

Das Präsenz-Seminar „Kids can code“ im Überblick

Vollständiger Titel:	Kids can code - Programmieren als Verbindung von Medienbildung und Digitalisierung in der Grundschule und Sek I
Beteiligte Institutionen:	Abteilung Didaktik der Informatik (Department für Informatik), Bildungswissenschaften (Professionalisierungsbereich Lehrkräftebildung)
Curriculare Einbindung:	Modul <i>Medienbildung und Digitalisierung</i>
(Online/ Präsenz/ Hybrid) Veranstaltungstyp:	(Präsenz) Seminar, als Wahlpflichtseminar mit zugehöriger Pflicht-Vorlesung
Anzahl der Dozierenden:	1
Dauer:	1 Semester
Format und Dauer:	wöchentlich, 2 SWS
Zahl der Teilnehmenden:	10 – 16
Zielgruppe:	Lehramtsstudierende aller Fächer und Schultypen im Master
ECTS:	3 KP (+ 3 KP Vorlesung)
Prüfungsleistung:	Didaktische Aufbereitung einer Technologie einschließlich Diskussion + verschriftlichtes Konzept zum Einsatz der Umgebung in der Schule
Durchführung:	Wintersemester 2020/21 bis Wintersemester 2021/22

1. Seminarkonzept

Das Seminar verfolgt das Ziel, angehenden Lehrkräften Einblicke in die Programmierung zu bieten, um diese Seite der Funktionsweise digitaler Medien offen zu legen. Das Seminar richtet sich an Lehramtsstudierende für Grundschulen und Sekundarstufe I und soll ihnen Möglichkeiten aufzeigen, Informatik als kreativ-schaffenden Prozess zu begreifen und in das eigene Fach zu integrieren.

Begleitet wird das Seminar von der Vorlesung „Medienbildung und Digitalisierung“, in der theoretische Grundlagen vermittelt werden. Es werden Grundbegriffe der Medienbildung, Medienpädagogik und Informatik definiert und gegeneinander abgegrenzt, grundlegende Theorien im Kontext des Lernens und Lehrens mit und über digitale Medien vermittelt. Teilnehmende erhalten einen Überblick über die historische Entwicklung und aktuelle Trends im Bereich der Bildungstechnologien und ein Verständnis von der mediatisierten Lebenswelt und dem Mediennutzungsverhalten von Kindern und Jugendlichen. Im Seminar wird ein Bezug zur Theorie aus der Vorlesung hergestellt, der Hauptfokus liegt jedoch auf der Programmierung als ein Element informatischer Bildung. Dazu werden verschiedene Programmierumgebungen für Kinder und Jugendliche praktisch erprobt und deren potentieller Einsatz in der Schule außerhalb der Informatik konzipiert, diskutiert und reflektiert.

Das Seminar besteht aus zwei Teilen: im ersten Teil werden verschiedene Technologien und Programmierumgebungen wie BeeBot, Calliope mini, Ozobot, Scratch, TurtleStitch, BlockSCAD, etc. vorgestellt. Dazu arbeitet sich jeweils eine Gruppe Studierender in die Technologie ein und bereitet sie für ihre Kommiliton:innen so auf, dass diese in 30 Minuten im Seminar einen aktiven Einblick erhalten und eine kleine hands-on Aufgabe selbst erledigen. Anschließend werden Eindrücke geteilt und die

Technologie/Programmierungsumgebung und ihre Einsatzmöglichkeiten in der Schule in verschiedenen Fächern und Schulstufen diskutiert. Im zweiten Teil der Veranstaltung liegt der Fokus auf dem konkreten Einsatz der Technologie/ Programmierungsumgebung in der Schule. Die Studierenden erarbeiten in ihren Kleingruppen Konzepte, die sie im Seminar mit den Kommiliton:innen erproben und zur Diskussion stellen.

Die Prüfungsleistung im Seminar besteht aus diesen beiden Teilen: der didaktischen Aufbereitung einschließlich der Diskussion der Technologie bzw. der Programmierungsumgebung im Seminar sowie das verschriftlichte Konzept zum Einsatz dessen in der Schule.

Thematisiert und diskutiert werden im Seminar zudem der eigene Zugang zur Programmierung, der theoretische Hintergrund zum Programmieren mit Kindern (u.a. Konstruktivismus, Konstruktivismus) sowie Algorithmisierbarkeit und Problemlösekompetenz. Es werden Beispiele und Initiativen gezeigt, die sich mit Coding mit Kindern und Jugendlichen befassen und die Materialien und/oder Anregungen bieten. Außerschulische Lernorte mit Möglichkeiten zur Kooperation werden besprochen (z.B. Schullabore, Makerspaces, etc).

Der grobe Aufbau des Seminars kann deshalb wie folgt aussehen:

Sitzung	Thema
1	Einführung
2	Was ist Programmierung?
3	Warum sollen Kinder programmieren?
4 – 8	Kennenlernen und Diskussion der Technologien bzw. Programmierungsumgebungen
9	Zusammenfassung
10 – 12	Vorstellung und Diskussion der eigenen Konzepte
13	Reflexion und Abschluss

2. Ausführlicher Ablauf

1. Sitzung: Einführung

Was bedeutet programmieren für die Studierenden? Was verbinden sie mit dem Begriff? Welche Assoziationen rufen „Programmierung“ und „Informatik“ hervor? Die Studierenden programmieren eine selbst ausgewählte Einheit der „Hour of Code“ (<https://hourofcode.com/de/learn>).

2. Sitzung: Was ist Programmierung?

Es erfolgt eine theoretische Einordnung von Programmierung als Teil der Informatik und Teil eines Problemlöseprozesses. Im Anschluss werden die Programmiererfahrungen der „Hour of Code“ ausführlich diskutiert und reflektiert. Wie haben die Studierenden die Programmierung für sich erlebt? Wie deckt sich das mit ihrer ursprünglichen Wahrnehmung von Programmierung und Informatik? Was wurde als positiv oder negativ wahrgenommen? Diese Eindrücke werden geteilt, diskutiert und festgehalten und dienen u.a. als Grundlage für die Diskussion zum Einsatz verschiedener Programmierungsumgebungen in der Schule. Aspekte wie persönliche Bedeutsamkeit, Erfolgserlebnisse und

Selbstwirksamkeit, Diversität aber auch Spaß, Grundverständnis, Interesse wecken und Angst nehmen werden thematisiert und in die weiteren Sitzungen mitgenommen.

3. Sitzung: Warum sollen Kinder programmieren?

In dieser Sitzung wird in den Blick genommen, warum Kinder mit Programmierung in Berührung kommen sollten. Dabei geht es nicht um die Ausbildung künftiger Programmierer:innen, sondern vielmehr um Allgemeinbildung, Konstruktivismus, die Bildung einer individuellen Persönlichkeit, die Bildung in der digital vernetzten Welt und auch darum, Interesse für die technische Welt zu wecken. Gemeinsam finden die Studierenden im Anschluss Argumente für (und gegen) das Programmieren mit Kindern. Hier werden u.a. die selbst wahrgenommenen Aspekte aus der 2. Sitzung übertragen und in Bezug auf Programmieraktivitäten mit Kindern und Jugendlichen betrachtet.

Außerdem wird hier auf Problemlösekompetenz und Algorithmen eingegangen. Was ist ein Algorithmus? Wo findet man Algorithmen im Alltag? Wie entstehen sie und wie kommen sie in den Computer? Dies wird in Bezug auf die digital vernetzte Welt und die Inhalte des Seminars gesetzt und betrachtet und diskutiert.

4. – 8. Sitzung: Kennenlernen und Diskussion der Technologien bzw. Programmierumgebungen

In den folgenden Sitzungen werden verschiedene Technologien und Programmierumgebungen erprobt. Die Studierenden bereiten jeweils zu zweit oder zu dritt eine Technologie/ Programmierumgebung als Lerneinheit auf und führen diese mit den Studierenden in etwa 30 Minuten durch. Anschließend werden dann etwa 15 Minuten Vor- und Nachteile der Technologie/ Umgebung sowie deren möglicher Einsatz in der Schule diskutiert. Pro Sitzung können zwei Technologien bzw. Programmierumgebungen thematisiert werden.

Die Technologien und Programmierumgebungen sind vielfältiger Natur: es werden Mikrocontroller (Calliope mini, Arduino, ...), haptische Roboter (Ozobot, Botley, Cubetto, ...), digitale Produktion (BlockSCAD + 3D-Druck, TurtleStitch + Stickmaschine oder Laser-Cutter, ...) sowie rein virtuelle Programmierumgebungen (Scratch, MIT App Inventor, ...) vorgestellt. Auf diese Weise können Argumente wie persönliche Bedeutsamkeit, Selbstwirksamkeit, Interesse wecken etc. betrachtet und abgewogen werden. Die Studierenden erhalten einen Überblick über die Vielfalt an Möglichkeiten, sie können Potentiale erkennen und für ihr Fach geeignete Umgebungen identifizieren.

9. Sitzung: Zusammenfassung

Als Abschluss zum ersten Teil des Seminars werden die thematisierten Technologien und Programmierumgebungen zusammengefasst und nach Eigenschaften geclustert (Mikrocontroller, Roboter, etc). Es wird mit den Studierenden diskutiert, welche Aspekte für sie für die Nutzung einer Technologie oder Programmierumgebung in der Schule erfüllt sein müssen und welche sie davon abhalten würden. Anschließend werden zur Inspiration bereits umgesetzte Projekte, beispielsweise aus dem eigenen Schullabor, gezeigt.

10. – 12. Sitzung: Vorstellung und Diskussion der eigenen Konzepte

In den kommenden Terminen stellen die Studierenden ihre eigenen Ideen für den Einsatz einer der gezeigten Technologie/ Programmierumgebung im Unterricht vor. Da allen Studierenden alle Technologien bzw. Programmierumgebungen bekannt sind, gibt es kaum Verständnisschwierigkeiten und der Einsatz kann diskutiert und reflektiert werden. In insgesamt je 45 Minuten stellen die Gruppen ihre Idee unter aktiver Einbindung der Kommiliton:innen vor und diskutieren sie gemeinsam. Ideen, Kritik und mögliche auftretende Probleme können besprochen werden und in die schriftliche Ausarbeitung einfließen.

13. Sitzung: Reflexion und Abschluss

In einer abschließenden Sitzung wird das gesamte Seminar und dessen Inhalte zusammengefasst und reflektiert.

3. Voraussetzungen

Die Dozentin oder der Dozent sollte sich vorab mit verschiedenen aktuellen Technologien und Programmierumgebungen, die als ProgrammierEinstieg genutzt werden können, befassen und diese gut kennen. Die Person sollte selbst Programmierkenntnisse in verschiedenen Bereichen besitzen. Zudem sollten die Technologien in ausreichender Menge für die gemeinsame Arbeit im Seminar vorhanden sein.

Die Studierenden sollten neugierig und offen sein. Sie sollten sich bereits mit fachdidaktischen Inhalten und Fragestellungen befasst haben, um Programmierung methodisch für ihr Fach einordnen und anwenden zu können. Das Konzept sollen sie entwickeln, testen, reflektieren und ausformulieren. Daher sollten sich die Teilnehmenden bereits im Master-Studiengang befinden.

Das Seminar lebt von der aktiven Teilnahme in Form von Hands-on Aktivität, Reflexion und Diskussionen der Studierenden. Aus diesem Grund sollte es eine überschaubare Zahl Teilnehmender sein. Aufgrund pandemiebedingter Kontaktbeschränkungen wurde das Seminar in Präsenz bisher mit 10 Studierenden durchgeführt, was eine sehr angenehme Arbeitsatmosphäre zulässt und zudem für eine intensive Mitarbeit der Studierenden sorgt. In Abhängigkeit von der Anzahl Studierender können entweder die Gruppen aus zwei oder drei Personen bestehen oder die Länge der aktiven Auseinandersetzung mit den Technologien zwischen einer halben und einer ganzen Sitzung variieren. Es sollten etwa fünf bis acht Arbeitsgruppen gebildet werden können.

4. Prüfungsleistung

Die Prüfungsleistung im Modul *Medienbildung und Digitalisierung* wird als Portfolio bewertet. Hierbei handelt es sich um insgesamt 3-5 einzelne Leistungen, die im Seminar und der Vorlesung zusammen zu erbringen sind und insgesamt 6 KP entsprechen. In der zugehörigen Vorlesung werden zwei Kurzttests zum Thema „Medienbildung und Digitalisierung“ geschrieben, die sich auf die Inhalte der Vorlesung beziehen. Diese stellen die Hälfte der zu erbringenden Leistungen dar. Im Seminar werden die restlichen Portfolio-Leistungen erbracht. Es handelt sich hierbei um eine Aufbereitung einer Technologie (Roboter, Tools, rein digitale Programmierumgebung, ...) einschließlich Diskussion und eines verschriftlichten

Konzepts zum Einsatz der behandelten Umgebung in der Schule. Die Prüfungsleistung des Seminars wird als Gruppenarbeit bewertet und abgegeben.

5. Verwendetes Material

- Verwendete Programmierumgebungen:
 - Scratch
 - Turtlestitch
 - BlockSCAD
 - MIT App Inventor
- Verwendete Technologien:
 - Ozobot
 - Calliope mini + MakeCode oder NEPO
 - Arduino
 - Botley
 - Cubetto
- Grundlagenliteratur:
 - Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt - GI - Gesellschaft für Informatik e.V." <https://dagstuhl.gi.de/dagstuhl-erklaerung> (Zugriff am 26.07.2023).
 - Frankfurt-Dreieck zur Bildung in der digital vernetzten Welt <https://dagstuhl.gi.de/fileadmin/GI/Allgemein/PDF/Frankfurt-Dreieck-zur-Bildung-in-der-digitalen-Welt.pdf> (Zugriff am 26.07.2023).

6. Beispielergebnisse

Im ersten Durchlauf des Seminars im Wintersemester 2020/21 wurden neun Szenarien entwickelt, in denen Programmierumgebungen in Fächern außerhalb der Informatik bzw. interdisziplinär eingesetzt werden können. Entstanden ist u.a. ein Szenario in dem im Mathematikunterricht mit Hilfe von BlockSCAD ein Würfel programmiert und ausgedruckt wird. Die Programmierung findet am Ende einer Geometrieinheit zu „Raum und Form“ statt und dient als Überleitung zur Unterrichtseinheit „Daten und Zufall“ in der der Würfel zur Erhebung direkt eingesetzt werden kann.

Dieses Werk steht unter einer Creative Commons Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0) Lizenz. Um die Lizenz anzusehen, gehen Sie zu <https://creativecommons.org/licenses/by-sa/4.0/deed.de>.