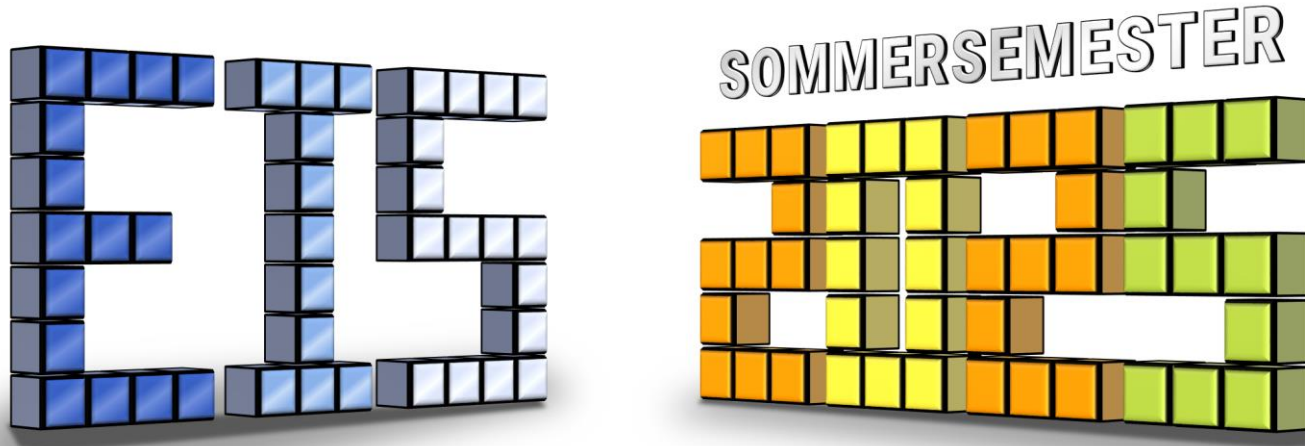


EINFÜHRUNG IN DIE PROGRAMMIERUNG

Sommersemester 2025



Organisation & Inhalte Abschlussklausur

Michael Wand
Institut für Informatik
Michael.Wand@uni-mainz.de



Klausur EiP

Klausurtermin (→Jogustine)

- **Datum:** Donnerstag, 07. August 2025
- **Zeit:** 09:00-12:00 (s.t.)
- **Raum:** RW 1 (großer Hörsaal)

Gebäude Recht und Wirtschaft 1
Jakob-Welder-Weg 9
55128 Mainz

Eintrag in Jogustine beachten!

Klausurzulassung

Kriterium

- Mindestens 50% der Punkte in Übungsblättern
- Eintrag in Jogustine:
 - Ab Montag nächste Woche wird eingetragen
 - Mitte nächster Woche Ergebnis verfügbar
 - Nachricht via LMS, wenn Eintrag verfügbar
- Bei Unstimmigkeiten bitte so schnell wie möglich beim Oberassistenten melden!
 - Christian ALI MEHMETI-GÖPEL, chalimeh@uni-mainz.de

Ablauf & Hilfsmittel

Ablauf der Klausur

Dauer der Klausur

- Drei volle Zeitstunden
- Wir fangen pünktlich an!

Nachteilsausgleich

- Bei medizinisch nachgewiesenen Nachteilen
 - Antrag nötig, unbedingt mit Studienbüro sprechen
 - Rechtzeitig vor der Klausur stellen (ggf. zeitnah Handlungsbedarf)

Bei Schwierigkeiten

- Gerne Gespräch suchen (Studienberater, Dozenten, Psychologische Beratungsstelle der Universität, etc.)

Zulässige Hilfsmittel

Sie müssen mitbringen

- Schreibmaterial (Stifte)
- Studierendenausweis & Lichtbildausweis

Sie dürfen zusätzlich mitbringen

- Bis zu **zwei Din-A4 Seiten**, *handbeschrieben*
 - Eigenhändig angefertigt
 - Beidseitiges Beschreiben erlaubt
- Ein Wörterbuch
 - Übersetzung Deutsch ↔ Fremdsprache^{*)}
 - Klausur: Beantwortung auf Deutsch oder Englisch möglich!

^{*)} für menschliche Sprachen, nicht für Programmiersprachen

Zulässige Hilfsmittel

Nicht erlaubt & nicht nötig

- Keine Taschenrechner
- Mobiltelefone ausschalten!
- Weitergehende Hilfsmaterialien / Unterlagen
 - Auch keine Chat-Bots :-)

Ablauf

Deckblatt

- Mit Namen, Studiengang, Matrikelnummer

Klausur & zusätzliches Papier

- Nur Matrikelnummer
- Anonyme Korrektur

Abgabe

- Nur eine, eindeutige Lösung pro Aufgabe
 - „Falsche“ Lösungen bitte durchstreichen
- Schmierzettel sollen nicht abgegeben werden

Klausurinhalte

Ziel der Klausur

EIS-Sommer-2025

- Praktisch orientiert
- Projektartige Übungen
- Aktive Teilnahme an den Übungen zentral

Ziel der Klausur

Klausurinhalte

- **Praxis** (ca. 70-80%)

Programmieraufgaben, die konzeptionell den Übungen nah sind

- Leicht zu lösen, wenn man die Übungen gelöst hat
- Kein Auswendiglernen (keine APIs)

- **Theorie** (ca. 20-30%)

Einige konzeptionelle Fragen zur Vorlesung

- Verständnisfragen zu Konzepten
- Keine „Schlagwörter“ auswendiglernen, Bedeutung kennen!
- Geringeres Gewicht

Konkrete Inhalte: Praxis

Mögliche Themen (praktisch)

- Programmieren von GUIs
 - Ereignisorientiertes Programmieren
 - Programmiertechniken aus dem Umfeld des Vektorgraphik-Beispielprojektes
- Objektorientierter und funktionaler Entwurf

Programmiersprachen praktisch

- Python oder Scala, freie Auswahl

Kein Auswendiglernen

Was man für die praktischen Aufgaben können muss

- Programmieren mit Python oder Scala
- Freie Wahl der Sprache

Was man nicht können muss

- Keine APIs auswendiglernen (z.B. Sockets, Qt/PySide)
- Statt dessen dummy-APIs
- Konzeptionelles Verständnis auch hier zentral

Altklausur SoSem 2024

PYTHON	SCALA
<pre>class EISPainter: # Ein Pixel setzen (col = Farbe; nur True / False) def set_pixel(x: int, y: int, col: bool) -> None</pre>	<pre>class EISPainter { // Ein Pixel setzen (col = Farbe; nur True / False) def set_pixel(x: Int, y: Int, col: Boolean): Unit }</pre>
PYTHON	SCALA
<pre>class EISWidget(ABC): # (1) Maustaste gedrückt def mouse_down(x: int, y: int) -> None # (2) Maustaste losgelassen def mouse_up(x: int, y: int) -> None # (3) Maus bewegt (Zustand Maustaste egal) def mouse_move(x: int, y: int) -> None # (4) Aufruf, wenn Widget sich zeichnen soll def paint(p: EISPainter) -> None</pre>	<pre>abstract class EISWidget { // (1) Maustaste gedrückt def mouse_down(x: Int, y: Int): Unit; // (2) Maustaste losgelassen def mouse_up(x: Int, y: Int): Unit; // (3) Maus bewegt (Zustand Maustaste egal) def mouse_move(x: Int, y: Int): Unit; // (4) Aufruf, wenn Widget sich zeichnen soll def paint(p: EISPainter): Unit; }</pre>

Vermutlich wieder ähnliches „API“ in Klausur

„Scala für Wiederholer/innen“

Eine Wahlaufgabe

- Schwierigere Programmiertechnik aus unseren Übungen
- Alternativ: Schwierigere Aufgabe zum Programmieren in Scala

Freie Auswahl für alle Teilnehmenden!

- Nur eine Aufgabe lösen!
- Klar festhalten, welche Aufgabe ausgewählt wurde!

Konkrete Inhalte: Theorie

Beispiele für konzeptionelle Fragen

- Konzepte in anderen Programmiersprachen
 - Python, Scala, C++ – kein Coding, nur Ideen/Konzepte
- Funktionale Entwurfsmuster
- Objektorientierte Prinzipien
- Effiziente / low-level Programmierung
- Konstrukte in Programmiersprachen (z.B. Typisierung)

Geringeres Gewicht (ca. 20-30%, ohne Gewähr)

- typ. Freitextfragen, Verständnis-orientiert

Fragen?