibt sie ihnen die Möglichkeit zu sagen, auf was sie sich in der nächsten Informatikstunde am meisten freuen.

Unterrichtsbeispiel *Wir entwickeln unseren eigenen Algorithmus*



Phase	Unterrichtsaktivität
Einstimmung und Motivation	<p>Die Lehrkraft zeigt den Schülerinnen und Schülern das Bild eines Muffins.</p> <ul style="list-style-type: none"> → Assoziationen der Schülerinnen und Schülern unkommentiert sammeln (auch visuell) <p>Impuls der Lehrkraft: „Du kannst mir sicher sagen, was ich bei einem Rezept beachten muss, damit ich am Ende einen leckeren Muffin bekomme.“</p> <ul style="list-style-type: none"> → Schülerinnen und Schülern erschließen, dass das Rezept Schritt für Schritt abgearbeitet werden muss
Problemstellung	<p>Die Lehrkraft zeigt den Film Muffin Code.</p> <ul style="list-style-type: none"> → Schülerinnen und Schülern erhalten Einblick in die algorithmische Denkweise → Reaktionen der Schülerinnen und Schülern sammeln (Partnerarbeit): Sammlung im Plenum → Problem benennen, analysieren und eingrenzen <p>Zielangabe: „Wir entwickeln ein eigenes Programm“</p>
Problemlösung	<p>Die Lehrkraft zeigt den Schülerinnen und Schülern ein auf dem Boden aufgeklebtes 5x5 Gitternetz (Seitenlänge der Quadrate ca. 40 cm).</p> <p>Die Schülerinnen und Schülern überlegen in Kleingruppen einen Programmcode, mit dem sie ihre Lehrkraft programmieren und von einem Startfeld zu einem Zielfeld schicken.</p> <ul style="list-style-type: none"> → Kurze Vorstellung der Aufgabe → „Schreiben“ eines Codes mit Befehlskärtchen → Testung, Analyse und Korrektur des Programms → Die Lehrkraft moderiert und unterstützt bei Schwierigkeiten
Anwendung und Sicherung	<p>Die Kleingruppen stellen ihre Programmierung bzw. ihr Programm vor und geben sich gegenseitig Rückmeldung.</p>
Reflexion	<p>Reflexion mit digitalem Tool (Mentimeter) sowohl inhaltlich, sozial und methodisch: thematischer Bezug, Zusammenarbeit in der Gruppe, Aufgabenbewältigung</p>

In den ergänzenden Informationen wird auf externe Webangebote hingewiesen, die aufgrund ihres Inhalts pädagogisch wertvoll erscheinen. Wir bitten jedoch um Verständnis, dass eine umfassende und insbesondere eine laufende Überprüfung der Angebote unsererseits nicht möglich ist. Vor einem etwaigen Unterrichtseinsatz hat die Lehrkraft das Angebot in eigener Verantwortung zu prüfen und ggf. Rücksprache mit der Schulleitung zu halten. Sofern das Angebot Werbung enthält, ist die Schulleitung stets einzubinden zwecks Erteilung einer Ausnahme vom schulischen Werbeverbot nach Art. 84 Abs. 1 Satz 2 BayEUG, § 2 Abs. 2 Satz 1 Nr. 4 BaySchO.