

Modul: Programmanalyse (6 Credits)	
Name im Diploma Supplement	Programmanalyse
Verantwortlich	Dr.-Ing. Dominik Helm
Voraussetzungen	Siehe Prüfungsordnung.
Workload	180 Stunden studentischer Workload gesamt, davon: <ul style="list-style-type: none"> • Präsenzzeit: 60 Stunden
Dauer	Das Modul erstreckt sich über 1 Semester.
Qualifikationsziele	Die Studierenden <ul style="list-style-type: none"> • haben ein Verständnis der zugrundeliegenden Konzepte von statischen Programmanalysen • kennen wichtige Probleme und Algorithmen der statischen Programmanalyse • sind in der Lage, grundlegende Eigenschaften von statischen Programmanalysen wie Präzision und Korrektheit zu beurteilen
Prüfungsmodalitäten	Zum Modul erfolgt eine modulbezogene Prüfung in Form einer Klausur.
Verwendung in Studiengängen	<ul style="list-style-type: none"> • SE Bachelor 2023 > Wahlpflichtbereich > Wahlpflichtbereich Informatik > Wahlpflichtmodule aus dem Bereich Informatik > 5.-6. FS, Wahlpflicht
Bestandteile	<ul style="list-style-type: none"> • Vorlesung: Programmanalyse (3 Credits) • Übung: Programmanalyse (3 Credits)
ZED62021	

Vorlesung: Programmanalyse (3 Credits)			
Name im Diploma Supplement	Programmanalyse		
Anbieter	Abteilung für Software Engineering		
Lehrperson	Dr.-Ing. Dominik Helm		
SWS	2	Sprache	englisch
Turnus	Sommersemester	maximale Hörschaft	unbeschränkt
<p>empfohlenes Vorwissen</p> <p>Wir erwarten, dass Studierende die Konzepte der Programmiersprachen, wie Zuweisungen, Schleifen, Ausnahmebehandlung, Objekte, und anonyme Funktionen gut verstanden haben.</p> <p>Des Weiteren sollten die Kursteilnehmer*innen mit grundlegender Universitätsmathematik und Logik vertraut sein.</p> <p>Abstract</p> <p>Statische Analysen sind Werkzeuge, die Informationen über Programmen extrahieren, ohne diese auszuführen. Statische Analysen haben vielseitige Anwendungen in der Erkennung von Bugs, Sicherheitslücke und Schadsoftware, in Typechecking und Optimierung während des Kompilierens sowie als Grundlage für die formale Verifikation von Programmen. Daher sind sie ein wichtiges Werkzeug, um die Korrektheit, Sicherheit und Performanz von Software sicherzustellen.</p> <p>Lehrinhalte</p> <p>Dieser Kurs gibt einen Überblick über die zugrundeliegenden Konzepte von statischen Analysen. Der Kurs wird wichtige Programmeigenschaften vorstellen, die von statischen Analysen hergeleitet werden, darunter Kontrollfluss, Datenflüsse, Callgraphen, Points-To, Purity und Immutability. Sie lernen auch Algorithmen und Frameworks kennen, die verwendet werden, um diese Eigenschaften herzuleiten, darunter Typsysteme, abstrakte Interpreter und IFDS/IDE. Schließlich werden wir auch Abwägungen zwischen Präzision, Korrektheit und Performanz statischer Analysen diskutieren.</p> <p>In der begleitenden Übung wenden Sie die neuen Analysekonzepte praktisch an und erweitern oder entwickeln existierende Analysen.</p> <p>Literaturangaben</p> <ul style="list-style-type: none"> • Nielson/Nielson/Hankin: Principles of Program Analysis, Springer, Corrected Edition, 1999 • Vorlesungsskript 			
ZED62021			

Übung: Programmanalyse (3 Credits)

Name im Diploma Supplement	Programmanalyse		
Anbieter	Abteilung für Software Engineering		
Lehrperson	Dr.-Ing. Dominik Helm		
SWS	2	Sprache	englisch
Turnus	Sommersemester	maximale Hörschaft	unbeschränkt
empfohlenes Vorwissen	keines		
Lehrinhalte	Siehe Vorlesung		
Literaturangaben	siehe Vorlesung		
ZED62021			