

Modulcode (1.)	Modulbezeichnung (2.)	Zuordnung (3.)
BAAI-1230	Objektorientierte Programmierung (OOP)	
	Studiengang (4.)	Bachelor Angewandte Informatik
	Fakultät (5.)	Gebäudetechnik und Informatik

Modulverantwortlich (6.)	Prof. Dr.-Ing. Jörg Sahm
Modulart (7.)	Pflicht
Angebotshäufigkeit (8.)	SS
Regelbelegung / Empf. Semester (9.)	BA2
Credits (ECTS) (10.)	5 CP
Leistungsnachweis (11.)	PL (N)
Unterrichtssprache (12.)	Deutsch
Voraussetzungen für dieses Modul (13.)	BAAI-1140: Grundkonzepte der Programmierung
Modul ist Voraussetzung für (14.)	BAAI-1310: Programmierung Java 1 BAAI-1410: Programmierung Java 2 BAAI-8610: Einführung in die KI
Moduldauer (15.)	1 Semester
Notwendige Anmeldung (16.)	-
Verwendbarkeit des Moduls (17.)	-

Lehrveranstaltung (18.)	Dozent/in (19.)	Art (20.)	Teilnehmer (maximal) (21.)	Anzahl Gruppen (22.)	SWS (23.)	Workload	
						Präsenz (24.)	Selbststudium (25.)
1 Objektorientierte Programmierung	Sahm	V	100	1	2	30	15
2 Objektorientierte Programmierung	Sahm	Ü	25	4	2	30	50
Summe					4	60	65
Workload für das Modul (26.)						125	

Qualifikationsziele	<p>Die Studierenden können...</p> <ul style="list-style-type: none"> • die grundlegenden Prinzipien der OOP benennen und mit eigenen Worten am Beispiel beschreiben; • Konzepte wie Referenzen, abstrakte Klassen, virtuelle Methoden, Klassenkonstanten, -attribute und -methoden richtig einordnen und Anwendungsmöglichkeiten aufzeigen; • Aggregations- und Kompositionsbeziehungen zwischen Klassen in geeigneter Weise implementieren; • den Lebenszyklus von Objekten konsistent gestalten (Konstruktoren, Destruktor, Kopie, Wertzuweisung, dynamische Inhalte); • aus einer verbalen Aufgabenstellung ein sinnvolles System von Objektklassen ableiten, passende Schnittstellen entwerfen und begründen; • ein Projekt sourcecode-verwaltet im Team implementieren; • analysieren, wo Container-Templates der C++-Standardbibliotheken effizient eingesetzt werden können, und dies in korrekter Weise umsetzen.
Inhalte	<ul style="list-style-type: none"> • Ziele und Grundprinzipien der objektorientierten Programmierung; • Klassenentwurf in C++ mit und ohne Klassenassistent in VisualStudio als Grundlage von OOP-Projekten; • Objekt-Lebenszyklus; • Vererbung und Polymorphismus; • Operatoren und Funktionsobjekte; • Iteratoren; • Streams und Strings in C++; • Sachgerechter Einsatz von Container-Templates C++; • Defensive Programmierung mit Exceptions; • Ausgewählte Design-Patterns.
Vorleistungen und Modulprüfung	<p>Vorleistungen:</p> <ul style="list-style-type: none"> • keine <p>Modulprüfung:</p> <ul style="list-style-type: none"> • 100 % Klausur über 90 min im Prüfungszeitraum

Literatur

30.

- Robert SEDGEWICK: Algorithmen in C++, Teil 1-4. – 3. Aufl. – München: Pearson Studium, 2002;
- Bjarne STROUSTRUP: The C++ Programming Language. – Special 4th Edition. – Addison Wesley, 2013;
- Ulrich KAISER, Christoph KECHER: C/C++ - Von den Grundlagen zur professionellen Programmierung. – Bonn: Galileo Press, 2008;
- Jürgen WOLF: C++ von A bis Z. Das umfassende Handbuch, aktuell zum Standard C++11. – Galileo Computing, 2014;
- Christoph KECHER, Alexander SALVANOS: UML 2.5. Das umfassende Handbuch. – 5. Aufl. – Bonn: Rheinwerk Computing, 2015;
- Bernd OESTEREICH, Axel Scheithauer Die UML-Kurzreferenz 2.3 für die Praxis. Kurz, bündig, ballastfrei. – 6. Aufl. – München: Oldenbourg, 2014;
- Erich GAMMA, Ralph JOHNSON, John VLISSIDES, Richard HELM, Martin FOWLER: Design Patterns. Elements of Reusable Object-Oriented Software. – 2nd rev. ed. – Pearson Education, 2015
- Matthias GEIRHOS: Entwurfsmuster. Das umfassende Handbuch. – Bonn: Rheinwerk, 2015