

Modulcode (1.)	Modulbezeichnung (2.)	Zuordnung (3.)
BAI4530	Praxiswerkstatt „AI4Kids“ (AIK)	
	Studiengang (4.)	Bachelor Angewandte Informatik
	Fakultät (5.)	Gebäudetechnik und Informatik

Modulverantwortlich (6.)	Prof. Dr.-Ing. Oksana Arnold
Modulart (7.)	Wahl
Angebotshäufigkeit (8.)	SS
Regelbelegung / Empf. Semester (9.)	BA4 (BA6)
Credits (ECTS) (10.)	3 CP
Leistungsnachweis (11.)	SL (N)
Unterrichtssprache (12.)	Deutsch (Unterlagen teilweise Englisch)
Voraussetzungen für dieses Modul (13.)	BAI1040: Grundkonzepte der Programmierung BAI2040: Objektorientierte Programmierung BAI3020: Softwaretechnik 1 incl. PM-Grundlagen
Modul ist Voraussetzung für (14.)	-
Moduldauer (15.)	1 Semester
Notwendige Anmeldung (16.)	-
Verwendbarkeit des Moduls (17.)	Informatik & technische Studiengänge, in denen grundlegende Programmierkenntnisse benötigt, technische Systeme entwickelt und unterschiedliche Technologien genutzt werden, aber auch Studiengänge, die grundlegende Programmier- und Medienkenntnisse im Bildungskontext vermitteln

Lehrveranstaltung (18.)	Dozent/in (19.)	Art (20.)	Teilnehmer (maximal) (21.)	Anzahl Gruppen (22.)	SWS (23.)	Workload		
						Präsenz (24.)	Selbststudium (25.)	
1 AI4Kids	Arnold	S	12	4	2	30	45	
Summe						2	30	45
Workload für das Modul (26.)						75		

Qualifikationsziele

27.

Ziel der Praxiswerkstatt „AI4Kids&KinderKult“ ist es, dass Studierende ihr in den ersten 3 bzw. 5 Semestern erworbenes Wissen an Kinder im Alter zwischen 8 und 14 Jahren weitergeben. Auf diese Weise bestärken sie die Kinder in ihrem Interesse an Informatik und Technologie; lenken und unterstützen sie. Im Fokus steht dabei die Methodik zur Entwicklung von computerbasierten Anwendungen, in deren Kern Programmierfertigkeiten erforderlich sind. Daraus ergibt sich die didaktische Grundidee des Moduls: „Lernen durch Lehren“. Studierende festigen und erweitern ihr Wissen durch Reproduktion, Analyse, anwendungs- und sehr zielorientierte Vertiefung. Als Seiteneffekt wird auch die Fähigkeit entwickelt, fachspezifische Kenntnisse und Zusammenhänge in eine populärwissenschaftliche Darstellung zu transferieren und Zielgruppen ohne tiefgehendes Informatikwissen mit einfachen, alltäglichen Begriffen die Aufgaben, Probleme sowie Architekturlösungen und Wirkprinzipien zu erklären. Natürlich spielt auch die Zielgruppenanalyse eine Rolle, deren Kenntnisstand und präferierte Lernformen und –settings. Dazu sollen 6 Kurseinheiten unter Berücksichtigung des Lernzyklus von David Kolb und didaktischer Modelle konzipiert werden. Konkrete Projekte aus zwei spannenden Themenbereichen, ggf. auf unterschiedlichen Kompetenzniveaus, sind dazu auszuwählen, selbst zu erproben und für die Kinder geeignet als Schritt-für-Schritt-Abfolge aufzubereiten. Als Themenbereiche stehen zur Verfügung:

- Roboter und autonome Technik mit Lego® Mindstorms® für Anfänger (Kurs A)
- Roboter und autonome Technik mit Lego® Mindstorms® für Fortgeschrittene (Kurs B)
- Animieren und Programmieren mit Adobe® Flash® oder Unity3D® (Kurs C)
- Animieren und Programmieren mit Adobe® Flash® oder Unity3D® für Fortgeschrittene (Kurs D)

Jeder Kurs wird von 2 max. 3 Studierenden so bearbeitet, dass sie konkrete Aufgabenstellungen für die 6 Kurseinheiten in Abstimmung mit der verantwortlichen Dozentin entwickeln, Lösungen ausarbeiten und ausprobieren, die dazu notwendigen Wirkprinzipien und Wissens Elemente didaktisch mit Powerpoint-Präsentationen sowie mit Fragen und Aufgaben aufbereiten sowie die Kursstunden selbständig mit den Kindern durchführen. Dies ist damit eine Gelegenheit, angewandte Informatik bereits während des Studiums zu praktizieren und zielorientiert selbständig zu vertiefen. Konzeptionell setzt man sich ebenfalls mit der immer wiederkehrenden Frage auseinander, wie man visionäre Software-Wünsche, komplexe Anforderungen und Aufgaben sowie programmiertechnische Zusammenhänge durch die Zerlegung in kleine umsetzbare Aktivitäten erfüllen bzw. berücksichtigen kann. Anhand der gesammelten Erfahrungen kann das eigene Lernverhalten hinterfragt, die eigene präferierte Lernform identifiziert und daraus der Lernzyklus von David Kolb für die Optimierung der eigenen Lernergebnisse bzw. der Persönlichkeitsentwicklung individuell genutzt werden.

- **Fach-/Systemkompetenz:** Erwerb und Festigung von

	<p>Kenntnissen zu Begriffen und Inhalten der Softwaretechnik, der Programmierung, der technischen & theoretischen Informatik sowie zu algorithmischen Lösungen.</p> <ul style="list-style-type: none"> ○ technologische Kompetenz: Erwerb und Festigung von Kenntnissen zu konkreten Programmierumgebungen und zur konkreten Umsetzung von Programmierparadigmen; Zuordnung von Aufgabeklassen zu Konzepten & Konstrukten im jeweiligen Programmierparadigma ○ Analyse-, Design-, Realisierungs- und Projektmanagementkompetenz: Analyse konkreter Aufgabenstellungen und Applikationen; selbständige Umsetzung kleiner Aufgaben im gewählten Themenbereich; Aufbereitung der Problemlöse- und Softwareentwicklungsschritte; selbständige Organisation des Kurses mit termingerechter Vorbereitung der Unterlagen, der Rechner und der Funktionsfähigkeit von Musterlösungen <ul style="list-style-type: none"> ● Methodenkompetenz: Fähigkeit zur bewussten Gestaltung von Anwendungsentwicklung, von Lernprozessen, von Veranstaltungseinheiten, von Präsentationen; Erwerb von Transferkompetenz zur zielorientierten Auswahl von Aufgaben, Lösungen für eine gelungene Kompetenzentwicklung und Wissensvermittlung; Transferkompetenz von Fachbegriffen der Informatik in eine allgemeinverständliche Terminologie; Reflexion zur Optimierung eigener Handlungsgestaltung; didaktische Modelle; Lernen durch Lehren ● Sozialkompetenz: Kennenlernen teamdynamischer Prozesse, Erprobung und Adaption von Teamführung und Kommunikation, Aufbau von Beziehungen, Interpretation von menschlichem Verhalten, Zielgruppenanalyse allgemein und speziell bzgl. technologischer Kenntnisse und Kompetenzen, Umgang mit Feedback, Umgang mit Kindern, Repräsentation einer Institution ● Selbstkompetenz: Analyse und Optimierung des eigenen Lernverhaltens; Stärkung der Fähigkeit, selbständig neue Wissensgebiete zu erschließen, eigene Programmierfähigkeiten zu erweitern sowie die eigenen Arbeitsaufgaben zu organisieren; Erweiterung der eigenen Präsentations- und Darstellungsfähigkeiten, Selbstbewusstsein und Interaktionsfähigkeit, Zielorientierung und Projektmanagement.
<p>Inhalte</p>	<p>Die Praxiswerkstatt „AI4Kids“ bietet eine Möglichkeit, das eigene Wissen zur Informatik in konkreten Anwendungsbereichen über das didaktische Konzept „Lernen durch Lehren“ zu vertiefen. Dabei wird auch der Frage nachgegangen, wie lässt sich der Lernprozess optimal gestalten, auf dessen Basis die Studierenden selbständig eine Konzeption des gewählten Kurses</p>

	<p>vornehmen, mit konkreten Inhalten füllen und mit Kindern erproben. Ebenfalls wird semesterbegleitend Feedback gegeben und über die Veranstaltungen reflektiert. Konkrete Lehrinhalte sind:</p> <ul style="list-style-type: none"> • Lernzyklus nach David Kolb • Didaktische Modelle nach Jank & Meyer • Zielgruppenanalyse: Kenntnisstand, Teilnahmemotivation und Lernpräferenzen • Entwicklung von Problemlösestrategien – Rolle von Exploration • Notwendigkeit von Anleitung und/oder kreativem Freiraum • Rolle: Trainer versus Coach (GROW-Modell) • Definition von Lernzielen für einen Kurs und daraus abgeleitet für Kurseinheiten • Strukturierung und Planung einer Kurseinheit (90 min) • Entwicklung von Kursunterlagen: Powerpoint-Präsentation, Handouts, Beispielprogrammen • Kernbegriffe der Informatik und informatikrelevante Kernbegriffe des gewählten Themenbereichs: Bedeutung, Definition, Darstellung, Nutzung • Rolle von Erfolg, Misserfolg, Feedback und Reflexion
<p>Vorleistungen und Modulprüfung</p>	<p>(29.) Voraussetzungen –planmäßige Abgabe der Kursunterlagen Modulleistung – 100% Vorbereitung und Durchführung der eigenen AI4Kids-Kursveranstaltungen</p>
<p>Literatur</p>	<p>(30.)</p> <ul style="list-style-type: none"> • Eigene Unterlagen und Literatur des BA-Studiums (1.-3.FS) <p>didaktische Literatur</p> <ul style="list-style-type: none"> • Werner JANK, Hilbert MEYER: Didaktische Modelle. 5. stark überarbeitete Auflage, Cornelsen Verlag, Berlin 1991. • David A. KOLB: Learning Style Inventory. McBer and Company, Boston 1985. • http://wiki.zum.de/Unterrichtsmethoden <p>kursspezifische Literatur zu LEGO® Mindstorms® Modellen</p> <ul style="list-style-type: none"> • Fay RHODES: NXT one-kit creatures : five designs for the LEGO Mindstorms NXT 1.0 or NXT 2.0. Create Space, North Charleston, SC, USA 2010. • Daniele BENEDETTELLI: Lego Mindstorms NXT thinking robots: build a rubik's cube solver and a tic-tac-toe playing robot. No Starch Press, San Francisco, CA 2010. • Fay RHODES: Robots alive! endangered species : LEGO Mindstorms NXT. Create Space, North Charleston, SC, USA 2009. <p>kursspezifische Literatur zu Adobe® Flash®</p> <ul style="list-style-type: none"> • William SANDERS, Chandima CUMARANATUNGE: ActionScript 3.0 Design Patterns: Object Oriented Programming Techniques (Adobe Developer Library). O'Reilly Media, Inc., Sebastopol, CA, USA 2007 • Colin MOOCK: Essential ActionScript 3.0. O'Reilly Media, Inc., Sebastopol, CA, USA 2007 • Joey LOTT, Darron SCHALL, Keith PETERS: ActionScript 3.0 Cookbook: Solutions for Flash Platform and Flex Application Developers. O'Reilly Media, Inc., Sebastopol, CA, USA 2006 • Rob BATEMAN, Richard OLSSON: The Essential Guide to 3D in Flash. Apress, Springer-Verlag New York Inc., New York 2009.

- | | |
|--|--|
| | <ul style="list-style-type: none">• Jeff FULTON, Steve FULTON: The Essential Guide to Flash Games: Building Interactive Entertainment with ActionScript 3.0. Apress, Springer-Verlag New York Inc., New York 2010.• Colin MOOCK: ActionScript für Flash MX - Die Referenz. O'Reilly Media, Inc., Sebastopol, CA, USA 2003• Colin MOOCK: ActionScript für Flash MX - Das Handbuch. O'Reilly Media, Inc., Sebastopol, CA, USA. |
|--|--|