



Algorithmen (3.1.1.2)

Std.	Inhaltsbezogene Kompetenzen	Inhalt / Material
1+2	Wiederholung Klasse 7	Wiederholung Grundbegriffe aus Klasse 7 (Kontrollstrukturen, Variable): kahoot-Quiz „Algorithmen in Scratch“ 00_alg_kahoot_wiederholung_klasse7 Auftrag zur Wiederholung in Scratch: Video (movie_einstieg), AB (01_alg_auftrag_wiederholung_klasse7)
3+4	Wiederholung Variablen (2) Zufallszahlen in eigenen Programmen verwenden	Froschhüpfen – Teil I: Aktion mit Auftrag (s. Unterricht) Froschsprungprotokoll (02_alg_frosch_auftrag_1_protokoll) Programmierauftrag (02_alg_frosch_auftrag_1 ¹)
5-8	(3) eine indexbasierte Datenstruktur zur Speicherung und Verarbeitung gleichartiger Daten (auch per Iteration) verwenden (4) grundlegende <i>Algorithmen</i> auf einer indexbasierten Datenstruktur (zum Beispiel Füllen mit Werten (s.o.), Maximumsuche, Summenbildung) erläutern und implementieren <u>PbK:</u> <i>2.11 (4) gleichartige Daten in geeigneten Datenstrukturen zusammenfassen</i>	Froschhüpfen – Teil II: Aktion mit Auftrag (02_alg_frosch_auftrag_2_aktion) Beispielprotokoll (s. Lösungen) Arbeitsblatt zu Listen (02_alg_frosch_auftrag_2_listen) Programmierauftrag (mit Maximum und Mittelwert) (02_alg_frosch_auftrag_2)
9-14	(1) Logische Verknüpfungen (UND, ODER, NICHT) in Bedingungen von <i>Schleifen</i> und <i>Verzweigungen</i> anwenden	„Einarmiger Bandit“ Auftrag und Information Aussagen und log. Verknüpfungen (03_alg_bandit_vorbereitung_aussage)

1 Die Dateien finden Sie in den Kopiervorlagen. Je nach Programmierumgebung ist an den Dateinamen „_scratch“ oder „_ai2“ angehängt. Sie finden die Versionen in einem Unterordner.



	<p>(2) Zufallszahlen in eigenen Programmen verwenden (s.o.)</p> <p>(5) Unterprogramme verwenden, um Programmcode zu strukturieren und redundanten Code zu vermeiden</p> <p>(6) Anforderungen an Programme oder Programmteile beschreiben, dazu geeignete Testfälle entwerfen und die Implementierungen dagegen testen</p> <p>PbK:</p> <p><i>2.11 (6) Teillösungen zur Lösung des Gesamtproblems nutzen</i></p> <p><i>2.12 (9) Abläufe in einer (z.B. grafischen) Programmiersprache implementieren</i></p>	<p>Auftrag – Teil I (03_alg_bandit_auftrag_1)</p> <p>Auftrag – Teil II (mit Unterprogramm) (03_alg_bandit_auftrag_2_unterpr...)</p> <p>Übungen zu Unterprogrammen (03_alg_uebungen_unterprogramm)</p> <p>Protokoll fürs Testen des Projektes (03_alg_bandit_testfaelle_protokoll)</p> <p>Übungen zu logischen Verknüpfungen (03_alg_bandit_log_verknuepfungen_uebungen)</p>
<p>15-28</p>	<p>(6) Anforderungen an Programme oder Programmteile beschreiben, dazu geeignete Testfälle entwerfen und die Implementierungen dagegen testen</p> <p>(7) ein kleines Softwareprojekt (zum Beispiel Spiel, Smartphone-App, Robotik) unter Anleitung durchführen</p> <p>PbK:</p> <p><i>2.12 (4) charakteristische und verallgemeinerbare Bestandteile herausarbeiten (Abstraktion)</i></p> <p><i>(5) relevante Abläufe, Daten und ihre Beziehungen in informatischen Modellen darstellen</i></p> <p><i>(11) Programme gezielt gegen vorab formulierte Testfälle testen</i></p> <p>zum Projekt:</p> <p><i>2.11, 2.12, 2.13, 2.14</i></p>	<p>Softwareprojekt im Zweierteam</p> <p>Entwurf, Planung, Vorstellung</p> <p>Vorlage (04_alg_projekt_vorlage)</p> <p>jede Stunde Start mit Lagebesprechung samt Vorstellung der Vorhaben für die aktuelle Stunde</p> <p>an geeigneten Punkten Festlegung und Durchführung von möglichem Testen</p> <p>Protokoll fürs Testen des Projektes (04_alg_projekt_testfaelle_protokoll)</p> <p>immer wieder Präsentation der Zwischenergebnisse der Teams</p>

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>.
Scratch ist lizenziert unter [CC BY-SA 2.0](https://creativecommons.org/licenses/by-sa/2.0/deed.en) (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).