Hochschule Fulda
University of Applied Sciences

Programmierung 2

Formales schonmal vorab

Alexander Gepperth, April 2025

alexander.gepperth@cs.hs-fulda.de

Ankündigung: Pandora LAN

- Freitag/Samstag 2./3. Mai (**nächste Woche**)
- In Halle 8 (gegenüber) ab 18.00
- Organisiert von unserem FB, es gibt:
  - PC-Spiele
  - Konsolen
  - Brettspiele (gerne auch mitbringen!)
- Anmeldung: https://discord.gg/zuxaaRRgbE
- Studi-Ausweis mitbringen!

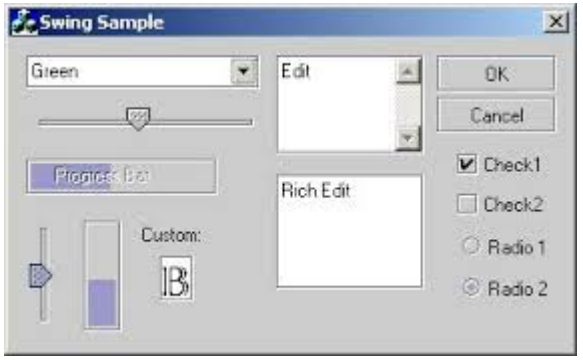
Inhalt der Vorlesung

- Fortführung der Vorlesung “Programmierung 1”
- Zielsetzung: Programmieren lernen (nicht: Java lernen)
- Behandlung generischer Konzepte der Programmierung

Inhalte der Vorlesung

- Was heißt “Programmieren lernen”?
  - **Java**
    - Programmiersprache Java
    - Objektorientierte Programmierung (Spezialfall: Java)
    - was passiert “hinter den Kulissen"?
  - **Generische Konzepte**:
    - Wichtige Datenstrukturen: Listen, Arrays, HashMaps
    - Vererbung, Objektorientierung, generische Typen, Exceptions, Iteration
  - **Programmierkompetenz**
    - Programme verstehen
    - Programme anhand klarer Vorgaben selbst erstellen
    - Mit und in großen Programmier-Projekten arbeiten
    - guten Code schreiben!
  - **Tools**
    - JavaDoc
    - IDEs
    - (git?)

Highlights



- Programmierung von grafischen Benutzeroberflächen

- Projekt: SpaceInvaders, Pacman, Breakout

- Vielleicht: Spiele-Programmierung in Unity3D

Organisation

- Sämtliche Kommunikation & Orga: Moodle
- VL wird aufgezeichnet (verfügbar auf Moodle), nur meine Slides und mein Bild zu sehen
- VL-Aufzeichnung ist Mitschnitt ohne Bearbeitung
- Aufgaben stehen sofort nach VL zur Verfügung
- Besprechung der Aufgaben in den Übungen der **folgenden Woche**!!

Übungen, Hausaufgaben, Testate

- Jeden Freitag: neue Online-Hausaufgaben
  - Coding-Aufgaben oder Quizzes
  - Abgabe per Moodle/CodeRunner
  - Abgabefrist: bis Freitag der **Folgewoche**
  - Bearbeitung: in Übung aber v.a. zu Hause
- Jeden Freitag: Übungsaufgaben v.a. zum Projekt
  - Übungsleiter vergeben Testate
  - Bearbeitung: in der Übung, oder vorher

Übungen

- Fragen zu Hausaufgaben stellen
- Jedes Team (2-3 Personen) bearbeitet Aufgaben, Mitglieder erklären Lösungen **einzeln** um Testate zu bekommen
- Sinn der Testate: Abprüfen von Fähigkeiten die in der Klausur schwer abprüfbar sind (mit IDEs arbeiten, Projekte verwalten, Tools benutzen, ...)

Portfolio-Prüfung

- Leistungen werden verteilt über das Semester erbracht
  - 5 Testate (**T**)
  - CodeRunner jede Woche (**C**)
  - Lernstandskontrolle am Semesterende (**L**)
- Was **muss** erfüllt sein um Credits zu bekommen?
  - zur LSK anmelden (in horstl)
  - aus allen 3 Leistungen >50% der Punkte erzielen
  - Gewichtung: $0.3T + 0.1C + 0.6L$

Lernstandskontrolle

- Findet im Semester statt, nicht in Prüfungsphase
- Wird aus dem Stoff der VL-Slides erstellt, Ähnlichkeit mit CodeRunner-Fragen möglich
- Dauer: 60min auf Papier, hpts. Theorie
- Keine Unterlagen
- Testat-Aufgaben nicht LSK-relevant

Literatur

- Viele Themen der Vorlesung werden in folgendem online frei (aber auch in der HSB) verfügbaren Buch behandelt:
  Christian Ullenboom: **Java ist auch eine Insel**
  http://openbook.rheinwerk-verlag.de/javainsel/
- Die Vorlesung wird ggf. auf die entsprechenden Buchkapitel hinweisen
- Buch soll
  - Dinge evtl. auf andere Art als als in der Vorlesung erklären
  - zur Vertiefung und zum Selbststudium dienen

Bei Fragen

- Bitte kontaktieren Sie **immer** zuerst Übungsleiter*innen wenn möglich
- Bei spezifischen Fragen an mich: bitte **keine** Mails schreiben
- Stattdessen: kommen Sie in meine **Online**-Sprechstunde (Details im E-Learning)

CUT: Q&A

Moodle: CodeRunner-Fragen

- Online-Hausaufgaben werden automatisch bewertet
- Tests mit numerischen und Multiple-Choice-Fragen
- CodeRunner-Modul im elearning-System, ähnlich CodingBat aus Prog1
- sieht so aus:

Moodle: CodeRunner-Fragen

Problem: In einem Array von Ganzzahlen soll ein bestimmter Wert mithilfe einfacher linearer Suche gefunden und zurückgegeben werden. Falls der Wert im Array nicht existiert, soll dies ebenfalls signalisiert werden.

Füllen Sie die Methode ***findeWertImArray (array, wert)*** so dass sie

- die Position (d.h. den Index) des ersten Auftretens des gesuchten Werts ***wert*** zurückgibt, falls dieser im Array existiert
- -1 zurückgibt falls dieser nicht im Array existiert

For example:

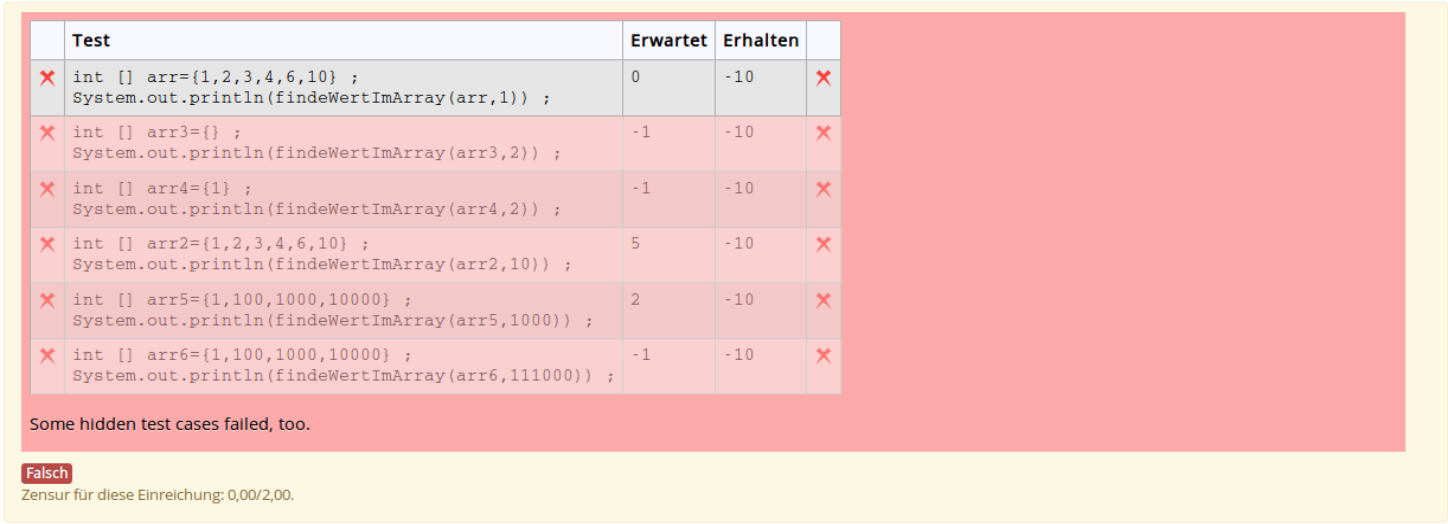
Test	Result
`int [] arr={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr,1)) ;`	0
`int [] arr3={} ;` `System.out.println(findeWertImArray(arr3,2)) ;`	-1
`int [] arr4={1} ;` `System.out.println(findeWertImArray(arr4,2)) ;`	-1

Antwort: (penalty regime: 0, 0, ... %)

```
1  /** Hier koennte man eintragen was die Funktion macht.. */
2  public int findeWertImArray (int [] array, int wert)
3  {
4      return -10 ;
5      /* for (int i=0; i<array.length; i++)
6        if (array[i] == wert) return i;
7      return (-1) ; // Loeschen und durch eigenen Code ersetzen! */
8  }
```

Prüfen

Moodle: CodeRunner-Fragen



	Test	Erwartet	Erhalten	
✗	`int [] arr={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr,1)) ;`	0	-10	✗
✗	`int [] arr3={} ;` `System.out.println(findeWertImArray(arr3,2)) ;`	-1	-10	✗
✗	`int [] arr4={1} ;` `System.out.println(findeWertImArray(arr4,2)) ;`	-1	-10	✗
✗	`int [] arr2={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr2,10)) ;`	5	-10	✗
✗	`int [] arr5={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr5,1000)) ;`	2	-10	✗
✗	`int [] arr6={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr6,111000)) ;`	-1	-10	✗

Some hidden test cases failed, too.

Falsch

Zensur für diese Einreichung: 0,00/2,00.

Code erfüllt keinen der hinterlegten Testfälle → keine Punkte

Moodle: CodeRunner-Fragen

	Test	Erwartet	Erhalten	
✓	`int [] arr={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr,1)) ;`	0	0	✓
✗	`int [] arr3={} ;` `System.out.println(findeWertImArray(arr3,2)) ;`	-1	0	✗
✗	`int [] arr4={1} ;` `System.out.println(findeWertImArray(arr4,2)) ;`	-1	0	✗
✗	`int [] arr2={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr2,10)) ;`	5	0	✗
✗	`int [] arr5={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr5,1000)) ;`	2	0	✗
✗	`int [] arr6={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr6,111000)) ;`	-1	0	✗

Some hidden test cases failed, too.

Teilweise richtig

Zensur für diese Einreichung: 0,33/2,00.

Code erfüllt manche aber nicht alle hinterlegten Testfälle → keine Punkte

Moodle: CodeRunner-Fragen

Code erfüllt alle hinterlegten Testfälle
→ Punkte!

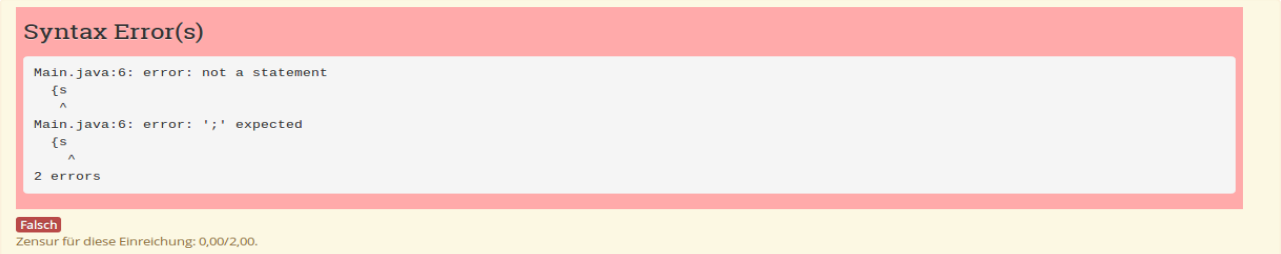
	Test	Erwartet	Erhalten	
✓	`int [] arr={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr,1)) ;`	0	0	✓
✓	`int [] arr3={} ;` `System.out.println(findeWertImArray(arr3,2)) ;`	-1	-1	✓
✓	`int [] arr4={1} ;` `System.out.println(findeWertImArray(arr4,2)) ;`	-1	-1	✓
✓	`int [] arr2={1,2,3,4,6,10} ;` `System.out.println(findeWertImArray(arr2,10)) ;`	5	5	✓
✓	`int [] arr5={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr5,1000)) ;`	2	2	✓
✓	`int [] arr6={1,100,1000,10000} ;` `System.out.println(findeWertImArray(arr6,111000)) ;`	-1	-1	✓

Passed all tests! ✓

Richtig
Zensur für diese Einreichung: 2,00/2,00.

Moodle: CodeRunner-Fragen

- Compiler-Fehler werden im Moodle-System angezeigt:
- im Prinzip kann eine Aufgabe komplett im Browser bearbeitet werden!



- trotzdem sinnvoll die Aufgaben in einem Editor zu bearbeiten und regelmäßig zu speichern
  - Sicherungskopie falls Browser abstürzt
  - Dokumentation der Abgabe