

Programmieren Best Practices & Lernpartnerschaftsbörse

2. Info-Bite

Diestag, 15.11.2022 um 17:30 Uhr



```
1983 $opts_order = array();
1984 $results = $this->model->extension->
1985 foreach ($results as $key => $value) {
1986     if (isset($value['code'])) {
1987         $code = $value['code'];
1988     } else {
1989         $code = $value['key'];
1990     }
1991     $sort_order[$key] = $this->config->get($code . '_sort_order');
1992 }
1993 array_multisort($sort_order, SORT_ASC, $results);
1994
1995 foreach ($results as $result) {
1996     if (isset($result['code'])) {
1997         $code = $result['code'];
1998     } else {
1999         $code = $result['key'];
2000     }
2001     if ($this->config->get($code . '_status')) {
2002         $this->load->model('extension/total/' . $code);
2003         // We have to put the totals in an array so that they pass
2004         // by reference.
2005         $this->['model_extension_total_' . $code]->getTotal($
2006             total_data);
2007         if (empty($totals[count($totals) - 1]) && !isset($totals[
2008             count($totals) - 1]['code'])) {
2009             $totals[count($totals) - 1]['code'] = $code;
2010         }
2011     }
2012 }
2013
2014 Carousel.prototype.getItemIndex = function (active) {
2015     var delta = direction == 'prev' ? -1 : 1;
2016     var activeIndex = this.getItemIndex(active);
2017     var itemIndex = (activeIndex + delta) % this.$items.length;
2018     return this.$items.eq(itemIndex);
2019 }
2020
2021 Carousel.prototype.to = function (pos) {
2022     var that = this;
2023     var activeIndex = this.getItemIndex(this.$active);
2024     if (pos > (this.$items.length - 1) || pos < 0) return;
2025     if (this.sliding) return this.$element.one('slid.bs.carousel', function () { that.to(pos) });
2026     if (activeIndex == pos) return this.pause().cycle();
2027     return this.slide(pos > activeIndex ? 'next' : 'prev', this.$items.eq(pos));
2028 }
2029
2030 Carousel.prototype.pause = function (e) {
2031     e || (this.paused = true);
2032     if (this.$element.find('.next, .prev').length && $.support.transition) {
2033         this.$element.trigger($.support.transition.end);
2034         this.cycle(true);
2035     }
2036 }
```

Agenda

- Gruß und Übersicht
- Programmieren-Team
- Erfolgsmethoden im Modul Programmieren...
- VL
- Übungsblätter,
- Q & A
- Lernpartnerschaftsbörse

Best Practices für „Programmieren“

eezi goIN' Info-Bites | Timur Sağlam



Das Programmieren-Team

Die Leiter



Dozentin
Prof. Dr.-Ing. Anne Koziolk
Übungsleiter
Dominik Fuchß, Yves Kirschner & Timur Saçlam
programmieren-vorlesung@cs.kit.edu

Ihre Tutor:innen

- Alexander Kutschera
- Alexander Milster
- Alexander Vogt
- Anton S.
- Aurelia Hüll
- Bastian
- Christopher Hommel
- Clemens Dautermann
- Dennis Moschina
- Elias Dirks
- Emil Hammer
- Fabian Metzger
- Florian Seligmann
- Gregor Czubyko
- Jonas Ludwig
- Julian Keck
- Liam Wachter
- Linus Dierheimer
- Luca Pallo
- Lucas Alber
- Lucas Altenau
- Martin Krausewitz
- Mattis Justus Kaiser
- Maxime Rambaud
- Metehan Yayla
- Moritz Gstür
- Moritz Hertler
- Moritz Rimpf
- Noël
- Peter Bohner
- Piotr Malkowski
- Robert Brune
- Simon Wülker
- Till Bernas
- Tobias Dudziak
- Tobias Kempf
- Tobias Thirolf

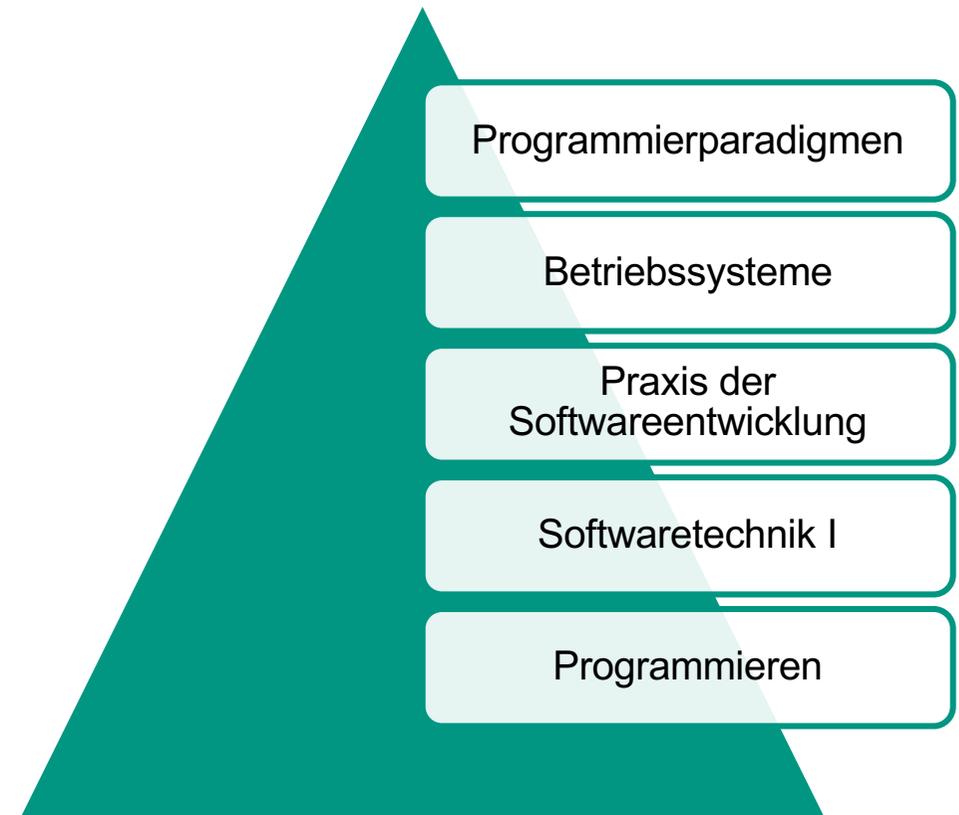
Wer seid Ihr? → Anonyme Umfrage in Zoom

Best Practices für „Programmieren“

- Wir freuen uns über Alle, die erfolgreich das Modul abschließt
 - Teilen von Erfahrung, woran es scheitern könnte
 - Tipps sollen Frust sparen und allen beste Chancen bieten
- Nicht alle Praktiken werden bei den ersten Blättern anwendbar sein
 - Es muss nicht unbedingt alles sofort verstanden werden
 - Aber spätestens bei den Abschlussaufgaben solle darauf zurückgegriffen werden

Grundlagenstudium praktische Informatik

- Programmieren bildet die Grundlage für weitere Module
- Fokus auf Orientierungsprüfung
- Programmieren sollte vor SWT abgeschlossen sein



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 Praxis 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

Termine und Fristen

- Anmeldung Übungsschein: 01.10.2022 - 07.12.2022, jeweils 12.00 Uhr
- Anmeldung Abschlussaufgaben: 13.02.2023 - 20.02.2023, jeweils 12.00 Uhr
- Präsenzübung am 13.01.2023 zwischen 17:30 - 19:30 Uhr
- Übungsblatt 1: 02.11.2022 ca. 12:00 Uhr - 17.11.2022, 06:00 Uhr
- Übungsblatt 2: 16.11.2022, ca. 12:00 Uhr - 01.12.2022, 06:00 Uhr
- Übungsblatt 3: 30.11.2022, ca. 12:00 Uhr - 15.12.2022, 06:00 Uhr
- Übungsblatt 4: 14.12.2022, ca. 12:00 Uhr - 12.01.2023, 06:00 Uhr
- Übungsblatt 5: 11.01.2023, ca. 12:00 Uhr - 26.01.2023, 06:00 Uhr
- Abschlussaufgabe 1: 13.02.2023, ca. 12:00 Uhr - 14.03.2023, 06:00 Uhr
- Abschlussaufgabe 2: 28.02.2023, ca. 12:00 Uhr - 29.03.2023, 06:00 Uhr
- Mögliche Aktualisierungen werden hier veröffentlicht: s.kit.edu/programmieren

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](https://www.s.kit.edu/programmieren)

MINT

- Wiederholung des Vorlesungsinhalt
- Begrenzte Teilnehmerzahl
- Aufzeichnungen im ILIAS
- [s.kit.edu/mint](https://www.s.kit.edu/mint)

eezi goIN'

- eezi Lehrveranstaltung
- Mentoring-Stammtische und Info-Bites
- Fachliche und persönliche Unterstützung
- Kennenlernen Anderer
- [s.kit.edu/studienstart](https://www.s.kit.edu/studienstart)

Forum-Wi / (Aktive) Fachschaft

- Vielfältigen Angebot auf dem Campus
- Klausuren und Prüfungsprotokollen
- Fachliche und persönliche Unterstützung
- [s.kit.edu/fsmi](https://www.s.kit.edu/fsmi)
- [s.kit.edu/forum](https://www.s.kit.edu/forum)

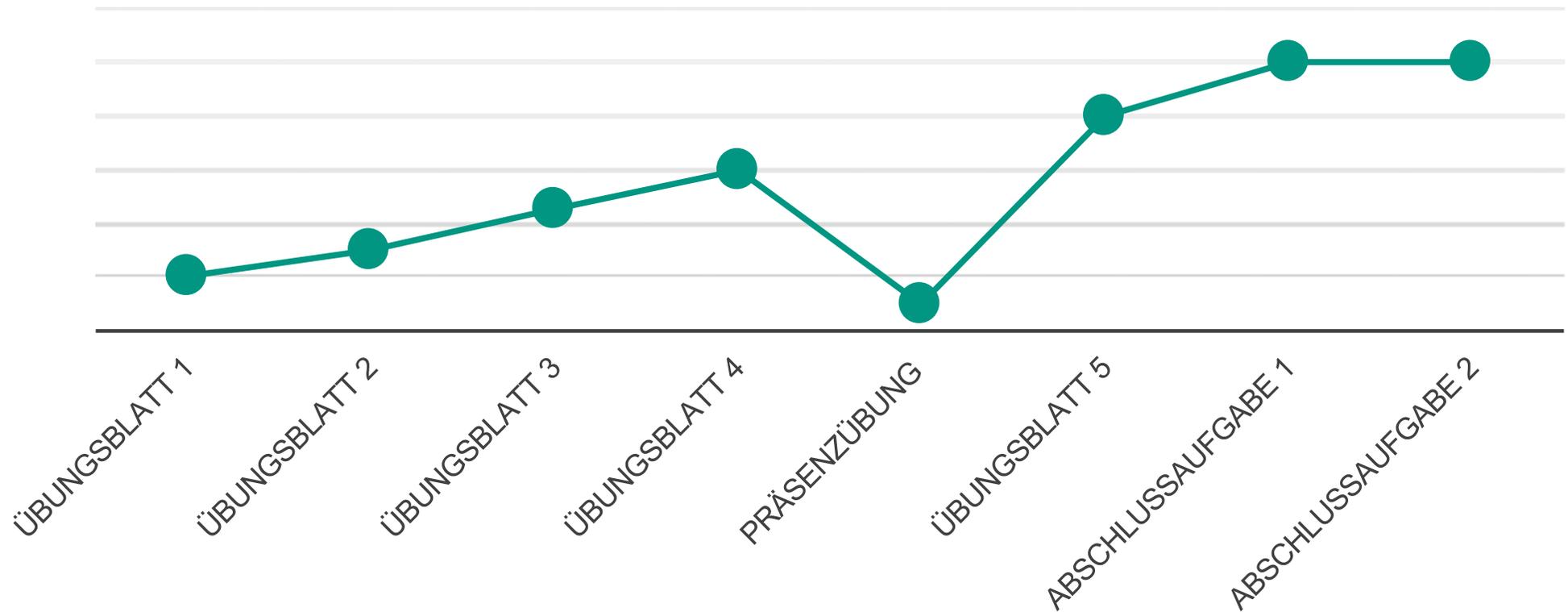
Programmieren Präsenzübung

- Prüft Wissen, welches bei der eigenständigen Bearbeitung der Übungsblätter erworben worden sein sollte
- Die Präsenzübung wird voraussichtlich als Präsenztermin stattfinden
 - Am 13. Januar 2023, zwischen 17:30 und 19:30 Uhr
 - Teilnahme setzt eine Anmeldung für den Übungsschein voraus
 - Informationen zur Zuteilung und Ablauf werden über Ilias angekündigt: s.kit.edu/ilias
- Einfache Fragen und Aufgaben zum Java-Grundlagenwissen
 - Keine Hilfsmittel, außer zuvor geprüften Wörterbücher
 - Aufgaben werden auf Deutsch gestellt

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

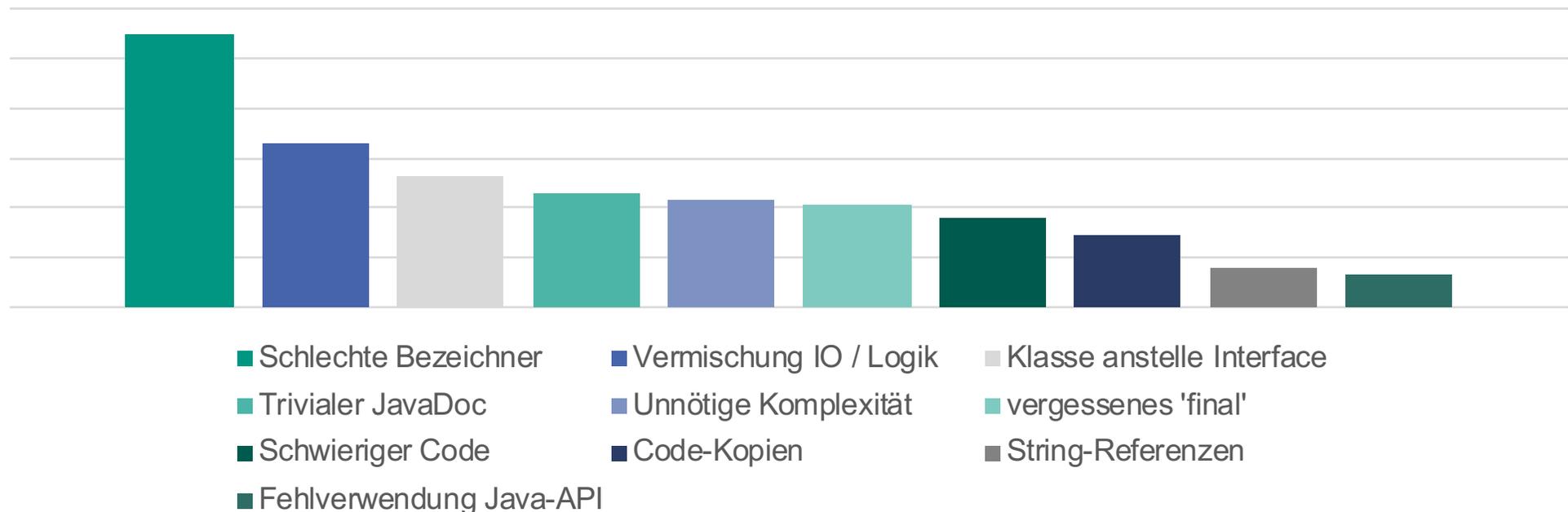
Organisatorische Tipps für die Übungsblätter

- 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
 - Von Anfang an „saubere“ Programmierung denken und sich daran gewöhnen
- 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 VPN und Artemis
 - Lösungen können beliebig häufig hochgeladen werden
 - Tutoren 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
- Overengineering 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
 - Final, wenn nur einmal zugewiesen
 - Nur Klassenvariablen sollten statisch sein
- Herangehensweise
 - Aufgabestellung genau lesen
 - Exakt die erwartete Ausgabe liefern
 - Bei Unklarheiten sofort nachfragen

Die häufigsten methodischen Fehler der letzten Abschlussaufgaben



Weitere Bewertungsrichtlinien im Wiki: s.kit.edu/wiki

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

- Websuche
- FAQ / Programmieren-Wiki prüfen
 - s.kit.edu/faq ■ s.kit.edu/wiki
- Diskussionsforum prüfen
 - Suchfunktion verwenden
- Im Diskussionsforum nachfragen
 - Aussagekräftigen Titel verwenden
- Eure Tutoren fragen
- Übungsleitung fragen
 - programmieren-vorlesung@cs.kit.edu

Weitere digitale Quellen

YouTube

- Prof. Dr.-Ing. Anne Koziolk – Wintersemester 2020/21: <https://www.youtube.com/watch?v=LmgKH0KPH3g&list=PLfk0Dfh13pBOKHJEvgdl1zLOLeOU2cng>
- Prof. Dr.-Ing. Anne Koziolk – Wintersemester 2019/20: <https://www.youtube.com/watch?v=pBexdqQDFcl&list=PLfk0Dfh13pBPtDhFFeFoa9wEclDKNQsX0>
- Prof. Dr. Ralf H. Reussner – Wintersemester 2018/19: <https://www.youtube.com/watch?v=aDuHbYxGviU&list=PLfk0Dfh13pBMZa4rJWYaR8HRQLJyHT5Vy>
- Prof. Dr.-Ing. Anne Koziolk – Wintersemester 2017/18: <https://www.youtube.com/watch?v=nTlwyd1OyK8&list=PLfk0Dfh13pBP3AkkIGjCHuI7ugNJI91Na>

Bücher (kostenlos über das KIT-Netzwerk erhältlich)

- Peter Pepper – Programmieren lernen: <https://www.springer.com/de/book/9783540723639>
- Dietmar Ratz et al. – Grundkurs Programmieren in Java: <https://www.hanser-elibrary.com/doi/book/10.3139/9783446453845>
- Elisabeth Jung – Java Übungsbuch: <http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&db=nlabk&AN=2347097>
- Christian Ullenboom – Java ist auch eine Insel: <https://openbook.rheinwerk-verlag.de/javainsel/>
- Kathy Sierra & Bert Bates – Java von Kopf bis Fuß: <https://oreilly.de/produkt/java-von-kopf-bis-fuss/>

Tutorials

- Oracle – The Java™ Tutorials: <https://docs.oracle.com/javase/tutorial/>
- Bradley Kjell – Introduction to Computer Science using Java: <https://chortle.ccsu.edu/java5/index.html>
- Zahlreiche zusätzliche Informationsquellen sind kostenlos im Internet zugänglich: <https://duckduckgo.com/?q=Java+Tutorial>

Weiter Veranstaltungen im WS 2022/23

- Mentoring-Stammtisch
 - Zeit sinnvoll nutzen
 - Freizeitgestaltung, Prokrastination und Studienmotivation
 - Herausfordernde Situationen und Umgang mit Stress

- 2.eezi Vorlesung: „Den Überblick behalten, Motivation, und Berufsbilder“
 - Am Dienstag, den 29.11.2022, um 17:30 Uhr, im Audimax

- eezi goIN' Info-Bite: ERASMUS+
 - Am Mittwoch, den 30.11.2022, um 17:30 Uhr, im Gebäude 50.34, Raum -102

- 3.eezi Vorlesung: „Prüfungsvorbereitung, Prokrastination, Prüfungsangst“
 - Am Dienstag, den 17.01.2023, um 17:30 Uhr im Audimax

eezi: Eine Einführung zum Informatikstudium am KIT



- Die Lehrveranstaltung „eezi“ gesprochen „easy“, ist für Studierende im ersten Semester Informatikstudium empfohlen
- 1 ECTS
- Nachzügler Anmeldung für eezi Tutorien Frist: 17.11.2022
 - Anmeldung Webseite:
 - Anmeldung QR-Code:

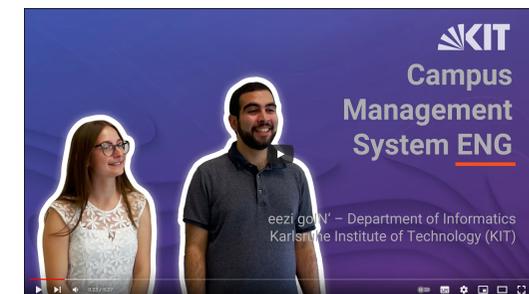


- E-Mail an eezi

■ Weiter Informationen: <https://www.informatik.kit.edu/10509.php>

eezi goIN' Info-Bits – Studieren leicht gemacht

- Kleine Videos (ca. 5 min) mit allgemeinen Tipps und Tricks zum Bewältigen des Studiums.
 - z.B.
 - „Die ersten sechs Wochen im Studium“
 - „Campus Management System for Students“
 - „Bachelor Module am KIT Fakultät für Informatik“



- Link zur Playlist:

<https://www.youtube.com/playlist?list=PL22ZNLSoHCRELZ8un710ObsaBfMvX4Yy6>

Fragen und Antworten



Also gladly in English

Lernpartnerschaftsbörse



(Bildquelle: <https://lernpartnerboerse.hoc.kit.edu/>)

Nicht nur für Programmieren,
sondern für alle Module des
Semesters