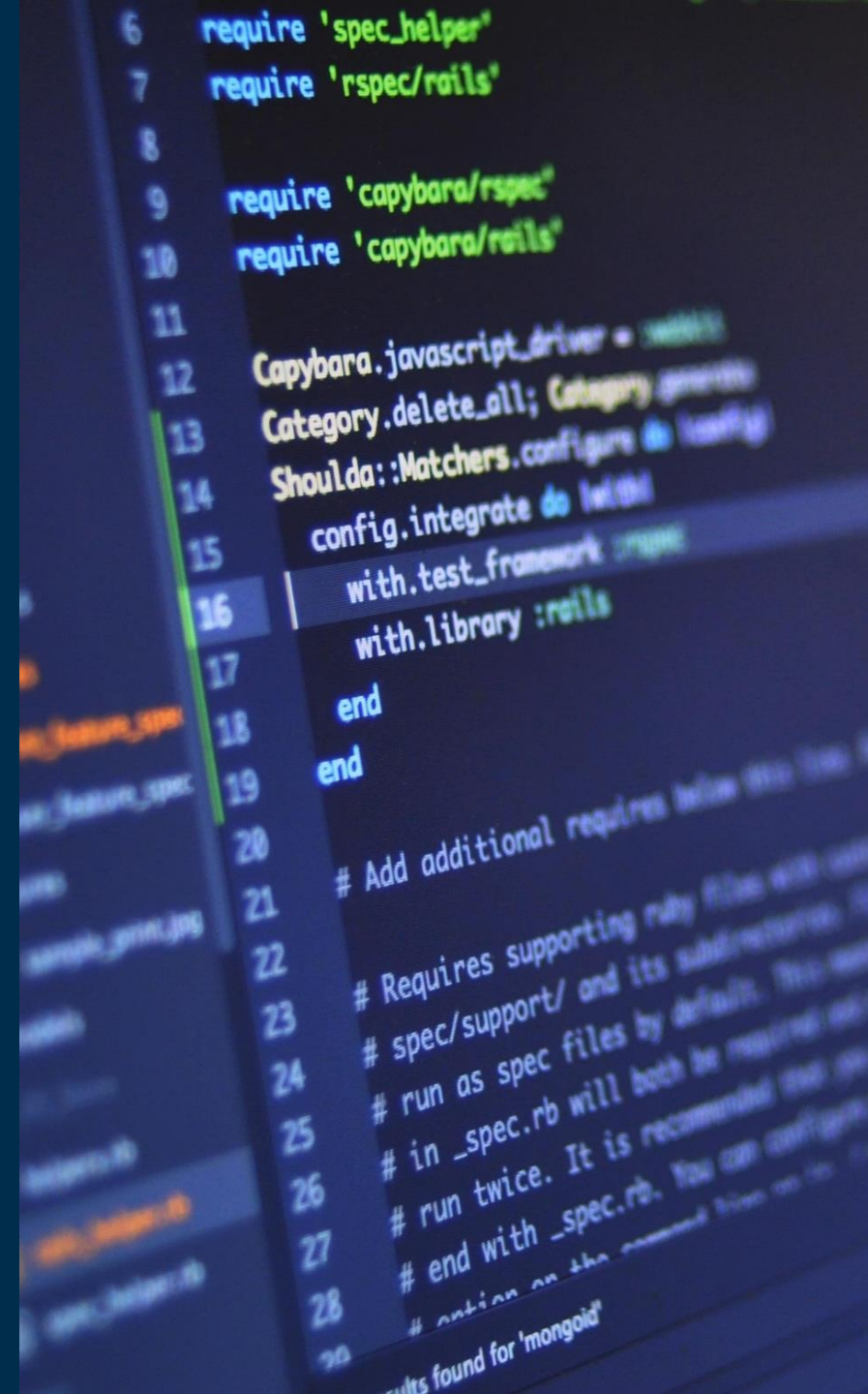


Info² : “Programmieren Best Practices ” + Lernpartnerschafts börse”

KIT-Fakultät für Informatik
Informatik Studiengangservice

Mittwoch, 20. Mai 2026 um 17:45 Uhr



Achtung



- Diese Veranstaltung wird aufgezeichnet!
- Fragen sind immer willkommen, auf Englisch und Deutsch
- Wir werden während der Q&A-Session am Ende fragen antworten
 - Der Q&A-Teil am Ende wird nicht aufgezeichnet



Auf unserer [Info²](#) -Website finden Sie ein Veranstaltungsarchiv mit:

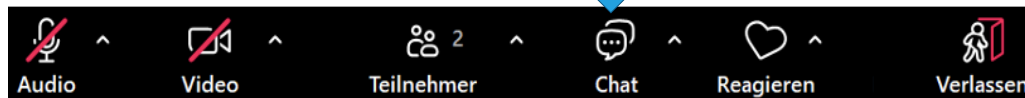
- die Folien der Vorlesung als PDF
- Link zu Video auf [YouTube Playlist UimS](#)



Unterstützung im
Studium
Youtube-Kanal

Für Online-Teilnehmer

- Aktiviert euren Chat!



- Stellt dort eure Fragen, wir werden sie sammeln und während der Q&A-Session am Ende fragen
- Der Chat wird nicht aufgenommen

1. Überblick
2. Projekt BESSEr@KIT
3. Best Practices für „Programmieren“
4. Q&A



BESSEr@KIT





Partizipation – was ist das?

BESSEr@KIT



Was ist BESSEr@KIT?



BESSEr@KIT ist ein einmaliges Projekt am KIT, welches sich um (fast) alle Themen der studentischen Beteiligung kümmert:

Warum ist studentische Beteiligung so wichtig?

Warum ist eure Beteiligung für das KIT und auch für euch als Studierende so wichtig?

Was hat Beteiligung für einen Mehrwert?

Wie kann man sich beteiligen?

Wie sind Studierende am KIT eigentlich organisiert?

Wahrscheinlich am bekanntesten:

AStA^{KIT}

&

Fachschaften

Teile der

Verfassten
Studierendenschaft
(VS)

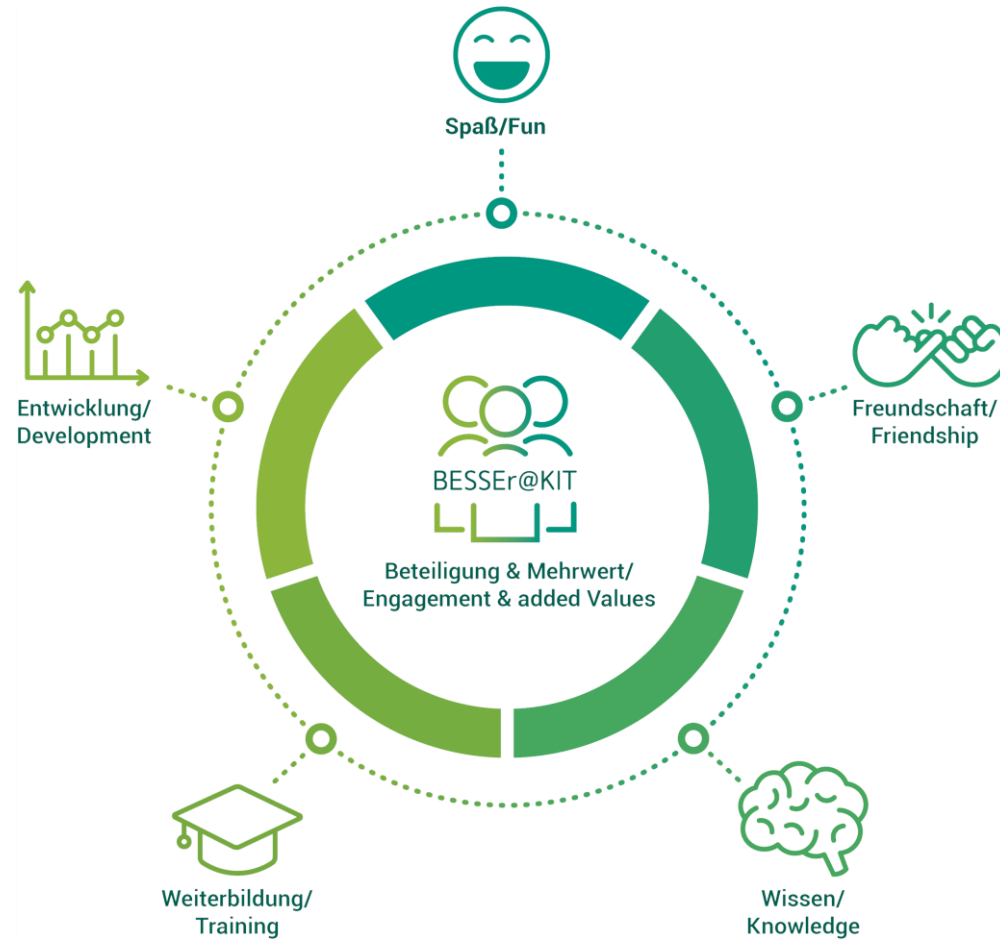
Wie sind Studierende am KIT eigentlich organisiert?



§65 (1) Die immatrikulierten Studierenden (Studierende) einer Hochschule bilden die Verfasste Studierendenschaft (Studierendenschaft).

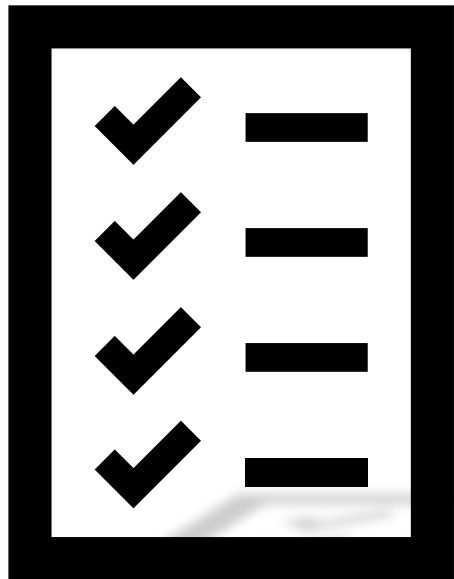
Sie ist eine rechtsfähige Körperschaft des öffentlichen Rechts und als solche eine Gliedkörperschaft der Hochschule.

Warum Partizipation?



Einfachste Formen der Beteiligung

Lehrevaluationen & Befragungen zu Studium und Lehre



KIT
Karlsruher Institut für Technologie

Immatrikulationsbefragung am KIT
Wintersemester 2023/2024
Nadja Legrum-Khaled, Anne Pfeifer

KIT
Karlsruher Institut für Technologie

Befragung zum Studium unter Pandemiebedingungen
Sommersemester 2021
Anne Pfeifer und Nadja Legrum-Khaled, Qualitätsmanagement KIT

KIT
Karlsruher Institut für Technologie

Absolventenbefragung am KIT
Abschlussjahrgang 2022 – Berichterstellung im Wintersemester 2023/24
Anne Pfeifer, Nadja Legrum-Khaled

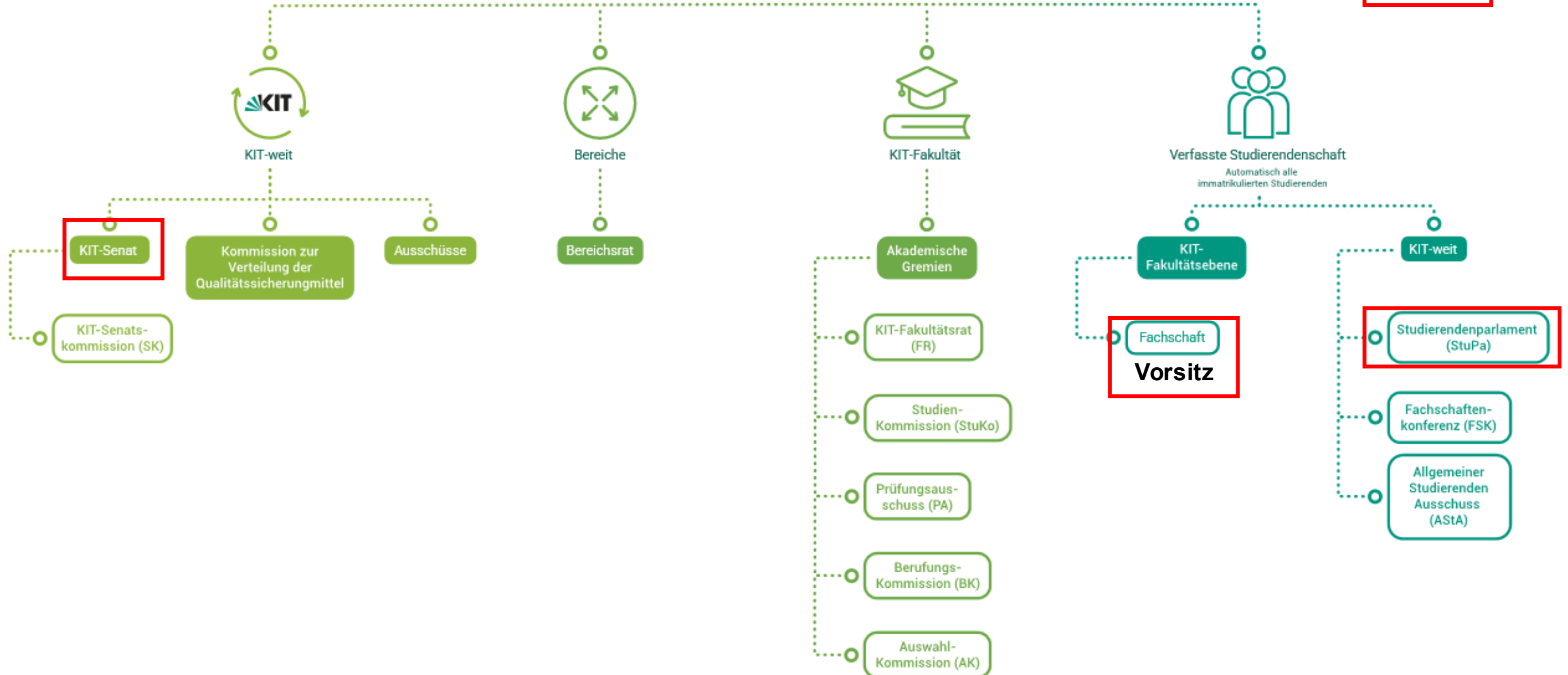
www.kit.edu



Wie? Wo? Was?

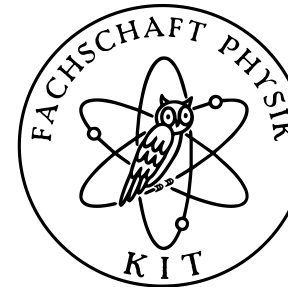


Ihr wählt...



Wer? Wo? Was?

KIT-intern (Auswahl)



AStA^{KIT}

FSK^{KIT}



Studierendenparlament
Deine Interessensvertretung an der Uni!

uvm....

Wer? Wo? Was?

Hochschulgruppen am KIT (Auswahl)



uvm. ...

Wer? Wo? Was?

Möglichkeiten des Engagements außerhalb des KIT



freier Zusammenschluss
von student*innenschaften

Demokratie
Menschenrechte

Bündnis Karlsruhe

AKK







bundesverband
ausländischer
studierender

uvm. ...

Informationen

Glossar - Glossary

Glossare

  A Glossary: Higher Education in Karlsruhe, Baden-Württemberg, Germany, and Europe Abbreviations and Terms - Higher education can be a maze. This glossary offers some clarity.	  Ein Glossar: Begriffe und Strukturen der Hochschulbildung - von Karlsruhe bis Europa Abkürzungen und Bezeichnungen - die Hochschullandschaft kann verwirrend sein. Hier gibt es Hilfe und Erklärungen.
---	--



<https://s.kit.edu/glossar>

Veranstaltungen & Angebote

Bei ILIAS, digital
und auf Wunsch
auch in Präsenz



Mythbusting
Hochschulpolitik

BESSER@KIT
Engagement ohne (viel) Arbeit?

Lehrevaluationen & Befragungen in Studium
und Lehre

BESSER@KIT
Gemeinsam für ein BESSERes KIT

Engagement, das
verbindet!

Das Politische Mandat
Streitfrage in der
Hochschulpolitik

Jan Fritz, STAB-QM, BESSER@KIT

Politisches Mandat

ASStA KIT Baden-Württemberg
Z90 BUNDESWEITER STREIK
LUCKY STREIK
AKK #USTA
Unter den Talaren
Muff von 1000 Jahren
gegen HRG, LHG und
politische Entrechtung
bologna process VERURTEILTE DEUTSCHE STUDIENSCHEITEN

Mehr Informationen



Programmieren Best Practices

Best Practices für „Programmieren“

Info² | Info-Bites | Nils Niehues



Das Programmieren-Team



Nils Niehues



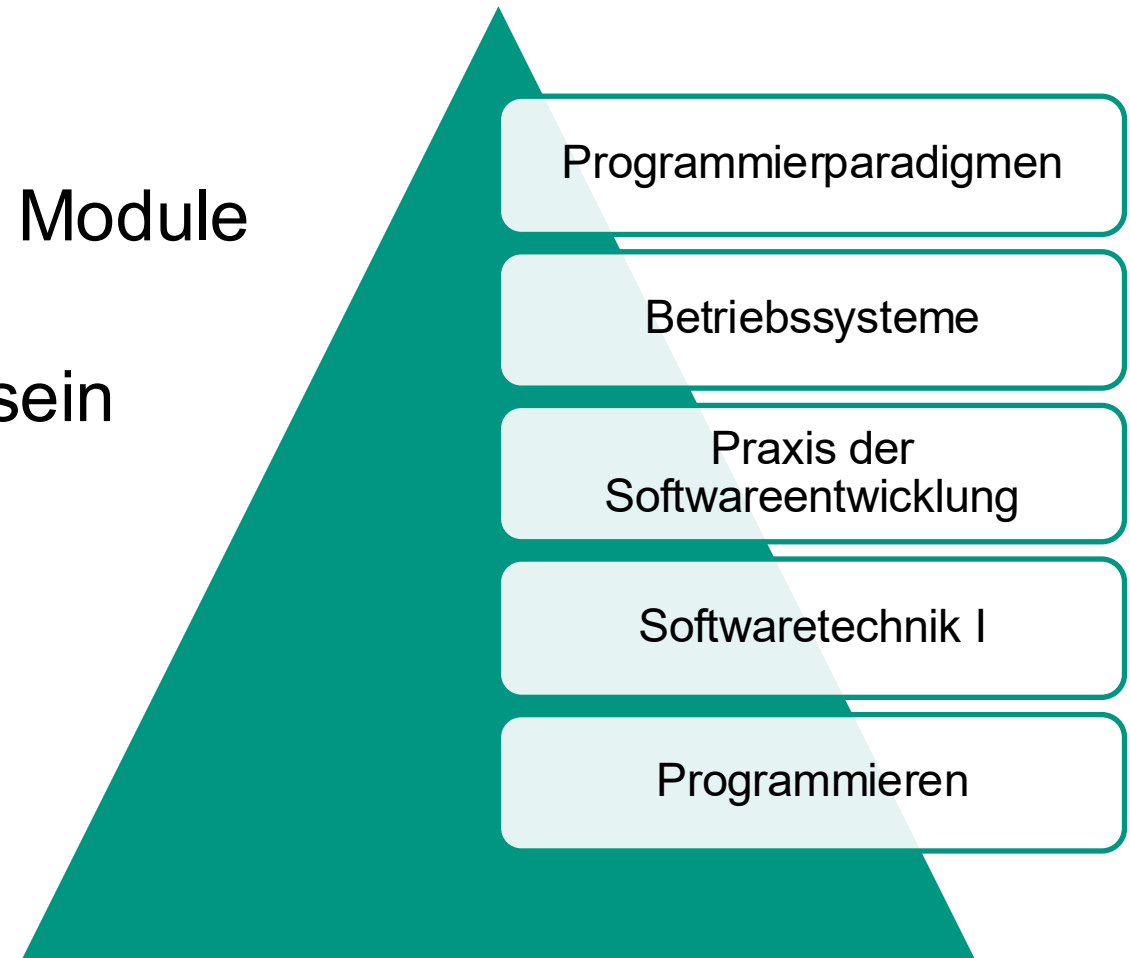
Robin Maisch

Best Practices für „Programmieren“

- Heutige Ziele:
 - Woran kann es scheitern?
 - Wie Frust sparen?
 - Wie beste Chancen haben
- Nicht alle Kriterien werden bei den ersten Blättern anwendbar sein!
 - Es muss nicht alles *sofort* verstanden werden
 - Aber spätestens bei den Abschlussaufgaben

Grundstudium: Praktische Informatik

- Programmieren...
 - ...bildet die Grundlage für weitere Module
 - ...ist Orientierungsprüfung
 - ...sollte vor SWT abgeschlossen sein
- Was kann man mit guten Programmierkenntnissen tun?
 - Studentische Hilfskraft
 - Werkstudent
 - Open-Source-Projekte



Was Sie aus diesem Modul mitnehmen sollten

- Die Grundlagen der objektorientierten Programmierung in Java
- **Abbildung:** Problemmodellierung
 - Wie man alltägliche Probleme in Programmiersprache modelliert
- **Automatisierung:** Lösungsformulierung
 - Entwicklung von Verfahren (Algorithmen) zur Lösung einfacher Probleme
- **Abstraktion:** Problemformulierung
 - Wie man nicht nur eine Probleminstance löst, sondern eine allgemeine Lösungsmethode findet, die für viele Probleminstance funktioniert
- **Sauber programmieren!**
 - Wichtig für Wartbarkeit, Wiederverwendbarkeit und Zusammenarbeit



Allgemeine Tipps zur Studienorganisation

- Termine und Fristen verfolgen
 - Bereits zu Beginn des Semesters veröffentlicht: s.kit.edu/programmieren
 - In den eigenen Kalender mit Erinnerung übernehmen
 - Keine nachträglichen Anmeldungen für Prüfungsleistungen
- Studien- und Prüfungsordnung beachten
 - Arbeit muss selbstständig angefertigt werden
 - Der Übungsschein kann mehrfach wiederholt werden
 - Abschlussaufgaben können nach dem Übungsschein erworben werden
 - Die Abschlussaufgaben können nur einmal wiederholt werden
 - Muss bis zum Ende des dritten Semesters bestanden werden
 - Keine mündliche Nachprüfung

Vorlesung und Übung nachbereiten

- Folien der Vorlesung während des Semesters aktiv nacharbeiten
 - Im Semester nacharbeiten und nicht kurz vor den Abschlussaufgaben
 - Aufzeichnungen der Vorlesung sind auf YouTube verfügbar
 - Vorlesungsfolien sind im ILIAS vorhanden: s.kit.edu/ilias
 - Übungsblätter orientieren sich an der Struktur der Vorlesung
- Beispiellösungen der Übungsblätter nachvollziehen
 - Geben Hinweise zur Bearbeitung der Abschlussaufgaben
 - Nur Beispiele dafür, wie die Aufgabe gelöst werden könnte
 - Verstehen der verwendeten Konzepte in den Beispiellösungen
 - Entwurfsentscheidungen versuchen zu verstehen
 - Das Vorgehen nachvollziehen und verstehen

Unterstützung während des Semesters

Tutorien

- Möglichkeit zum Nachfragen
- Übungsblatt vor- und nacharbeiten
- Vorbereitung auf Präsenzübung
- s.kit.edu/programmieren

Info²

- Info² und Lerngruppenbörse
- Mentoring und Kennenlernen Anderer
- Fachliche und persönliche Unterstützung
- Studienplanung Hilfsmittel

MINT

- Wiederholung des Vorlesungsinhalt
- Begrenzte Teilnehmerzahl
- Aufzeichnungen im ILIAS
- s.kit.edu/mint

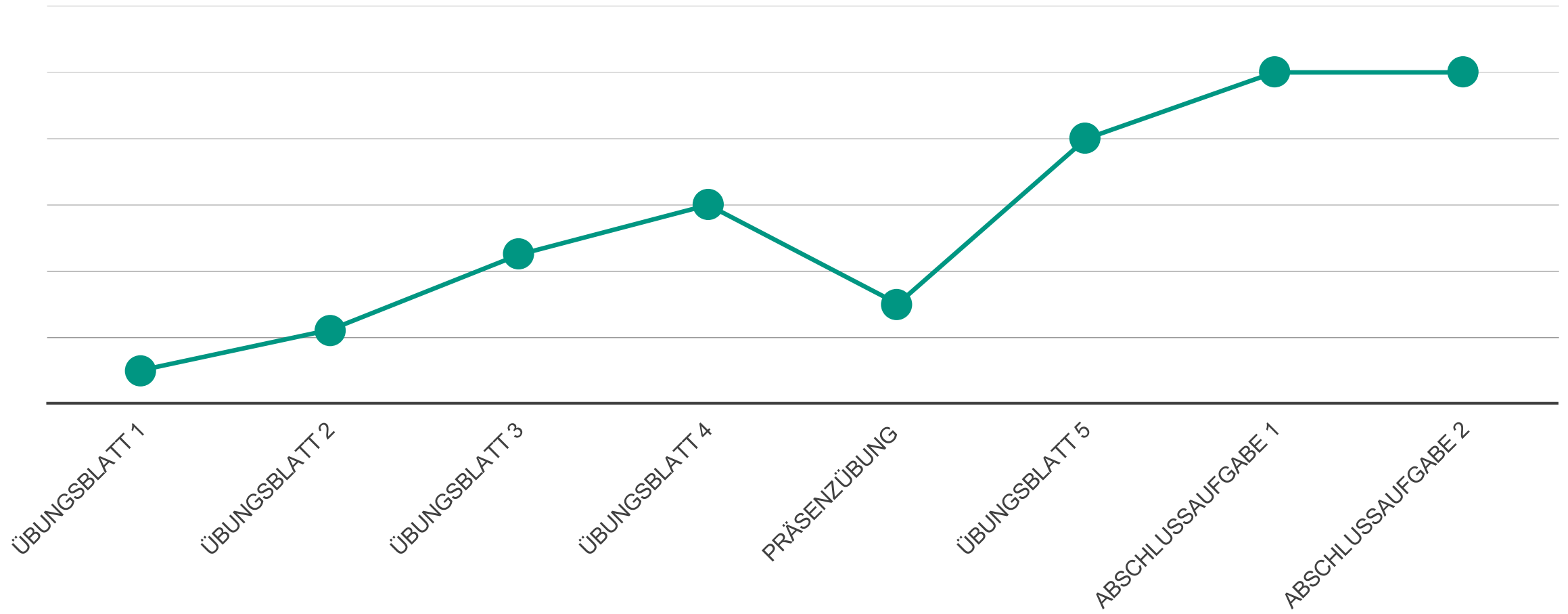
Forum-Wi / (Aktive) Fachschaft

- Vielfältigen Angebot auf dem Campus
- Klausuren und Prüfungsprotokollen
- Fachliche und persönliche Unterstützung
- s.kit.edu/fsmi ■ s.kit.edu/forum

Programmieren Präsenzübung

- Prüft Wissen, welches bei der Bearbeitung der Übungsblätter erworben worden sein sollte
 - Kleine Quelltextausschnitte schreiben oder ergänzen
 - Verhalten von kleinen Quelltextausschnitten analysieren
 - Verbeißen Sie sich nicht zu lange in eine einzige Aufgabe
- Sie sind bereits gut vorbereitet, wenn Sie die Übungsaufgaben gewissenhaft und erfolgreich gelöst haben!
 - Kein besonderes Hintergrundwissen nötig
 - Keine Vorgaben zu Checkstyle-Regeln
- Zur Vorbereitung auf das „Programmieren auf Papier“
 - Frischen Sie die Java-Syntax auf
 - Nicht die Folien auswendig lernen

Subjektive Schwierigkeit



Übungsblätter und Abschlussaufgaben

Übungsschein nicht erworben?

- Vorlesung beginnt bei null
- Übungsblätter orientieren sich an der Vorlesung
- Sehr steile Lernkurve
 - Sehr anspruchsvoll, wenn keine Vorkenntnisse vorhanden sind
 - Fortgeschrittene Studenten sollten für die Methodik am Ball bleiben

Übungsschein bereits erworben?

- Übungsblätter als erneute Vorbereitung nutzen
- Das Abgabesystem anschauen
- Programme selbstständig implementieren
 - Aufgaben aus dem Internet
 - Algorithmen aus den Mathematikvorlesungen

Organisatorische Tipps für die Übungsblätter

- Übungsblätter als Vorbereitung nutzen
 - Durch das eigenständige Bearbeiten lernt man am meisten
 - Praktisch fast kein zusätzlicher Lernaufwand für die Präsenzübung
 - Das letzte Übungsblatt als Vorbereitung auf die Abschlussaufgaben nutzen
- Früh genug anfangen, die Übungsblätter zu bearbeiten
 - Unklarheiten können so frühzeitig beseitigt werden
 - Die Übungsblätter werden schnell deutlich anspruchsvoller
 - Auf den ersten Blättern so viele Punkte wie möglich sammeln
- Früh genug anfangen, die Übungsblätter abzugeben
 - Frühzeitiges erproben von Git und Artemis
 - Lösungen können beliebig häufig hochgeladen werden
 - Tutor sieht nur die letzte gültige Abgabe

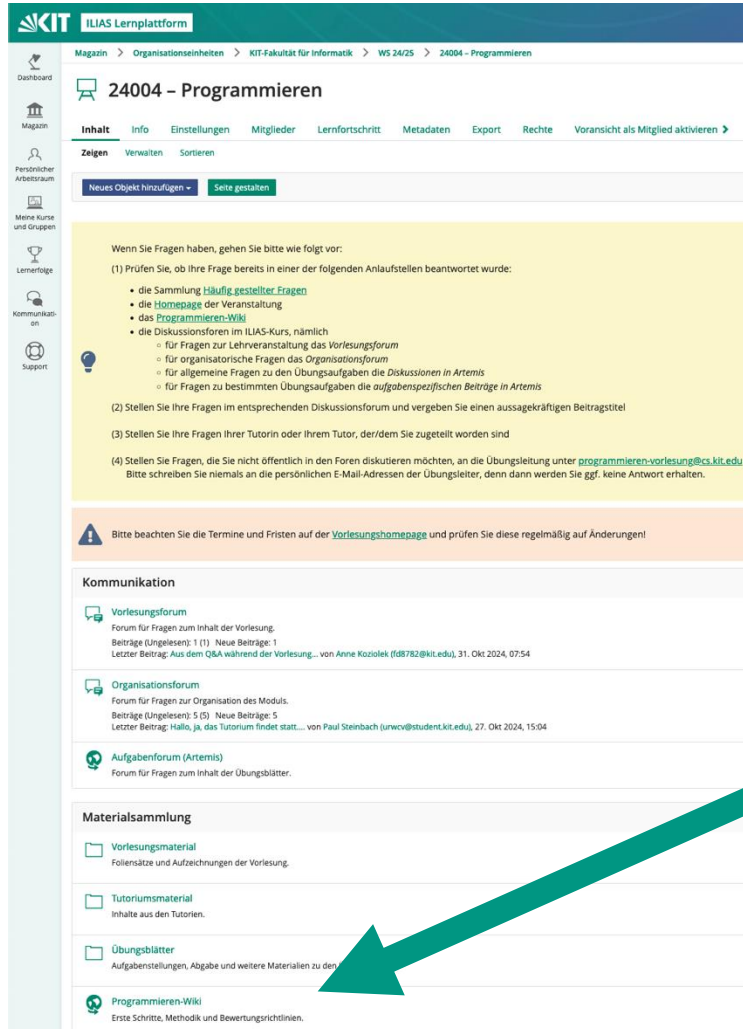
Praktische Tipps für die Aufgaben

- Quelltextklone vermeiden
 - Sich nicht Wiederholen und alles nur einmal im Quelltext implementieren
 - Aus doppelten Quelltextstellen versuchen, Methoden zu extrahieren
- Lange Methoden vermeiden
 - Methoden zerteilen und auf mehrere kleine (private) Hilfsmethoden verteilen
 - Dokumentation der Hilfsmethoden durch Javadoc-Kommentare
- Over-Engineering vermeiden
 - Quelltext so einfach wie möglich halten
 - Keine nicht benötigte Funktionalität
- Trennung der Anliegen
 - Datenstrukturen nicht nach außen geben
 - Trennung von Anwendungslogik und Benutzungsschnittstelle
- Geeignete Datentypen verwenden
 - Enums bei abgeschlossenen Mengen
 - Primitive Datentypen verwenden
 - Modellierung komplexer Typen mit Klassen
- Herangehensweise
 - Aufgabestellung genau lesen
 - Exakt die erwartete Ausgabe liefern
 - Bei Unklarheiten sofort nachfragen

Methodik Abzüge

- Neben Funktionalität wird auch Methodik geprüft
- Großer Teil der Bewertung von Übungsblättern und Abschlussaufgaben
- Alle Bewertungskriterien sind im Wiki: sdq.kastel.kit.edu/programmieren

Ilias-Forum



24004 – Programmieren

Inhalt Info Einstellungen Mitglieder Lernfortschritt Metadaten Export Rechte Voransicht als Mitglied aktivieren

Neues Objekt hinzufügen Seite gestalten

Wenn Sie Fragen haben, gehen Sie bitte wie folgt vor:

- Prüfen Sie, ob Ihre Frage bereits in einer der folgenden Anlaufstellen beantwortet wurde:
 - die Sammlung [Häufig gestellter Fragen](#)
 - die [Homepage](#) der Veranstaltung
 - das [Programmieren-Wiki](#)
 - die Diskussionsforen im ILLIAS-Kurs, nämlich
 - für Fragen zur Lehrveranstaltung das [Vorlesungsforum](#)
 - für organisatorische Fragen das [Organisationsforum](#)
 - für allgemeine Fragen zu den Übungsaufgaben die [Diskussionen in Artemis](#)
 - für Fragen zu bestimmten Übungsaufgaben die [aufgaben-spezifischen Beiträge in Artemis](#)
- Stellen Sie Ihre Fragen im entsprechenden Diskussionsforum und vergeben Sie einen aussagekräftigen Beitragstitel
- Stellen Sie Ihre Fragen Ihrer Tutorin oder Ihrem Tutor, der/dem Sie zugeteilt worden sind
- Stellen Sie Fragen, die Sie nicht öffentlich in den Foren diskutieren möchten, an die Übungsleitung unter programmieren-vorlesung@cs.kit.edu. Bitte schreiben Sie niemals an die persönlichen E-Mail-Adressen der Übungsleiter, denn dann werden Sie ggf. keine Antwort erhalten.

Bitte beachten Sie die Termine und Fristen auf der [Vorlesungshomepage](#) und prüfen Sie diese regelmäßig auf Änderungen!

Kommunikation

- Vorlesungsforum**
Forum für Fragen zum Inhalt der Vorlesung.
Beiträge (Ungelesen): 1 (1) Neue Beiträge: 1
Letzter Beitrag: Aus dem Q&A während der Vorlesung... von Anne Koziolek (08782@kit.edu), 31. Okt 2024, 07:54
- Organisationsforum**
Forum für Fragen zur Organisation des Moduls.
Beiträge (Ungelesen): 5 (5) Neue Beiträge: 5
Letzter Beitrag: Hallo, ja, das Tutorium findet statt... von Paul Steinbach (urwcv@student.kit.edu), 27. Okt 2024, 15:04
- Aufgabenforum (Artemis)**
Forum für Fragen zum Inhalt der Übungsblätter.


Materialsammlung


- Vorlesungsmaterial**
Foliensätze und Aufzeichnungen der Vorlesung.
- Tutoriumsmaterial**
Inhalte aus den Tutorien.
- Übungsblätter**
Aufgabenstellungen, Abgabe und weitere Materialien zu den Übungsblättern.
- Programmieren-Wiki**
Erste Schritte, Methodik und Bewertungsrichtlinien.


Materialsammlung

 **Vorlesungsmaterial**
Foliensätze und Aufzeichnungen der Vorlesung.

 **Tutoriumsmaterial**
Inhalte aus den Tutorien.

 **Übungsblätter**
Aufgabenstellungen, Abgabe und weitere Materialien zu den Übungsblättern.

 **Programmieren-Wiki**
Erste Schritte, Methodik und Bewertungsrichtlinien.

 **Vorlesungsaufzeichnungen WiSe 24/25**
Kursnummer: WS242424004 ; Ilias ID: 2476137 ; Semester: WS 24/25 ; CampusID

Wiki: Übersicht

Hauptseite

[Hauptseite](#) [Diskussion](#)

[Lesen](#) [Formular anzeigen](#) [Quelltext anzeigen](#) [Versionsgeschichte](#)

Beschreibung

Dieses Wiki wurde im Sommersemester 2024 erstellt und erweitert, das vorausgegangene Ilias-Wiki, das ab 2013/14 von Lehrenden für diese Vorlesung erstellt und instand gehalten wurde.

Aufgrund dessen, dass dieses Wiki sehr neu ist, kann es sein, dass sich Fehler eingeschlichen haben, die bislang nicht gefunden und korrigiert wurden. Diese dürfen gerne über das [Artemis-Forum](#) weitergeleitet werden, dass diese dann zeitnah korrigiert werden. Auch Ergänzungswünsche sind gerne gesehen, seien es zusätzliche Informationen oder gar neue Artikel. Das Wiki lebt von dem Feedback der Studierenden und wird dadurch nur besser.

Zur Installation von Java s. [Java](#).

Im Folgenden werden die aus unserer Sicht wichtigsten, allerdings nicht alle, Artikel aufgelistet (diese sind dann über die Suchleiste oben zu erreichen):

Organisation	[Einklappen] Grundlagen	[Einklappen] Bewertungsrichtlinien	[Einklappen]
<ul style="list-style-type: none"> – Artemis – Beratungsstellen – FAQ – Nutzung Testfälle – Präsenzübung 	<ul style="list-style-type: none"> – Abstrakt – Algorithmik – Checkstyle – Command Pattern – Datentypen – Debugging – Dokumentation – Enum – Exceptions Kontrollfluss 	<ul style="list-style-type: none"> Blatt 1 – Dokumentation – Einheitliche Sprache – Komplexität – Leerer Block/Leerer Konstruktor – Nur Main – Scanner – Schlechter Bezeichner – Schwieriger Code 	

Wiki: Bewertungsrichtlinien

Bewertungsrichtlinien

[Einklappen]

Blatt 1 Abzüge

- Einheitliche Sprache
- Systemabhängiger Zeilenumbruch

Blatt 2 Abzüge

- Dokumentation
- Instanceof außerhalb der equals-Methode
- Komplexität
- Leerer Block/Leerer Konstruktor
- Nur Main
- Schlechter Bezeichner
- Schwieriger Code
- Unbenutztes Element

Blatt 3 Abzüge

- Bedeutungslose Konstanten
- Datenkapselung
- Duplikate
- Enum
- Fehlermeldungen
- Final
- Hilfsklasse

- JavaDoc
- Konstanten-Klasse
- Magic Literal
- Object statt konkreter Klasse
- Pakete
- Polymorphie
- Raw Types
- Reimplementierung
- Runtime Exceptions
- Sichtbarkeit
- Stringreferenzen
- Ungeeigneter Schleifentyp

Blatt 4 Abzüge

- Cast außerhalb der equals-Methode
- Exceptions Kontrollfluss
- Große Try-Catch Blöcke
- Interface statt konkreter Klasse
- Statische Methoden und Attribute
- Verweigte Vererbung



Alle gelten für die
Abschlussaufgaben!

Wiki: Einheitliche Sprache

Seite Diskussion Lesen Mit Formular bearbeiten Bearbeiten Quelltext bearbeiten Versionsgeschichte ⭐ Weitere ▾

☹️ Diese Seite ist eine Bewertungsrichtlinie, die ab Blatt 1 annotiert und ab Blatt 1 abgezogen wird. ☹️

Beschreibung [Bearbeiten | Quelltext bearbeiten]

Es kann sehr anstrengend für eine Person werden, die den Quellcode verstehen möchte, Kommentare in unterschiedlichen Sprachen zu lesen. Auch gibt es das Problem der Sprachbarriere. Kommentare in einer unbekanntenen Sprache sind demnach unverständlich und nicht hilfreich. Sind Kommentare in einheitlicher, teils vorgegebener Sprache, kann sichergestellt werden, dass alle in dem Projekt involvierten Personen in der Lage sind, die Kommentare zu lesen und verstehen. Für Kommentare und Ein-/Ausgabe gibt es verschiedene Regelungen:

JavaDoc und Kommentare

JavaDoc und Entwicklerkommentare sollen in diesem Übungsbetrieb auf Deutsch oder Englisch verfasst werden. Hierbei soll es über das gesamte Projekt einheitlich sein. Auch zwischen verschiedenen Kommentartypen. Welche Sprache das ist, darf selbst ausgesucht werden.

Ein-/Ausgabe

Auch für die Ein-/Ausgabe soll auch eine einheitliche Sprache gewählt werden. Für Ausgaben ist zu beachten, dass diese, wie andere String-Literale auch, als Programmcode zählen und demnach auf Englisch verfasst werden sollten. Es allerdings Aufgaben geben, in denen eine deutsche Ausgabe gefordert wird. In diesem Fall sind die Ausgaben auf Deutsch zu verfassen.

Randnotiz

Die Ausgabe und JavaDoc/Kommentare dürfen in unterschiedlicher Sprache verfasst werden. Sprich: Deutsche Kommentare mit englischer Ausgabe sind okay, sofern die Aufgabenstellung keine deutsche Ausgabe verlangt. Auch ist zu beachten, dass sich diese Regel wirklich nur auf Kommentare und JavaDoc bezieht. Quellcode und demnach auch Bezeichner sollen nach wie vor auf Englisch verfasst werden (siehe [Schlechter Bezeichner](#)).

Das wäre ein **leichtes Vergehen**.

Wiki: Einheitliche Sprache

1 Negativbeispiel

```
1  /**
2  * Prints the status of the player to the console.
3  * @param player The id of the player
4  */
5  public void printPlayerStatus(int playerId) {
6      // Falls der Spieler unbekannt ist wird null zurückgegeben
7      Player player = findPlayerById(playerId);
8
9      if (player == null) {
10         System.err.println("Error, unbekannter Spieler");
11     } else {
12         System.out.println("Player " + playerId + ": " + player.getHP() + "HP");
13     }
14 }
```

Wiki: Einheitliche Sprache

2 Positivbeispiel

```
1  /**
2  * Prints the status of the player to the console.
3  * @param player The id of the player
4  */
5  public void printPlayerStatus(int playerId) {
6      // Returns null if the player is unknown
7      Player player = findPlayerById(playerId);
8
9      if (player == null) {
10         System.err.println("Error, unknown player");
11     } else {
12         System.out.println("Player " + playerId + ": " + player.getHP() + "HP");
13     }
14 }
```


Schlechte Bezeichner

- Bezeichner sollten *sprechend* sein
- Keine Abkürzungen, konform zu Konventionen
- Ausnahmen: i, j, x, y, it

```
1 public static int s(final int a, final int b) {  
2     return a - b;  
3 }
```



```
1 public static int subtract(final int minuend, final int subtrahend) {  
2     return minuend - subtrahend;  
3 }
```



Final

- Attribute sollten immer, wenn möglich, final sein
- Verhindert unabsichtliche Änderungen der Attribute (nicht ihrer Zustände)


```
class Event {  
    private static int BASE_FEE = 1000;  
    private List<String> participants;  
    public Event() {  
        participants = new ArrayList<>();  
    }  
    public void addParticipant(String name) {  
        participants.add(name);  
    }  
    public int calculateProfit(int ticketPrice, int venueCost) {  
        int fixedCost = BASE_FEE + venueCost;  
        return participants.size() * ticketPrice - fixedCost;  
    }  
}
```



Final

- Attribute sollten immer, wenn möglich, final sein
- Verhindert unabsichtliche Änderungen der Attribute (nicht ihrer Zustände)

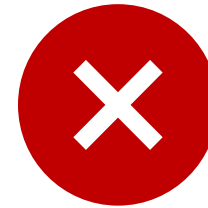
```
class Event {  
    private final static int BASE_FEE = 1000; // constants should be final  
    private final List<String> participants; // fields should be final if possible  
    public Event() {  
        participants = new ArrayList<>();  
    }  
    public void addParticipant(String name) { // not required for parameters or local variables  
        participants.add(name); // (list is final, but not the content)  
    }  
    public int calculateProfit(int ticketPrice, int venueCost) {  
        int fixedCost = BASE_FEE + venueCost;  
        return participants.size() * ticketPrice - fixedCost;  
    }  
}
```



Ungeeigneter Schleifentyp

- Boolescher-Ausdruck => While-Schleife, Do-While-Schleife
- Einfache Iteration => For-Each-Schleife
- Index benötigt => For-Schleife

```
1 public void printList(List<String> list){  
2     for(int i = 0; i<list.size(); i++){  
3         System.out.println(list.get(i));  
4     }  
5 }
```



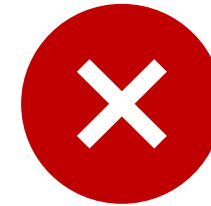
```
1 public void printList(List<String> list){  
2     for(String item : list){  
3         System.out.println(item);  
4     }  
5 }
```



Unnötige Komplexität

- Redundanz/Komplexität erhöht visuelles Rauschen
=> einfachste Lösung wählen

```
1  boolean isValid() {  
2      if (this.sold == false && !(this.price <= 0)) {  
3          return true;  
4      } else {  
5          return false;  
6      }  
7  }
```



```
1  boolean isValid() {  
2      return !this.sold && this.price > 0;  
3  }
```



Magic Numbers/Strings

- Alleinstehende Zahlen im Quelltext sind oft schwer zu verstehen
- Variablen erlauben Benennung
- Konstanten erlauben Wiederverwendung

```
1 final String password = "Test";  
2 if (password.length() < 8) {  
3     System.err.println("Password too short, please choose a longer password!");  
4 }
```




```
1 final String password = "Test";  
2 final int minimalPasswordLength = 8;  
3 if (password.length() < minimalPasswordLength) {  
4     System.err.println(PASSWORD_ERROR_MESSAGE);  
5 }
```



Sichtbarkeit und Datenkapselung

- Die Sichtbarkeit sollte so restriktiv wie möglich gesetzt werden
- Interner Zustand soll nur, falls wirklich nötig, herausgegeben werden
- Die Möglichkeit zu ungeplanten Zugriff führt zu Bugs


```
public final class Book {  
    public String title;  
    long isbn;  
  
    protected Book(String title, long isbn) { // protected sinnlos, da Klasse final  
        if (!isIsbnValid(isbn)) {  
            throw new InvalidIsbnException("The isbn is not valid");  
        }  
        this.title = title;  
        this.isbn = isbn;  
    }  
  
    public boolean isIsbnValid(long isbn) { // interne Hilfsmethode  
        // ...  
    }  
}
```



Sichtbarkeit und Datenkapselung

- Die Sichtbarkeit sollte so restriktiv wie möglich gesetzt werden
- Interner Zustand soll nur, falls wirklich nötig, herausgegeben werden
- Die Möglichkeit zu ungeplanten Zugriff führt zu Bugs

```
public final class Book {  
    private String title;  
    private long isbn;  
    public Book(String title, long isbn) {  
        if (!isIsbnValid(isbn)) {  
            throw new InvalidIsbnException("The isbn is not valid");  
        }  
        this.title = title;  
        this.isbn = isbn;  
    }  
    public String getTitle() { return this.title; }  
    public long getIsbn() { return this.isbn; }  
    private boolean isIsbnValid(long isbn) {  
        // ...  
    }  
}
```



Testen

- Testen hilft, Fehler zu finden!
 - Vergleich von tatsächlichem und erwünschtem Verhalten
 - Modularer Quelltext hilft beim Testen
- Ausführliches Testen vordem Hochladen
 - Nicht auf den Public-Test verlassen
 - Den Public-Test lokal ausführen
- Selber Tests schreiben
 - Für die Kommandozeileninteraktion
 - Über eigene Main-Methode
 - JUnit-Tests schreiben

Teststrategien

- Datenbasiert
 - Beispiele auf Aufgabenblättern
- Kontrollflussbasiert
 - Suche in Datenstrukturen
 - Alle Entscheidungen einmal treffen
- Grenzwertbasiert
 - Wertebereich bei Berechnungen
 - Off-by-one-Error

Rückfragen

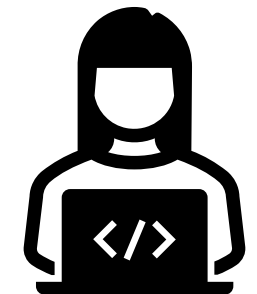
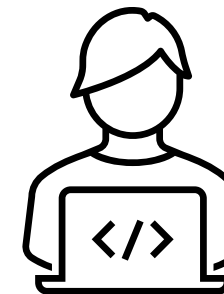
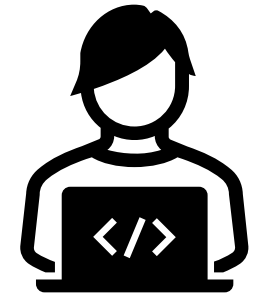
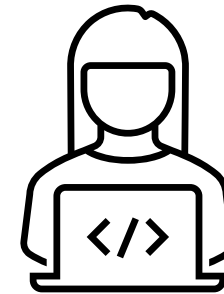
- Fragen und Unklarheiten können immer auftreten
 - Wenn ihr eine Frage habt, haben andere vielleicht die gleiche Frage
 - Wir bieten viele Möglichkeiten, Fragen zu stellen
- Bei Rückfragen werden Aufgaben gegebenenfalls aktualisiert
 - Lediglich minimale Aktualisierungen oder weitere Klarstellungen
 - Schaut immer mal wieder, ob die aktuelle Aufgabe aktualisiert wurde

Reihenfolge bei Fragen

1. Websuche
2. FAQ / Programmieren-Wiki prüfen
 - s.kit.edu/faq
 - sdq.kastel.kit.edu/programmieren
3. Diskussionsforum prüfen
 - Suchfunktion verwenden
4. Im Diskussionsforum nachfragen
 - Aussagekräftigen Titel verwenden
5. Eure Tutoren fragen
6. Übungsleitung fragen
 - programmieren-vorlesung@cs.kit.edu

Zusammenfassung

- Nehmen Sie sich **ausreichend Zeit** für die Übungsblätter
- Fangen Sie **rechtzeitig** mit der Bearbeitung an
- Beachten Sie die **Bewertungskriterien** im Ilias-Wiki
- Verwenden Sie früh **Checkstyle**
- Besuchen Sie die **Tutorien** und nehmen Sie aktiv Teil
- Üben Sie für die Präsenzübung **auf Papier**



Q&A – Panel



Überblick über das zweite Semester & SP026

Was ist die SPO?

Die Studienprüfungsordnung (SPO) regelt, ...

- Welche Fächer für einen Studiengang verpflichtend sind
- Wie viele Leistungspunkte Fächer geben
- Wann Fächer angeboten werden

An WS 26/27 es eine neue Informatik SPO.

- Alle, die jetzt Informatik studieren, sind in der SPO 2022 (oder 2015)
- Wer möchte, kann aber in die neue SPO wechseln, mehr dazu später

Überblick SPO 2022

Sem.	Grundlagenstudium				Wahlbereich 15 - 31 LP	Ergänzungsfach 12 - 21 LP	Überfachliche Qualifikationen 6 LP
Fach	Theoretische Informatik 18 LP	Praktische Informatik 48 LP	Mathematik 38 - 45 LP	Technische Informatik 12 LP			
1	Grundbegriffe der Informatik 6 LP	Programmieren 5 LP	Höhere Mathematik I ¹ 9 LP / Analysis I ¹ 9 LP				
			Lineare Algebra I für Informatik ² 9 LP / Lineare Algebra I ² 9 LP				
2	Algorithmen I 6 LP	Softwaretechnik I 6 LP	Höhere Mathematik II ¹ 6 LP / Analysis II ¹ 9 LP	Digitaltechnik & Entwurfsverfahren ³ 6 LP			
			Lineare Algebra II für Informatik ² 5 LP / Lineare Algebra II ² 9 LP				
3	Theoretische Grundlagen der Informatik 6 LP	Praxis der Softwareentwicklung 7 LP	Wahrscheinlichkeitstheorie & Statistik ⁴ 4,5 LP	Rechnerorganisation ³ 6 LP			
		Betriebssysteme 6 LP					
4		Kommunikation & Datenhaltung 8 LP	Numerische Mathematik ⁴ 4,5 LP		Proseminar 3 LP + Stammmodul 6 LP + Sonstige Wahlmodule		
		Informationssicherheit 5 LP					
5		Programmierparadigmen 6 LP				Elektrotechnik / Mathematik / Physik / Recht / Maschinenbau / Wirtschaftswissenschaften	
		Grundlagen der künstlichen Intelligenz 5 LP					
6	Bachelorarbeit 15 LP						Teamarbeit in der Softwareentwicklung 2 LP + Sonstige Schlüsselqualifikationen 4 LP

¹ Die Veranstaltungen Höhere Mathematik I und Höhere Mathematik II sind dem Modul Höher Mathematik zugeordnet. Die Veranstaltungen Analysis I und Analysis II sind dem Modul Analysis 1 und 2 zugeordnet.

² Die Veranstaltungen Lineare Algebra I für Informatik und Lineare Algebra II für Informatik sind dem Modul Lineare Algebra für die Fachrichtung Informatik zugeordnet. Die Veranstaltungen Lineare Algebra I und Lineare Algebra II sind dem Modul Lineare Algebra I und II zugeordnet.

³ Die Veranstaltungen Digitaltechnik & Entwurfsverfahren und Rechnerorganisation sind dem Modul Technische Informatik zugeordnet.

⁴ Die Veranstaltungen Wahrscheinlichkeitstheorie & Statistik und Numerische Mathematik sind dem Modul Praktische Mathematik zugeordnet.

Tabelle 1: Struktur des Bachelorstudiengangs Informatik.

Überblick SPO 2026

Sem.	Grundlagenstudium					Wahlbereich 15 - 36 LP	Ergänzungsfach 12 - 21 LP	Überfachliche Qualifikationen 6 LP	
Fach	Theoretische Informatik 17 LP	Softwareentwicklung 24 LP	Weitere Kernfächer 20 LP	Mathematik 34 - 46 LP	Technische Informatik 16 LP				
1	Grundlagen der theoretischen Informatik 6 LP	Programmieren 1 5 LP		Lineare Algebra I für Informatiker ² 9 LP / Lineare Algebra I ² 9 LP	Digitaltechnik und Rechnerentwurf 6 LP			Ethik 2 LP	
2	Algorithmen I 6 LP	Softwaretechnik I 5 LP		Lineare Algebra II für Informatiker ² 5 LP / Lineare Algebra II ² 9 LP (Optional)	Rechnerorganisation 5 LP				
		Programmieren 2 5 LP		Höhere Mathematik I ¹ 9 LP / Anaysis I ¹ 9 LP					
3	Berechenbarkeit und Komplexität 5 LP	Praxis der Softwareentwicklung 9 LP		Wahrscheinlichkeitstheorie & Statistik 5 LP	Betriebssysteme 5 LP				
									Höhere Mathematik II ¹ 6 LP / Analysis II ¹ 9 LP
4			Datenbanksysteme 5 LP	Numerische Mathematik 5 LP		Proseminar 3 LP + Stammmodul 6 LP + Sonstige Wahlmodule			
			Einführung in Rechnernetze 5 LP						
5			Grundlagen der künstlichen Intelligenz 5 LP					Schlüsselqualifikationen 4 LP	
			Informationssicherheit 5 LP						
6	Bachelorarbeit 15 LP								

¹ Die Veranstaltungen Höhere Mathematik I und Höhere Mathematik II sind dem Modul Höhere Mathematik zugeordnet. Die Veranstaltungen Analysis I und Analysis II sind dem Modul Analysis 1 und 2 zugeordnet.
² Die Veranstaltungen Lineare Algebra I für Informatiker und Lineare Algebra II für Informatiker sind dem Modul Lineare Algebra für die Fachrichtung Informatik zugeordnet. Die Veranstaltungen Lineare Algebra I und Lineare Algebra II sind dem Modul Lineare Algebra I und II zugeordnet. Das Modul Lineare Algebra II für Informatiker oder Lineare Algebra II ist optional.

1. Semester

SPO 2022

- Grundbegriffe der Informatik (6 LP)
 - Verpflichtender Übungsschein
 - **Orientierungsprüfung**
- Programmieren (5LP)
 - Verpflichtender Übungsschein
 - **Orientierungsprüfung**
- Lineare Algebra 1 (für Informatik) (9 LP)
 - Verpflichtender Übungsschein, der Voraussetzung für die Klausur ist
 - **Orientierungsprüfung**
- Höhere Mathematik 1 (9 LP)
 - Nur Vorlesung, Klausur zusammen mit HM2 im WS
 - Übungsschein von HM1 oder HM2 ist Voraussetzung für die Klausur

SPO 2026

- **Grundlagen der theoretischen Informatik (6 LP)**
 - Inhalte vergleichbar zu GBI
 - Verpflichtender Übungsschein
 - **Orientierungsprüfung**
- Programmieren 1 (5 LP)
 - Verpflichtender Übungsschein
 - **Orientierungsprüfung**
- Lineare Algebra 1 (für Informatik) (9 LP)
 - Verpflichtender Übungsschein, **der nicht Voraussetzung für die Klausur ist**
 - **Orientierungsprüfung**
- **Digitaltechnik und Rechnerentwurf (6 LP)**
 - Nur Vorlesung, Klausur zusammen mit Rechnerorganisation im SS
- **Ethik (2 LP)**

2. Semester

SPO 2022

- Algorithmen 1 (6 LP)
- Softwaretechnik 1 (6 LP)
 - Verpflichtender Übungsschein
- Lineare Algebra 2 (für Informatik) (5 LP)
- Höhere Mathematik 2 (6 LP)
 - Gemeinsame Klausur mit HM1
 - Übungsschein von HM1 oder HM2 ist Voraussetzung für die Klausur
- Digitaltechnik und Entwurfsverfahren (6 LP)
 - Nur Vorlesung, Klausur zusammen mit Rechnerorganisation im WS

SPO 2026

- Algorithmen 1 (6 LP)
- Softwaretechnik 1 (**5 LP**)
 - Verpflichtender Übungsschein
- **Programmieren 2 (5 LP)**
 - Ersetzt Programmierparadigmen im 5. Semester
- **Höhere Mathematik 1 (9 LP)**
 - Verpflichtender Übungsschein
 - Eigene Klausur, nicht zusammen mit HM2
- **Rechnerorganisation (5 LP)**
 - Gemeinsame Klausur mit Digitaltechnik im SS
- **Optional:** Lineare Algebra 2 (für Informatik) (5 LP)

3. Semester

SPO 2022

- Theoretische Grundlagen der Informatik (6 LP)
- Praxis der Softwareentwicklung (7 LP)
- Teamarbeit in der Softwareentwicklung (2 LP)
 - An PSE geknüpft
- Betriebssysteme (6 LP)
- Wahrscheinlichkeitstheorie und Statistik (4,5 LP)
- Rechnerorganisation (6 LP)
 - Gemeinsame Klausur mit Digitaltechnik

SPO 2026

- **Berechenbarkeit und Komplexität (5 LP)**
 - Inhalte vergleichbar zu TGI
- Praxis der Softwareentwicklung (9 LP)
- Betriebssysteme (5 LP)
- Wahrscheinlichkeitstheorie und Statistik (5 LP)
- **Höhere Mathematik 2 (6 LP)**
 - Verpflichtender Übungsschein
 - Eigene Klausur, nicht zusammen mit HM1

Sonstige Änderungen

- Einführung in Rechnernetze (4. Semester) geht von 4 auf 5 LP
- Datenbanksysteme (4. Semester) geht von 4 auf 5 LP
- Numerik (4. Semester) geht von 4,5 auf 5 LP
- Informationssicherheit rutscht vom 4. ins 5. Semester
- Wahlbereich wird größer: von 15-31 LP auf 15-36 LP
- Mehr Informationen: <https://www.informatik.kit.edu/15034.php>

KIT-Fakultät für Informatik

[Startseite](#) > [Studium](#) > [Studienorganisation](#) > [FAQ Studium und Prüfungen](#) > [FAQ](#) > [Bachelor Informatik SPO 2026](#)

Bachelor Informatik SPO 2026



Was passiert, wenn ich wechsle?

- **Lineare Algebra 1 und 2 müssen bestanden sein, damit man wechseln kann**

Wenn ich nach dem 2. Semester SPO wechsle...

- Muss ich Ethik im WS (z.B. 3. Semester) nachholen
- Muss ich Programmieren 2 im SS (z.B. 4. Semester) nachholen

Programmieren im Wintersemester 26/27

Übungsschein

- Übungsblätter (bleiben gleich)
- **Keine Präsenzübung**

Prüfung

- Abschlussaufgabe 1 und Abschlussaufgabe 2**
oder
- Abschlussaufgabe 1 und Test vor Ort**

Übergangsregelung (**Nur im WS 26/27**):

Man kann es sich aussuchen, ob man die neue oder alte Prüfungsversion schreibt

Überblick über das zweite Semester

Wie geht das Studium weiter?

Module im 2. Semester: Informatik

- Softwaretechnik 1
 - Algorithmen 1
 - LA2
 - HM2
 - Digitaltechnik und Entwurfsverfahren
-
- Die Module sind eine Empfehlung vom Modulhandbuch, aber keine Pflicht!

Module im 2. Semester: Winfo

- Wirtschaftsinformatik 2
- Algorithmen 1
- Mathe 2
- Statistik 1
- Einführung in das Operations Research (1)
- Recht (BGB für Fortgeschrittene)

Übersicht über die Übungsscheine

Modul	Übungsschein notwendig für Prüfung?	Semester
Programmieren	Ja	WS + SS
Softwaretechnik	Nein, aber um Modul zu bestehen	SS
Lineare Algebra II (für Info)	Ja	SS
Lineare Algebra II (Mathe)	Ja	SS
Höhere Mathematik II	Ja	SS
Analysis II	Ja	SS
Mathe II	Nein, aber um Modul zu bestehen	SS
Wirtschaftsinformatik II	Nein, nur Klausurbonus	SS

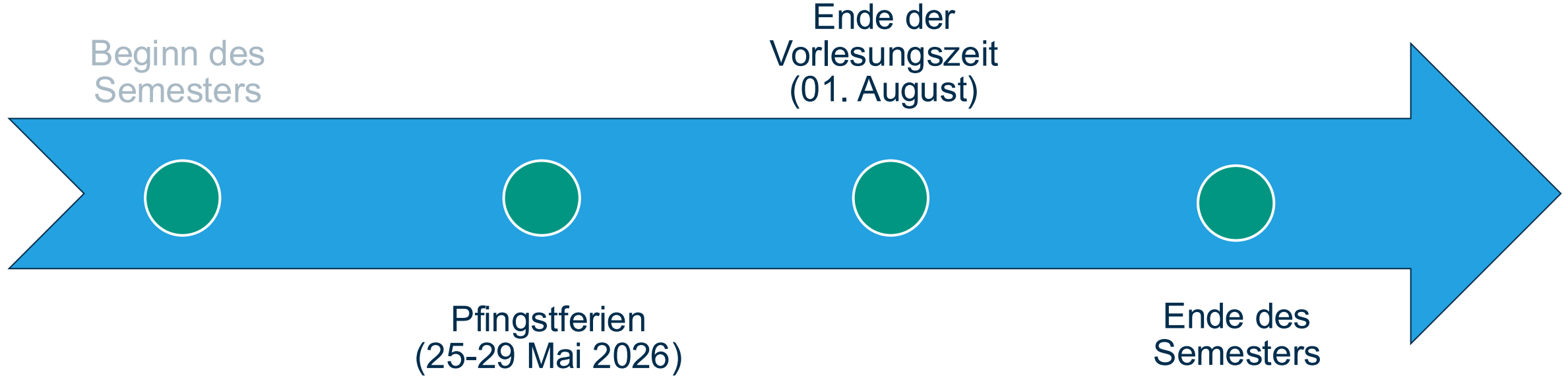
O-Prüfung nicht bestanden – Was tun?

- Nicht schlimm, das passiert vielen
- Erstversuch von O-Prüfungen spätestens im 2. Semester und Bestehen der O-Prüfungen spätestens im 3. Semester (Verlängerung durch MINT-Kolleg möglich)
- Programmieren Übungsschein + Abschlussprüfungen werden jedes Semester angeboten
- LA1 Klausur kann man auch mit dem LA2 Übungsschein schreiben
- Für die HM Klausur benötigt man nur den HM1, oder HM2 Übungsschein
- Prüfung werden in der Regel jedes Semester angeboten

Prüfungen SoSe 26

Informatik & Informatik Lehramt		Prüfungen				Wirtschaftsinformatik					
		Privatrechtliche Übung (während des Semesters)	13.05.2026 20.05.2026 24.06.2026 08.07.2026 22.07.2026	jeweils 15:45 - 17:15							
Days				Days							
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Ausgabe: 12.08.2026 (ca. 13:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Ausgabe: 26.08.2026 (ca. 13:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 1: Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Abschlussaufgabe 2: Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr) </div>				Programmieren (NT)		Abschlussaufgabe 1: Ausgabe: 12.08.2026 (ca. 13:00 Uhr), Abgabe: 26.08.2026 (12:00 Uhr) – 10.09.2026 (06:00 Uhr) Abschlussaufgabe 2: Ausgabe: 26.08.2026 (ca. 13:00 Uhr), Abgabe: 09.09.2026 (12:00 Uhr) – 24.09.2026 (06:00 Uhr)					
				Statistik 1		01.08.26	07:30 - 10:30	12	12		
				Softwaretechnik 1		03.08.26	08:00 - 10:00				
				Wirtschaftsinformatik 2		14.08.26	16:00 - 17:00				
		14		20		GBI (NT)		18.08.26	12:00 - 14:00	9	3
		5				Algorithmen 1		24.08.26	08:00 - 10:00	3	5
		7		7		Wirtschaftsinformatik 1 (NT)		28.08.26	08:00 - 09:00		
						LA 1 (NT) & 2		02.09.26	LA1: 08:00 - 10:00 LA2: 11:00 - 12:30	29	25
				12		Höhere Mathematik I & II		15.09.2026	12:00 - 17:00		
						Analysis 2		15.09.26	18:00 - 20:00		
						Analysis 1 (NT)		17.09.26	11:00 - 13:00		
						Mathematik 1 (NT)		24.09.26	10:00 - 11:30	3	
						Mathematik 2		28.09.26	08:00 - 9:30		
						Einführung in das Operations Research I		(Prüfung erst im WS)			

Zeit zu reflektieren



- Gut Reflektiert ist halb Gewonnen
 - „Fühlt sich das Studium für mich weiterhin richtig an?“
 - „Welche Fehler habe ich im ersten Semester gemacht und mache ich diese gerade wieder?“
- Ferien: Nimm dir Zeit für dich!
- Wieder zur Uni
 - Stundenplan überdenken
 - Stoff wieder einarbeiten
- Prüfungsvorbereitung beginnen

Ressourcen zur Reflexion

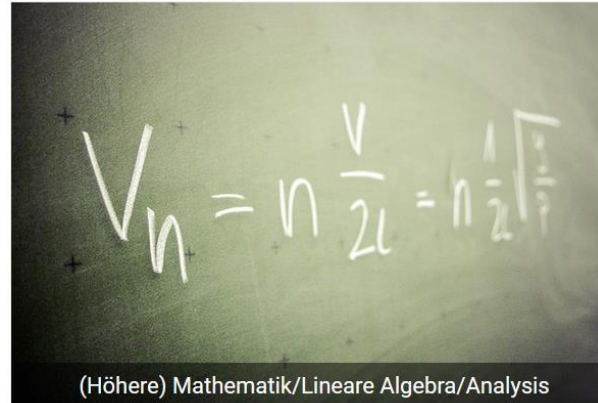
- Zentrale Studienberatung (ZSB)
- Angebote der Fakultät für Informatik
 - Informatik Studiengangservice (ISS)
 - eezi-Beratungs Gespräch
 - Fachschaft (FSMI) / Wi-Forum
- Mit Studierenden oder Tutoren aus höheren Fachsemestern reden
- Mit sozialem Umkreis und Familie reden

MINT-Kolleg

MINT-Kolleg: Semesterkurse, Helpdesk, Aufbaukurse

Bietet Semesterkurse an zu

- Höhere Mathematik I
- Höhere Mathematik II
- Lineare Algebra I
- Lineare Algebra II
- Programmieren



→ Im Ilias-Kurs können Ressourcen genutzt werden, auch wenn man keinen Kursplatz hat

*Im Sommer semester gibt keine Aufbaukurs

Weitere Informationen



Programmieren Übung SS26



Informatik Studiengangsservice

• ISS -FAQ

KIT-Fakultät für Informatik

Startseite > Studium > Studienorganisation > FAQ Studium und Prüfungen > FAQ > mehr

FAQs Studium und Prüfungen

In unseren FAQs sind erste Antworten auf Ihre Fragen zu den Studiengängen Informatik, Lehramt für Informatik und Bachelor Wirtschaftsinformatik zu finden. Masterstudierende des Studiengangs Wirtschaftsinformatik wenden sich an das [Prüfungssekretariat WiWi](#).

Information nicht gefunden? Sollten zum gewünschten Thema keine Informationen in den FAQs zu finden sein, senden Sie bitte eine E-Mail an beratung-informatik@informatik.kit.edu

Hier die FAQ-Themen:

Abschlussarbeiten Abschlussdokumente und Zeugnis Ärztliches Attest Anerkennung von Prüfungsleistungen Anträge Auslandsstudium / Erasmus	Interner Übergang vom Bachelor- zum Masterstudium am KIT Klausuren: Termine / Einsicht Lehramt LP-Verschnitt / Notenberechnung	Schülerstudium Seminare und Praktika im B.Sc. und M.Sc. Informatik SPO Bachelor Informatik 2022 Studienplan
Bachelor Informatik SPO26	Mastermodule: im Bachelor / erneut nach Studienortwechsel / Mastervorzug MINT-Kurse Orientierungsprüfung	Überfachliche Qualifikationen (Schlüsselqualifikationen) Überprüfungsmodul Praktika und Seminare (Master Informatik) Umbuchungen: Verschieben von Modulen und Prüfungen / Modulabbruch Urlaubssemester
BAföG Bescheinigungen allgemein/ Studienabschluss / Titelführungsbescheinigung / Mindestens bestanden Bewerbung für einen Studienplatz	Parallelstudium Programmierparadigmen Lehramt Proseminar (INFO) Prüfungen: An- und Abmeldung / Wiederholung Prüfungsausschuss Prüfungsleistungen / Studienleistungen Prüfungsplan Prüfungszeitraum	Wahlmodule Zusatzleistungen Zweitstudium Zweitwiederholung (auch Härtefallantrag) und Vorbehaltskonto
Doppelmaster		
ECTS-Einstufungstabellen Ergänzungsfach (B.Sc. und M.Sc. Informatik) Ersetzen von Modulen und Prüfungen Exmatrikulation		

Info2 - Youtube Playlist



Unterstützung im Studium
Youtube-Kanal

Unterstützung im Studium

von KIT Informatik

Playlist • 30 Videos • 1.220 Aufrufe

Das Programm Info² soll allen Studierenden an der KIT-Fakultät für Informatik in der Organisation ihres Studiums...

Alle abspielen

KIT Campus Management System ENG

Department of Informatics
Karlsruhe Institute of Technology (KIT)

5:28

Unterstützung im Studium – Website

Die ersten Semester

Orientierungsangebote und Unterstützung zu allen Themen rund um Ihren Studienbeginn.

MEHR

Studienplanung

Tipps zur erfolgreichen Planung Ihres Studiums.

MEHR

Mentoring

Angebote zur sozialen Einbindung am KIT.

MEHR

Info²

Info-Veranstaltungen & -Videos zur Studienorganisation und effektiven Lernmethoden.

MEHR

Internationales

Informationen für internationale Studierende am KIT sowie KIT-Studierende, die im Ausland studieren möchten.

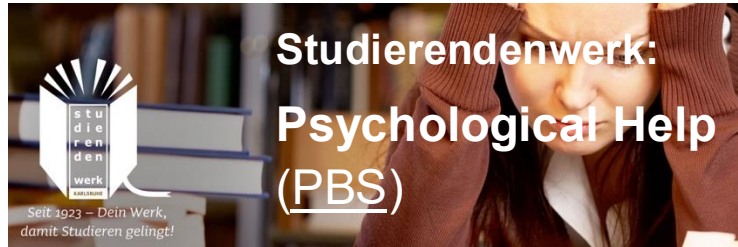
MEHR

Informationsveranstaltungen des ISS

ZUM EVENTKALENDER

	20.05.2026	Info²: Programmieren Best Practices + Lernpartnerschaftsbörse In der Veranstaltung stellt ein Mitglied des Übungsleiter-Teams des Moduls Programmieren nützliche Informationen und Tipps vor...
	17.06.2026	ISR2: Fit for exams and returning home Dieser ISR2 bietet wichtige Informationen für internationale Studierende zur Prüfungsphase und die am Ende dieses Semesters ihr Studium...

Mental health and dealing with stress



PBS

- With all the difficulties that affect the course of study such as:
 - Test anxiety - Concentration difficulties - Self-esteem problems?
 - Loneliness - Relationship conflicts - Fear of the future
- Studied therapists - Absolutely anonymous
- Everyone gets an appointment!



HoC for Internationals: Techniques and advice on all aspects of studying

- Courses
 - Advice
- Learning
 - Presentation & Communication
 - Writing
 - Shaping the future
 - Doing research

SCHLOSS LICHTSPIELE

Light Festival Karlsruhe



13 AUGUST TO 13 SEPTEMBER 2026 – ENTRANCE FREE!



Wahlbereich - Info

- Besteht aus Wahlmodulen, Stammmodul, Praktika (+ Proseminar)
- Insgesamt 15-31 ECTS
- Max. 9 ECTS aus Praktika und Seminaren (inkl. Proseminar)

Stammmodule (6 ECTS)

- Algorithms 2
- Computergrafik
- Formale Systeme
- Human Computer Interaction
- IT Security
- Machine Learning – Foundations and Algorithms
- Multimodal Artificial Intelligence
- Rechnerstrukturen
- Robotics 1
- Software Engineering 2
- Telematics

Ergänzungsfach: 12 bis 21 ECTS

- Recht
- Mathe
- Physik
- Elektro- und Informationstechnik
- BWL
- VWL
- Operations Research
- Philosophie
- Wirtschaftsinformatik
- Initiativantrag auf anderes Ergänzungsfach möglich: z.B. Musikinformatik an der Hochschule für Musik

- Zentrale Studienberatung (ZSB)
- Angebote der Fakultät für Informatik
 - Informatik Studiengangservice (ISS)
 - eezi-Beratungs Gespräch
 - Fachschaft (FSMI) / Wi-Forum
- Mit Studierenden oder Tutoren aus höheren Fachsemestern reden
- Mit sozialem Umkreis und Familie reden