

Handlungsorientierte
Programmierprüfungen
mit Moodle, CodeRunner, Jobe,
Nailgun, GNU/Linux

Stefan Müller, KS Zofingen

Ablauf

- 1) Prüfungen im obligatorischen Fach Informatik (15' Referat)
- 2) Beispielprüfungen (20' Workshop)
- 3) Architektur (15' Referat)
- 4) Fragerunde (10')

Fragen, Bemerkungen bitte im Chat formulieren!

Allgemein

- Bei Problemen mit Audio bitte Chat benutzen!
- Nach Abschluss geht es um 11:40 weiter mit den Lightning Talks.
- Einwahl für alle Programmpunkte:
openeducationday.ch/programm-2020-virtuell

Wer?

- A) Lehrpersonen Informatik mit Unterrichtserfahrung
- B) Lehrpersonen Informatik ohne Unterrichtserfahrung
- C) andere Angehörige von Schulen
- D) Andere

Prüfungen im obligatorischen Fach Informatik

- 1) Ausgangslage
- 2) Prüfungsformen
- 3) Moodle & Co.

Ausgangslage

- EDK (2017): Neues obligatorisches Fach Informatik spätestens ab Schuljahr 2022/23
 - Rahmenlehrplan
- Kt. AG: Einführung als kantonales Fach 2016
 - Kantonaler Lehrplan

Rahmenlehrplan (Auszug)

- besonders geeignet für handlungsorientierte oder projektartige Unterrichtsgestaltung
- Grundkonzepte einer Programmiersprache verstehen (Programmieren)
- Organisationsformen grösserer Datenmengen begreifen (Datenbankabfragen)

Programmiersprachen im Oinf? (geplant oder eingesetzt)

- A) Python
- B) Java
- C) etwas Anderes
- D) noch nicht bestimmt

Python: Tigerjython?

- A) Verwendung im Oinf
- B) schon gehört
- C) unbekannt

Problemstellung

- Welche Prüfungsformen passen zu handlungsorientierten Unterricht?
- Wie lassen sich derartige Prüfungen effizient organisieren und durchführen?

Prüfungsformen Programmieren, Datenbankabfragen (Oinf)
















- Papierprüfung: Liefert Alpha-Qualität
 - fokussiert auf Problemanalyse
 - anstrengend zu korrigieren
- Projekte: Für Anfänger (zu) anspruchsvoll
 - genügend individuelle Projekte
- Handlungsorientierte Prüfungen
 - entspricht Unterricht
 - Alleinstellungsmerkmal der Informatik
 - technische Umsetzung

Prüfungen Oinf: Erfahrungen


- 2016: 125x Oinf1
- 2017, 2018: 100x Oinf1 + 125x Oinf2
- Papierprüfungen
- alle Klassen quasi gleichzeitig
- Äquivalenz der Prüfungen bei gleichzeitiger Unabhängigkeit steigert Aufwand ins Absurde
- zusätzlich zur Aufbereitung des Stoffs
- 2019: Moodle to the rescue!

Moodle: Fragetypen (Standard)

QUESTIONS

-  Multiple choice
-  True/False
-  Matching
-  Short answer
-  Numerical
-  Essay
-  Calculated
-  Calculated multichoice
-  Calculated simple
-  Drag and drop into text
-  Drag and drop markers
-  Drag and drop onto image
-  Embedded answers (Cloze)
-  Random short-answer matching
-  Select missing words

OTHER

-  Description

Fragetypen Plugins

- moodle.org
Plugins > Activities > Quiz > Question types
- CodeRunner
 - programming questions
 - student answer is code
 - graded by running

CodeRunner: Bsp. 1

Write a Python3 function `sqr(n)` that returns the square of its numeric parameter `n`.

For example:

Test	Result
<code>print(sqr(-3))</code>	9
<code>print(sqr(11))</code>	121

Answer:

```
1 def sqr(n):  
2     return n * n
```

Check

CodeRunner: Bsp. 1 (Auswertung)

Test	Expected	Got	
<code>print (sqr (-3))</code>	9	9	✓
<code>print (sqr (11))</code>	121	121	✓
<code>print (sqr (-4))</code>	16	16	✓
<code>print (sqr (0))</code>	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

CodeRunner: Bsp. 2

Question 2

Correct

Mark 0.90 out of
1.00

Flag question

Edit question

Write a Python function *squares(n)* that returns a list of the squares of all integers from 1 to n inclusive. Returns the empty list if $n < 1$.

For example:

Test	Result
<code>print(squares(5))</code>	<code>[1, 4, 9, 16, 25]</code>

Answer:

```
1 def squares(n):  
2     return [i * i for i in range(1, n + 1)]
```

Check

CodeRunner: Bsp. 2 (Auswertung)

	Test	Expected	Got	
✗	<code>print(squares(5))</code>	<code>[1, 4, 9, 16, 25]</code>	<code>[0, 1, 4, 9, 16]</code>	✗
✗	<code>print(squares(1))</code>	<code>[1]</code>	<code>[0]</code>	✗
✗	<code>print(squares(7))</code>	<code>[1, 4, 9, 16, 25, 36, 49]</code>	<code>[0, 1, 4, 9, 16, 25, 36]</code>	✗
✓	<code>print(squares(0))</code>	<code>[]</code>	<code>[]</code>	✓
✓	<code>print(squares(-10))</code>	<code>[]</code>	<code>[]</code>	✓
✗	<code>print(squares(2))</code>	<code>[1, 4]</code>	<code>[0, 1]</code>	✗

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Incorrect

Marks for this submission: 0.00/1.00. This submission attracted a penalty of 0.10.

2) Praxis

- <https://moodle.munterwegs.net>
- User=Covid-# [#=1..18]
- Passwort=#Corona!
- Kurs: Open Education Day 2020
- Zwei Übungsprüfungen
 - Ich führe euch durch die Aufgaben.

Turtle-Programmierung mit Tigerjython

- A1) Erläuterungen
 - hier: kein zusätzliches Tigerjython
- A2) Fehlersuche (Multiple-Choice)
- A3) statische Figur (Pfeil)
- A4) variable Figur (Pfeil)
- A5) komplexe Figur unter Verwendung einer vorgegebenen Grundfigur (Flagge)

SQL mit Sqlite3

- DB-Modul von inf-schule.de
- A1) SELECT FROM WHERE ORDER BY
- A2) Zwei Tabellen
- A3) DISTINCT
- A4) Drei Tabellen
- A5) Aggregation
- A6) Selbstreferenz

Bemerkungen zu SQL mit Sqlite3

- Resultate vorgegeben hilft, Verständnisprobleme zu vermeiden
- Immer Sortierung verlangen!
- Sqlite3 \neq MySQL
- Sqlite3: einfaches DB-Format
 - DB muss mit der Aufgabe definiert werden
- Arbeitsplattform muss identisch sein

Bemerkungen zu Turtle- Programmierung mit Tigerjython

- Was ist ein korrektes Programm?
- Wie beurteilt man die Korrektheit von graphischem Output programmatisch?
- Wie erzeugt man graphischen Output von einem GUI-Programm ohne Graphikkarte?
- Tigerjython ist Jython, d.h. Python/Java
 - bei jeder Ausführung wird eine neue JVM gestartet: mind. 5 Sekunden Overhead
 - Server überlastet

→ Eigentlich gar nicht möglich!

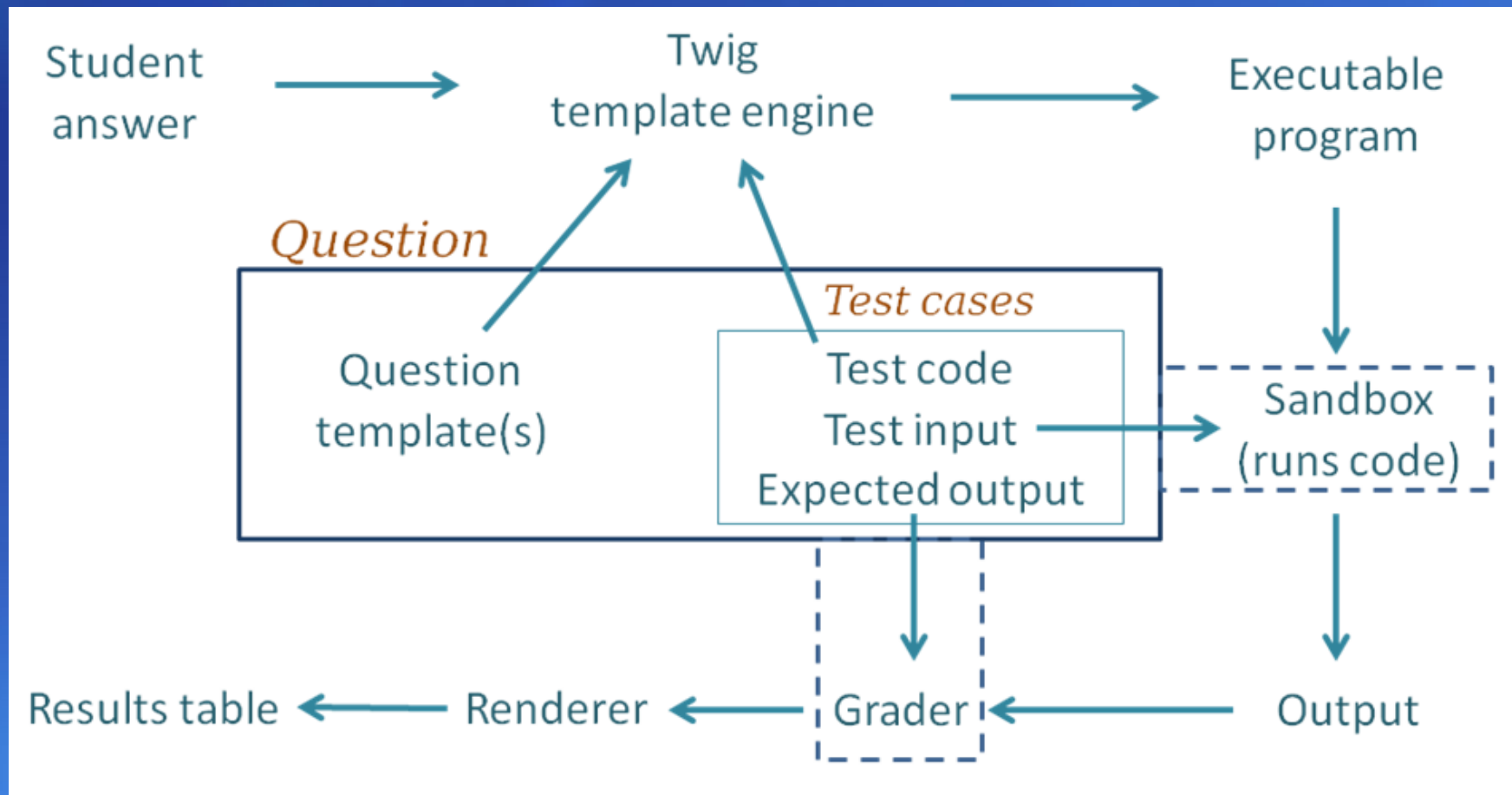
3) Architektur

- 1) CodeRunner & Jobe (Richard Lobe)
- 2) Fragen-Templates (sm)
- 3) Prüfungsorganisation

Moodle & CodeRunner

- Moodle = Rahmen
 - Benutzer, Kurse, Quiz, Bewertungen usw.
- CodeRunner = Fragentyp
 - Pre-Processing der Antworten (Einbettung in ausführbaren Code)
 - Weiterleitung an Sandbox zur Ausführung
 - Rückgabe der Bewertung an Moodle

coderunner.org.nz



Sandbox

- Jobe = "job execution server"
 - getrennt von Moodle-Instanz
 - stark eingeschränkte Nutzerrechte
- führt Code aus
 - in jeder installierbaren Programmiersprache:
Python 2/3, C(++), Java, Sqlite, ...
- wertet Resultat aus
 - simpler Vergleich
 - programmatische Auswertung

Installation

- Moodle (LXC) mit CodeRunner (Plugin)
 - Fragen-Template*
- Jobe (LXC) mit ...
 - benötigten Sprachen (Sqlite3, Jython etc.)
 - Nailgun-Server (persistente JVM, welche Jobs entgegennimmt) mit persistenten Java-Bibliotheken (Turtle etc.)
 - mehrfache Instanzen für Parallelisierung (*)
 - "guardian angel"-Prozess
 - ImageMagick, Xvfb

Fragen-Template Sqlite3

- **autochecking: Verbotene Ausdrücke in SQL**
 - LIKE (falls nicht notwendig)
 - Suchresultate
 - optional: Liste von Begriffen
- **limit: Anzahl Treffer**

Fragen-Template TigerJythonTurtle

- `dryrun`: Musterlösung produzieren (Bilder)
 - `imageBorder`: Bilder zuschneiden
- `penWidth`: Um Pixelfehler zu "übermalen"
 - `pHashPrecision = 0.01`
- `posHeading`: Turtle-Position überprüfen
 - `xyHprecision = [1, 1, 1]`
- `forbiddenFcts`: Verbotene Begriffe im Code
- `preload`: Bibliothek laden
- `Timeouts`: Nailgun (Jython), ImageMagick

Prüfungsorganisation

- Moodle
 - Benutzer: Login
 - Prüfung: Zeitpunkt, Dauer
- Zugriff
 - Prüfungsstick-Server Glados
(Roman Gruber, Ronny Standke)

4) Fragerunde