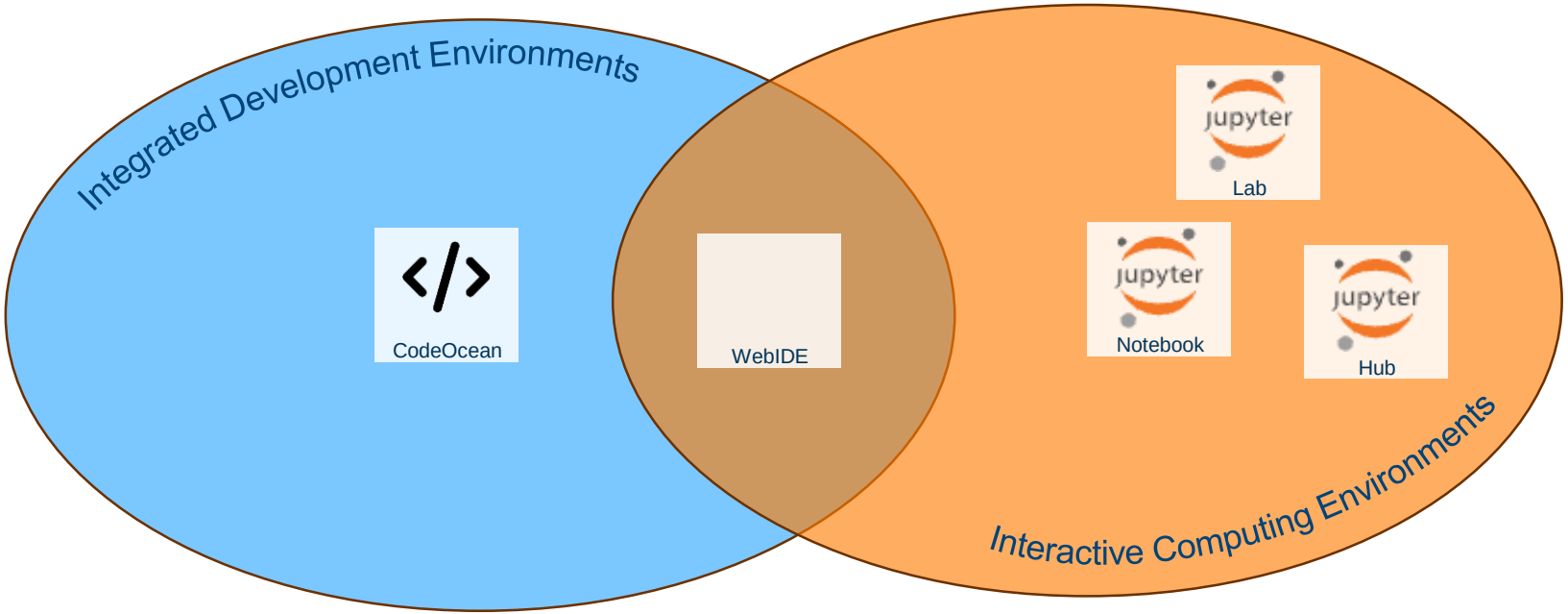
Tag der Lehre 2024

Technologie-Unterstützung in der Programmierausbildung

(Lern-)**Werkzeuge**

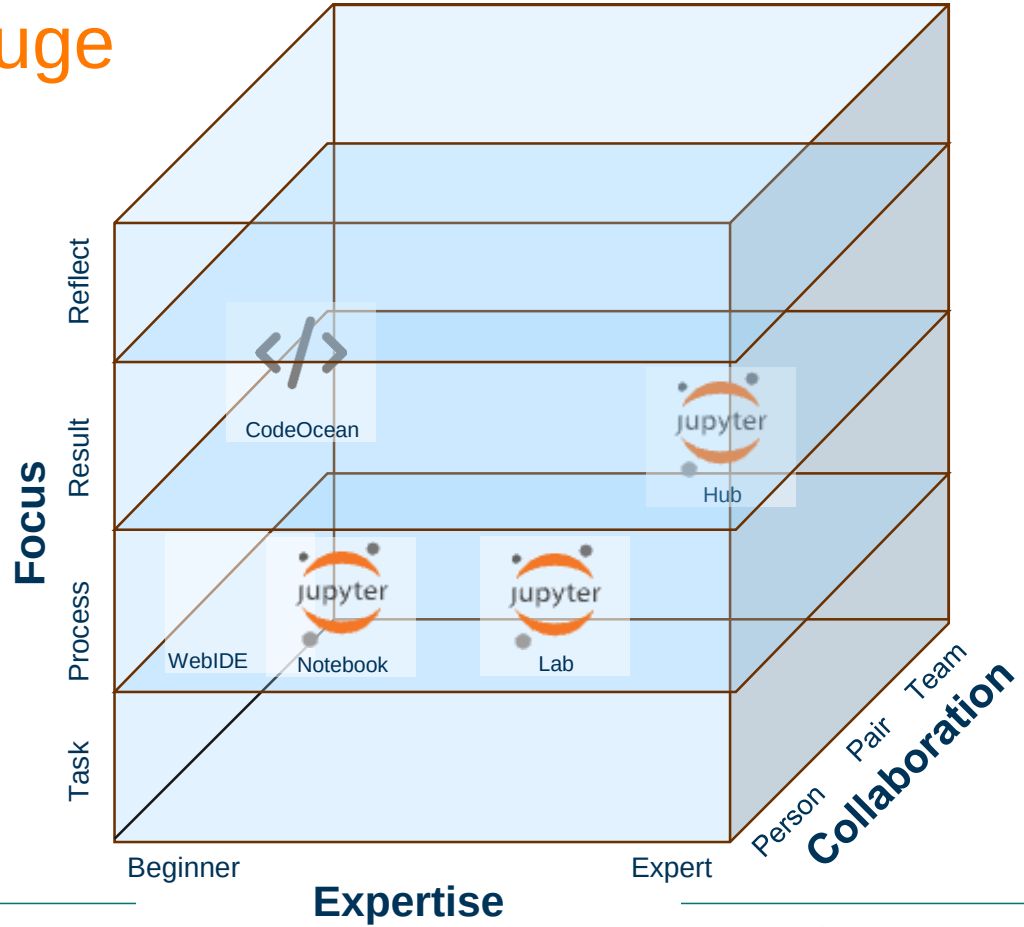


LERN-Werkzeuge

Usability?

Adaptability?

Maintainability?



Stiftung Innovation in der Hochschullehre

examING TU Ilmenau

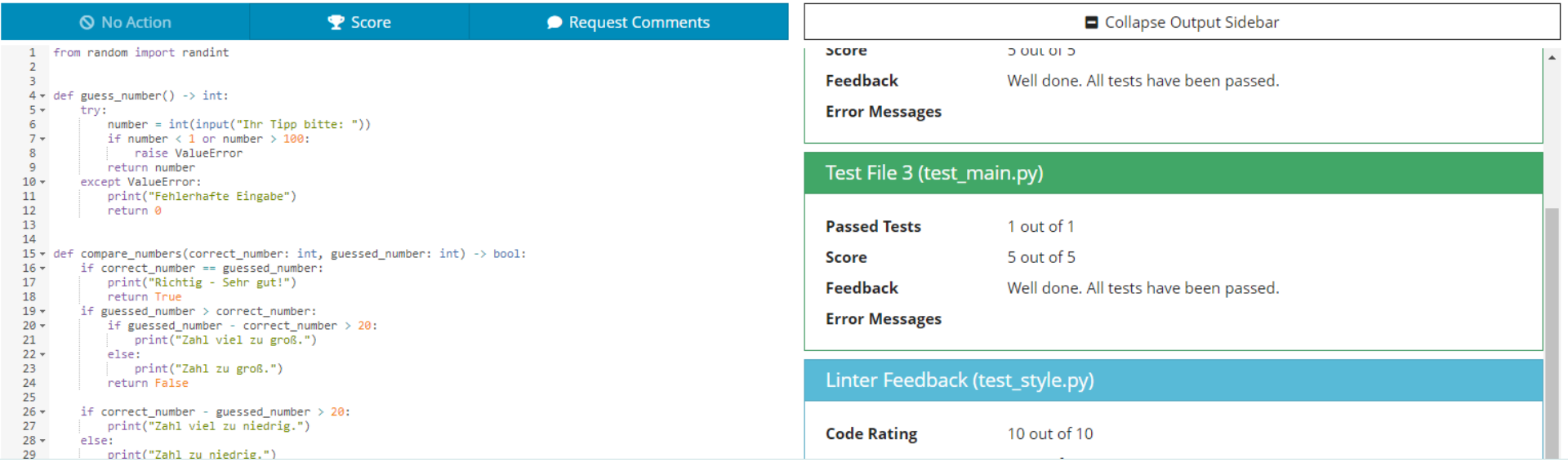
The SPIRIT of science

TECHNISCHE UNIVERSITÄT ILMENAU

Tag der Lehre 2024

Aufgabenorientierter Einstieg in die Programmierausbildung mit CodeOcean

examING-Teilprojekt: **Auto**grader in der **P**rogrammierausbildung für **ING**enieure (AutoPING)



Stiftung Innovation in der Hochschullehre

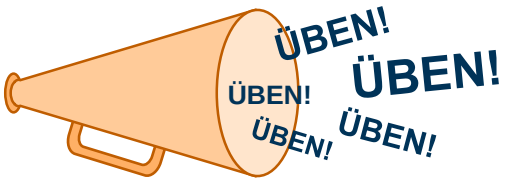
examING TU Ilmenau

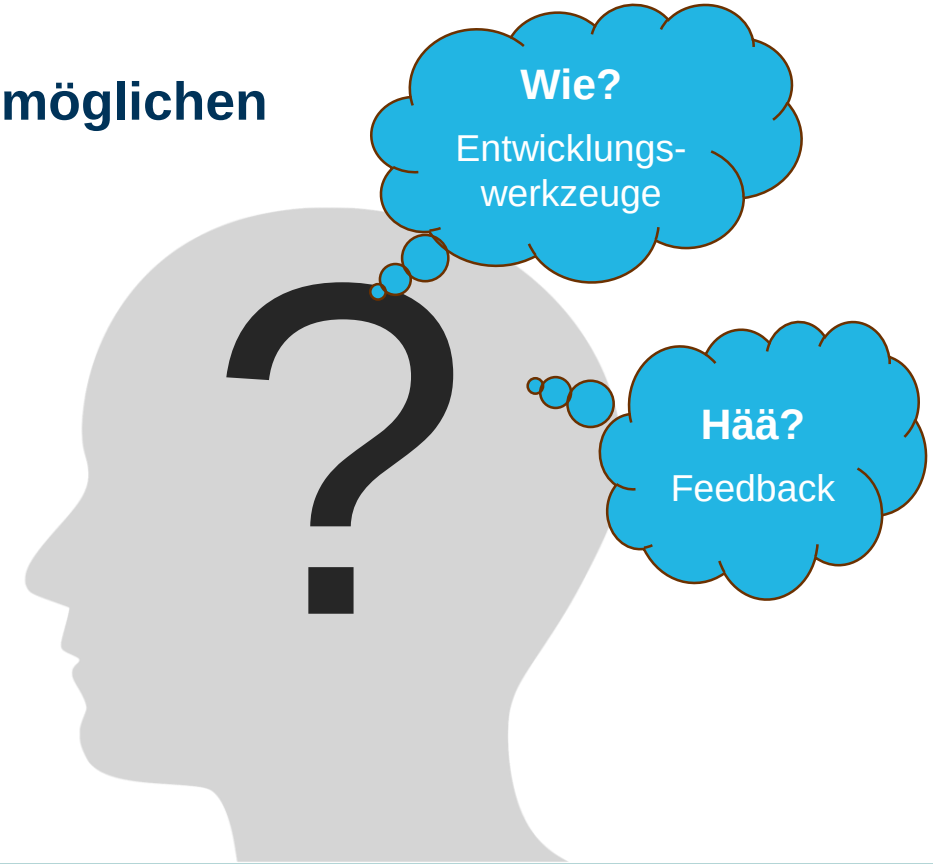
The SPIRIT of science

TECHNISCHE UNIVERSITÄT ILMENAU

Problem

Kompetenzorientiertes Lernen ermöglichen





Lösung

CodeOcean Autograding-System

1. im Produktivbetrieb bewährt
2. webbasierte Lösung mit LTI-Schnittstelle für Moodle-Integration
3. Protokollierungs-, Statistik-, Kollaborationsfunktionen
4. verspricht leichte Adaptierbarkeit
5. aktives Entwickler-Team

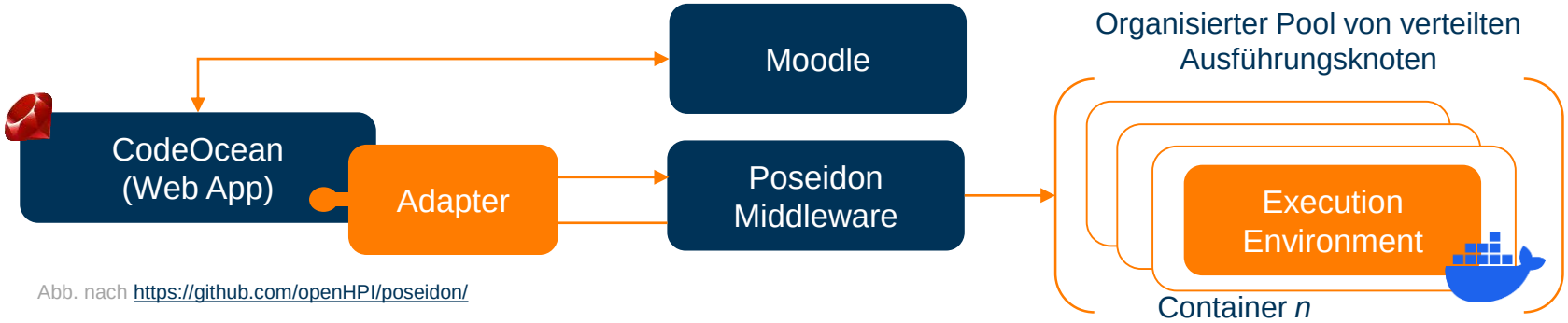
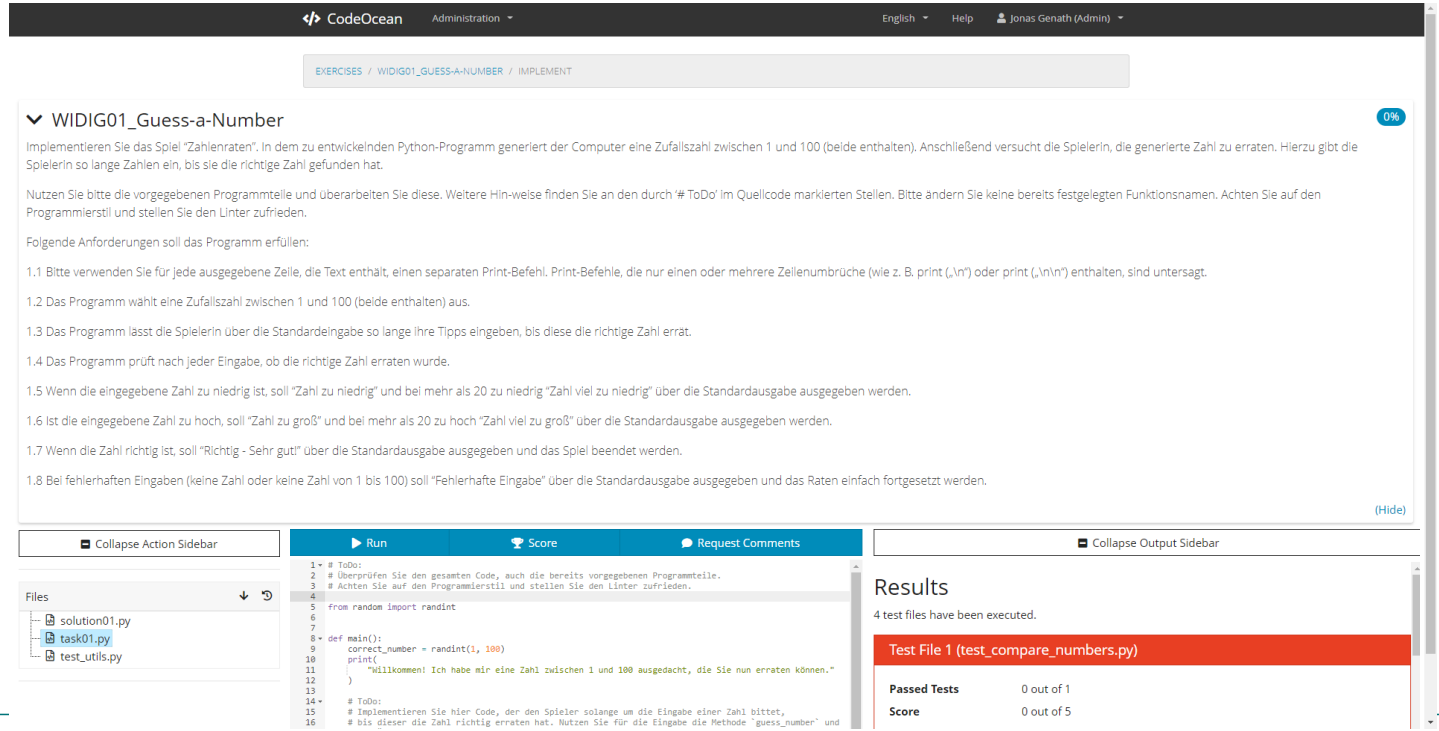


Abb. nach https://github.com/openHPI/poseidon/

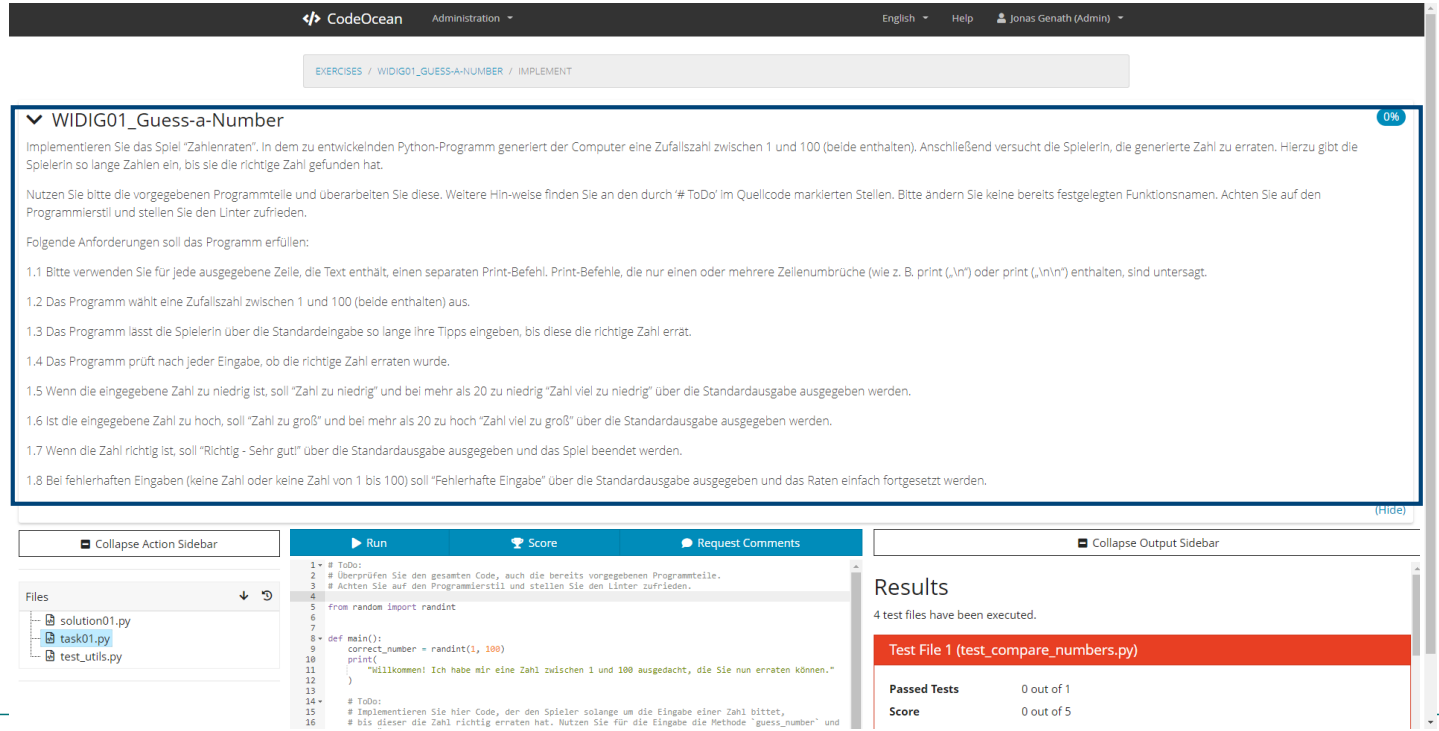
CodeOcean für Lernende

Aufgaben-Ansicht



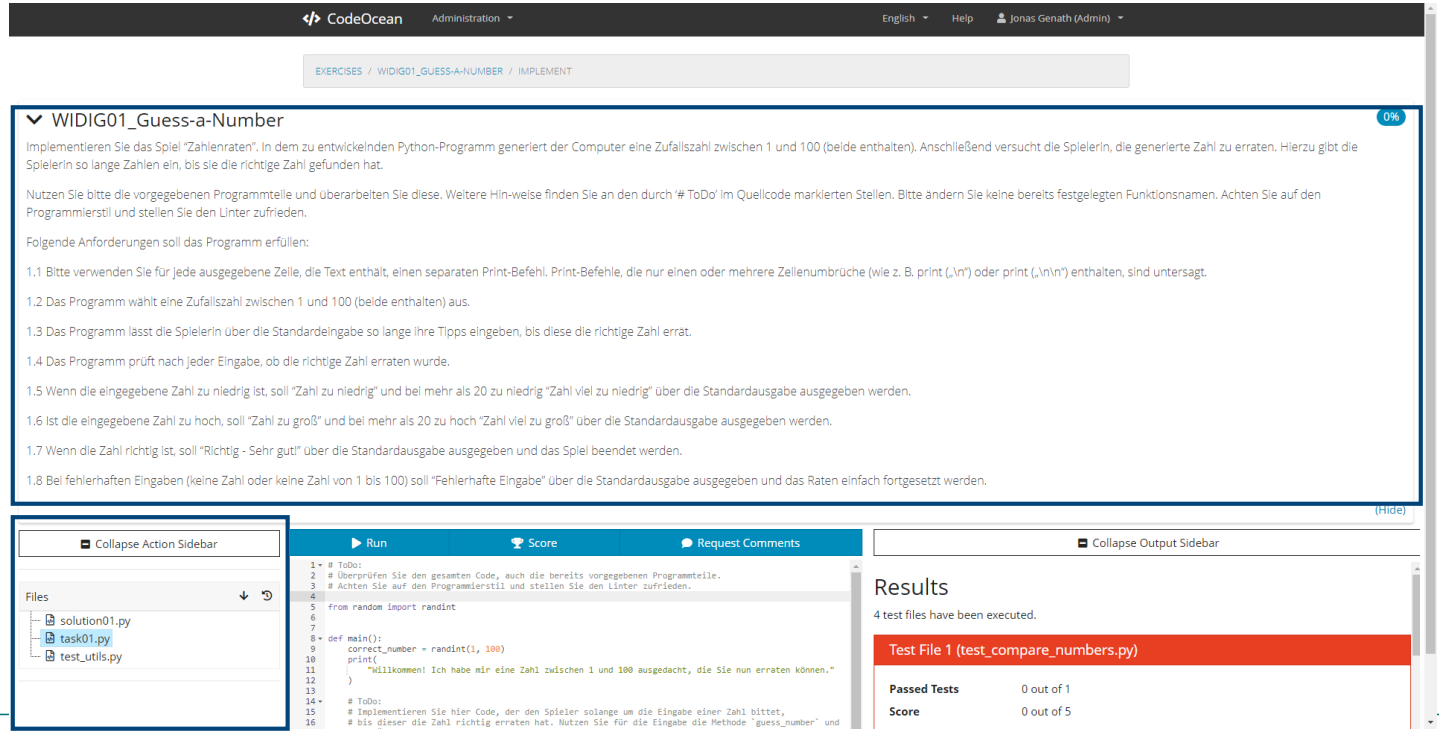
CodeOcean für Lernende

Aufgaben-Ansicht



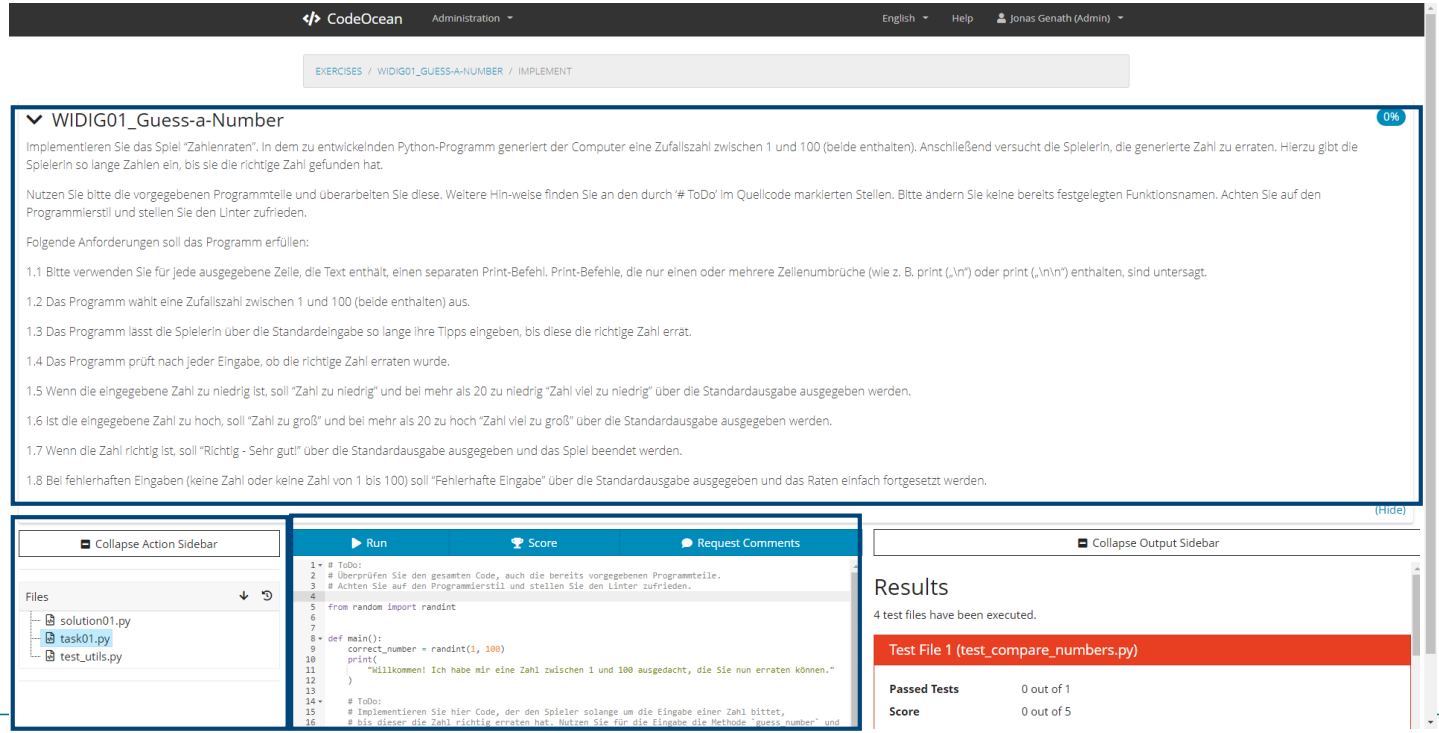
CodeOcean für Lernende

Aufgaben-Ansicht



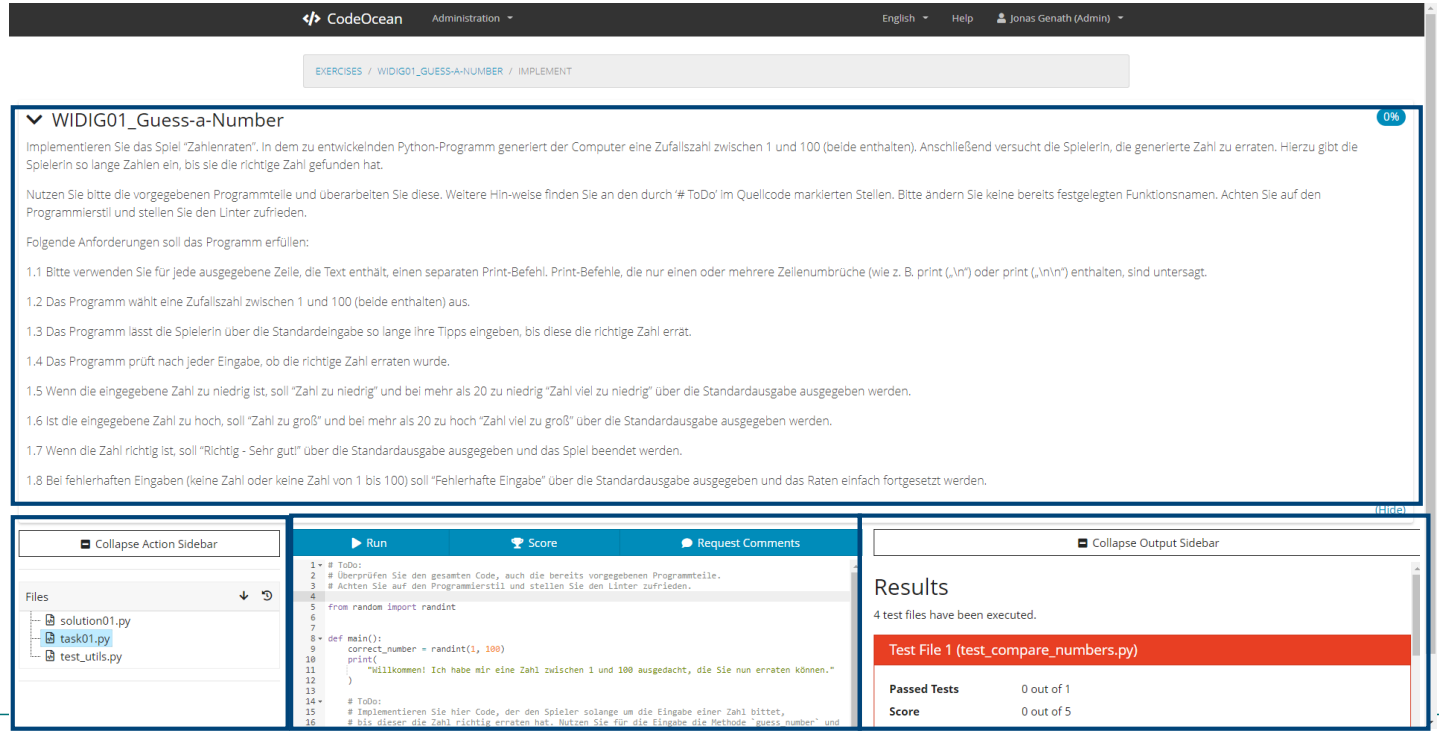
CodeOcean für Lernende

Aufgaben-Ansicht



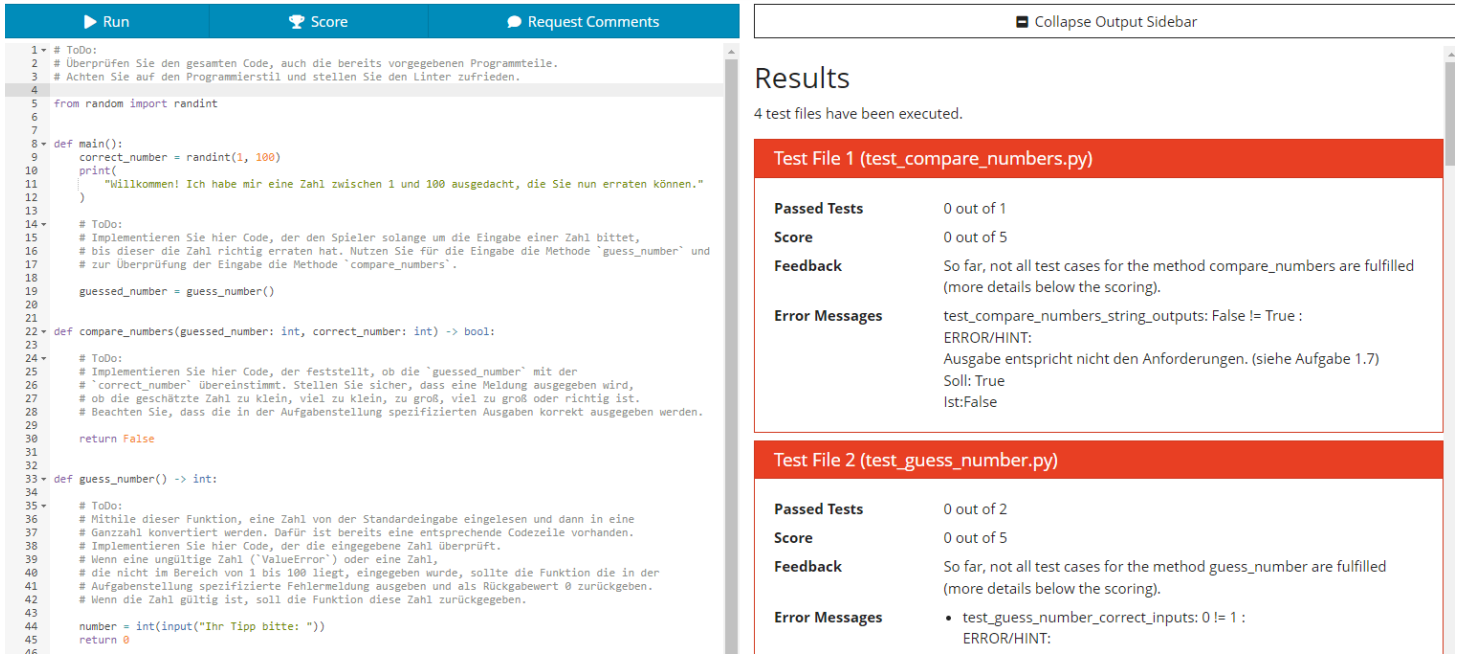
CodeOcean für Lernende

Aufgaben-Ansicht



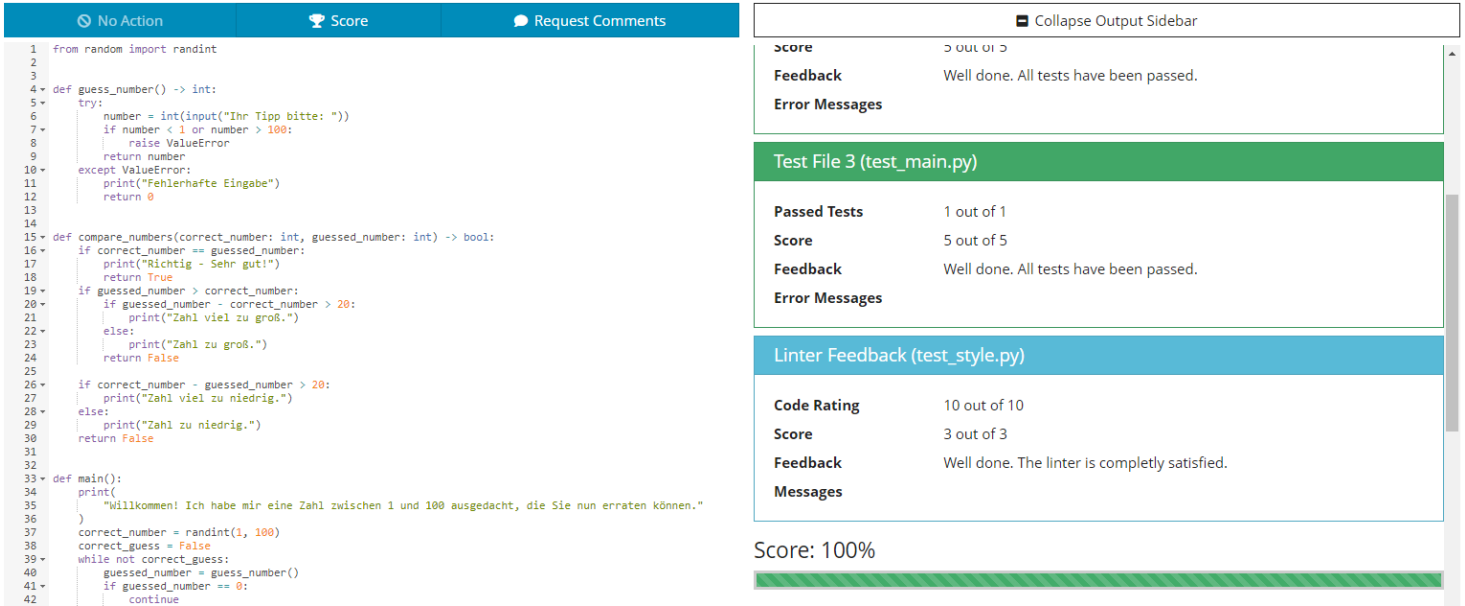
CodeOcean für Lernende

Feedback für nicht bestandene Tests



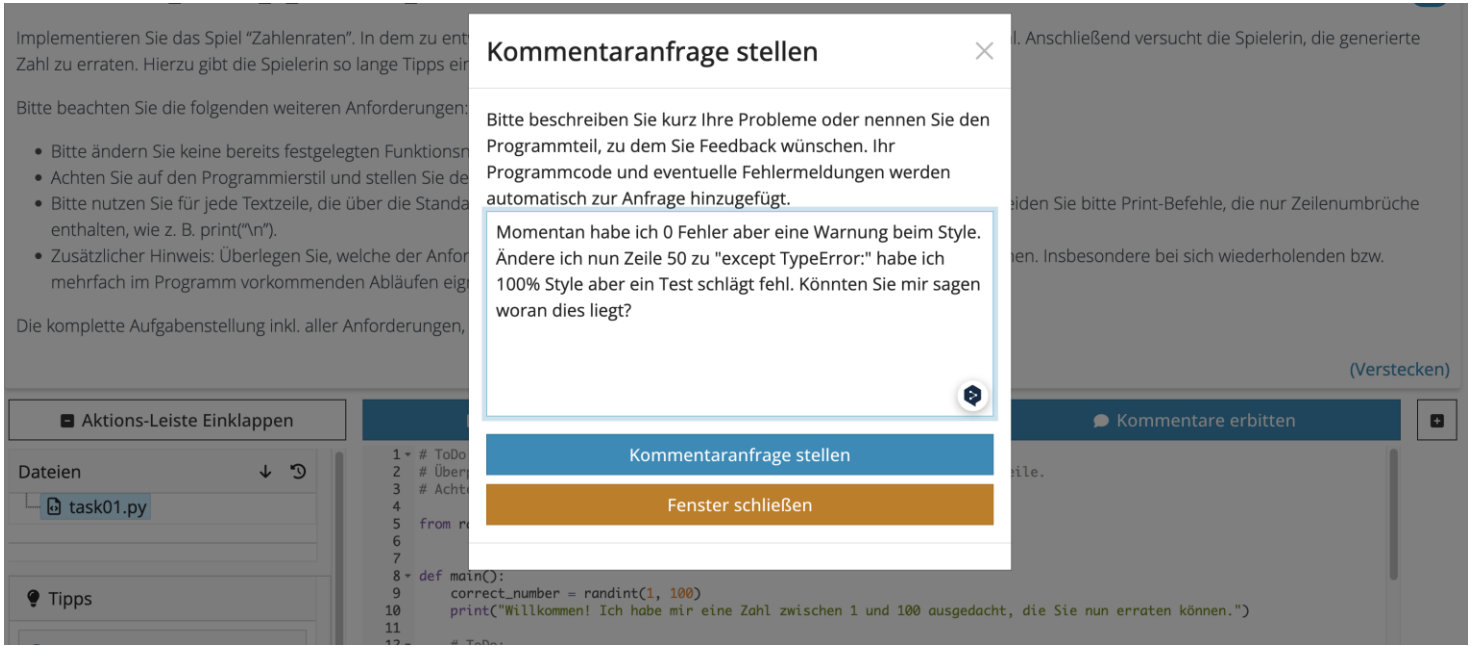
CodeOcean für Lernende

Feedback für bestandene Tests



CodeOcean für Lernende

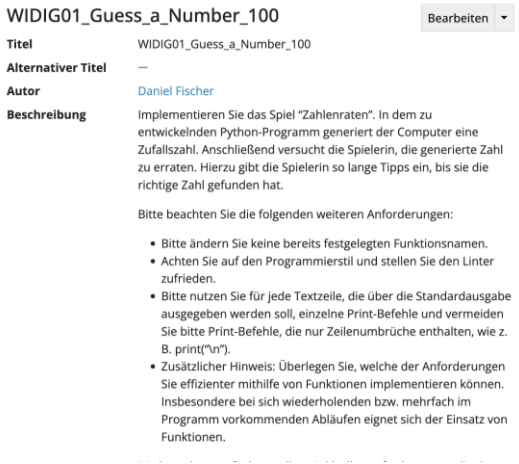
Kommentar-Anfragen in die Lerngemeinschaft



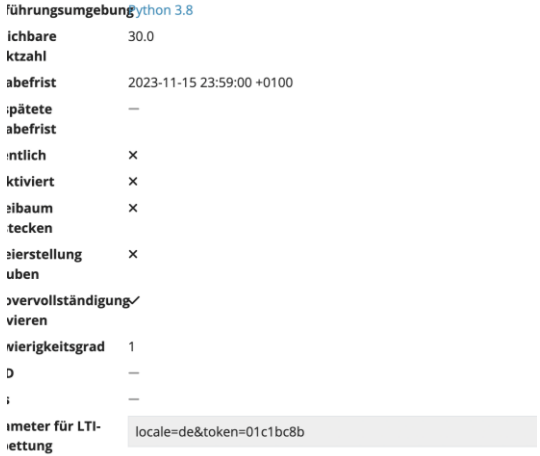
CodeOcean für Lehrende

Aufgaben erstellen

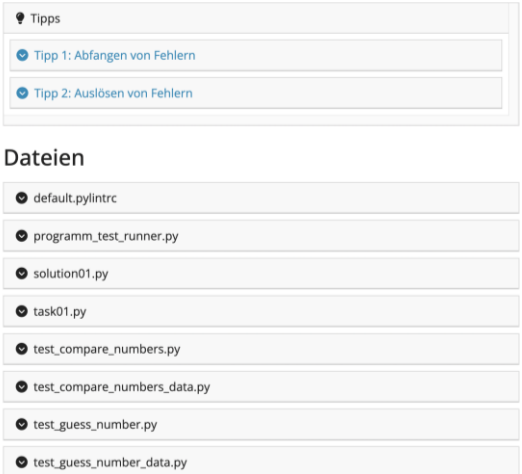
1. Aufgabenbeschreibung



2. Parameter

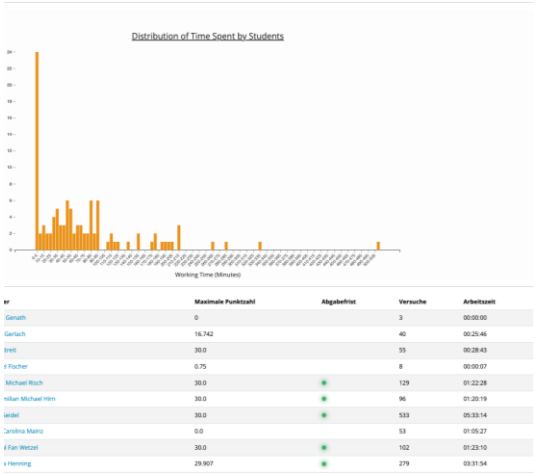


3. Dateien (Hauptdatei, Testdatei, …)

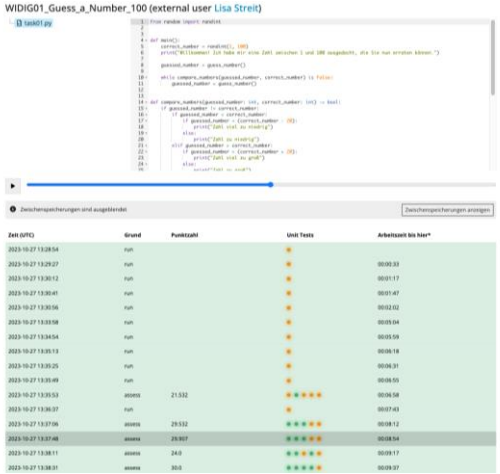


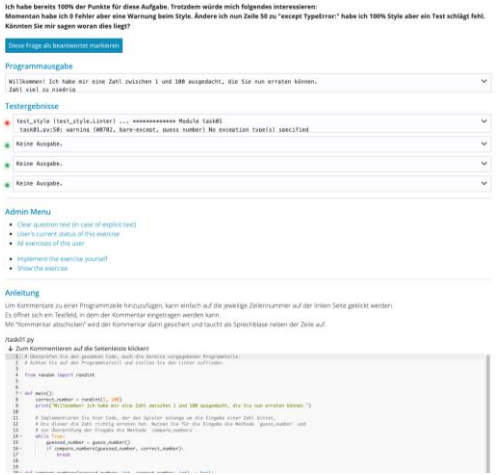
CodeOcean für Lehrende

Statistiken, Protokollierung, Kommentare



er	Maximale Punktzahl	Abgabefrist	Versuche	Arbeitszeit
Genath	0		3	00:00:00
Gerlach	16.742		40	00:25:46
treit	30.0		55	00:28:43
t Fischer	0.75		8	00:00:07
Michael Risch	30.0		129	01:22:28
nilian Michael Hirn	30.0		96	01:20:19
leidel	30.0		533	05:33:14
Carolina Mainz	0.0		53	01:05:27
t Fan Wetzel	30.0		102	01:23:10
Henning	29.907		279	03:31:54





Stiftung Innovation in der Hochschullehre

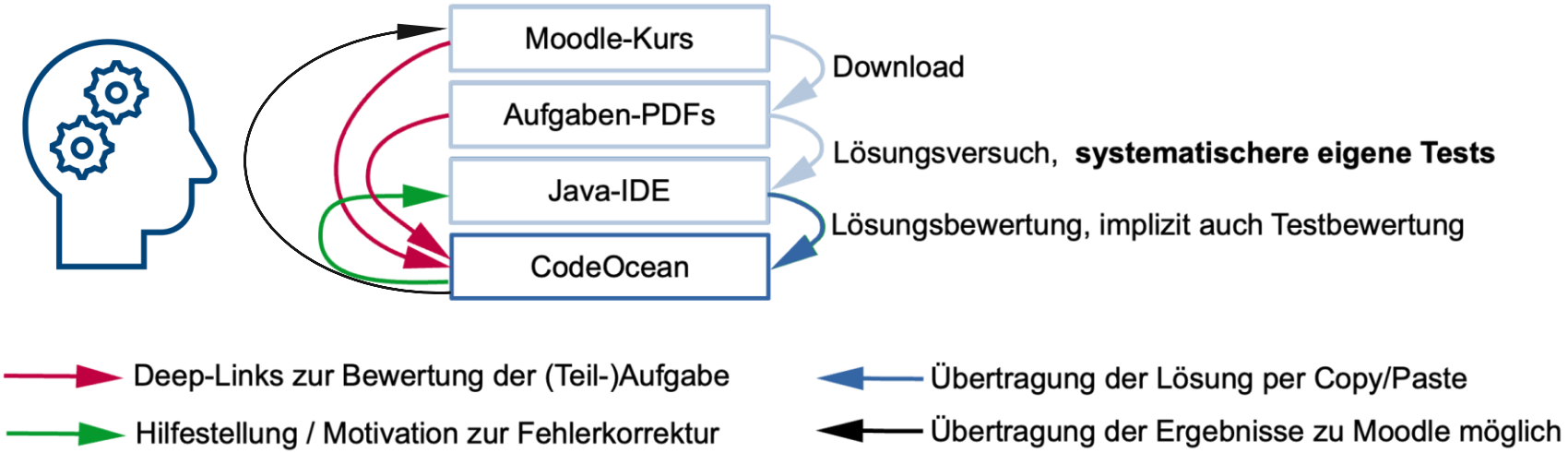
examING TU Ilmenau

The SPIRIT of science

TECHNISCHE UNIVERSITÄT ILMENAU

Didaktische Integration

Beispiel: Erste Programmierfähigkeiten in Java



Didaktische Integration

Welche Schwerpunktprobleme gibt es?

- neben funktionalen Tests auch Tests auf formale Anforderungen der Aufgabenstellung (Rückgabetypen, …)
- Fehlerbeschreibungen müssen an Anfängerniveau angepasst sein
  → Hilfe zur Fehlersuche oder Behebung
- Testwerkzeuge (JUnit, PyUnit) sind häufig auf Softwareentwickler zugeschnitten
  → Fehlermeldungen für Anfänger häufig nicht verständlich
  → Studierende verlieren ihre Motivation
- Voneinander unabhängige Tests dürfen sich nicht beeinflussen
  → wichtig für Erfolgserlebnisse / Motivation

Didaktische Integration

Wie sehen die verfolgten Lösungsansätze aus?

1. Hilfestellung für Fehlerlokalisierung (Anzeige von Parametern/Werten, die zu Fehlern führten) → Nachvollziehbarkeit sicherstellen
2. Entwicklung eigener Testframeworks auf Basis bestehender Testwerkzeuge
3. was die Aufgabenstellung nicht explizit fordert, wird auch nicht getestet

Erfolgreiche Tests	0 von 2
Punktzahl	0 von 5
Feedback	So far, not all test cases for the open_cars_with_most_reservations function are fulfilled (more details below the scoring).
Fehlermeldungen	• test_1_open_cars_with_most_reservations: 10 != 0 : Aufgabe 13: 10 Fehler für den User-Input 0 aufgetreten. • Die 3. Ausgabe ist fehlerhaft. Soll: 96 Tage - Opel Corsa. Ist: 86 Tage - Skoda Octavia.

Didaktische Integration

Welche Erfahrungen konnten gesammelt werden?

1. Befragungen und Beobachtung ergaben:
   - Einsatz von CodeOcean von den Studierenden insgesamt als positiv bewertet
   - Studierende lobten vor allem die Möglichkeit, jederzeit automatisches Feedback zu ihren Lösungsversuchen abrufen zu können (auch bei den Abschlusstests)
2. aber immer noch Kritikpunkte:
   - teilweise ungenaues, verwirrendes oder nicht gut nachvollziehbares Feedback
   - unklare und missverständliche Aufgabenstellungen
3. Workflow muss für die Studierenden einfach nachvollziehbar und durchführbar sein
4. Hauptarbeitspunkt: Verständlichkeit des Feedbacks und der Aufgabenstellungen

Technische Integration

Adaptierbarkeit von CodeOcean

1. Angepasste Execution Environments für weitere Use Cases
2. Adapter für weitere Programmiersprachen / Testumgebungen

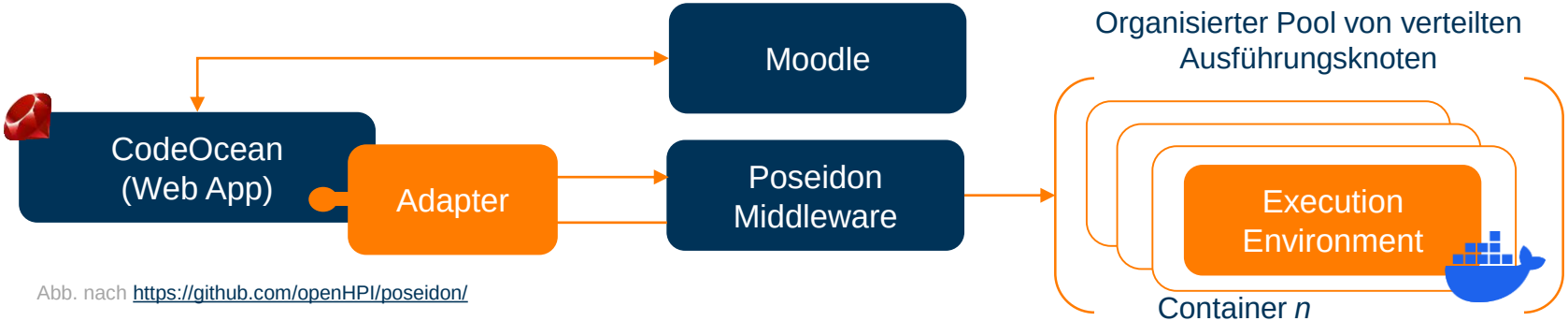


Abb. nach https://github.com/openHPI/poseidon/

Beispiel: Angepasstes Execution Environment

Wie kann kompetenzorientiert mit E2E-Tests geprüft werden?

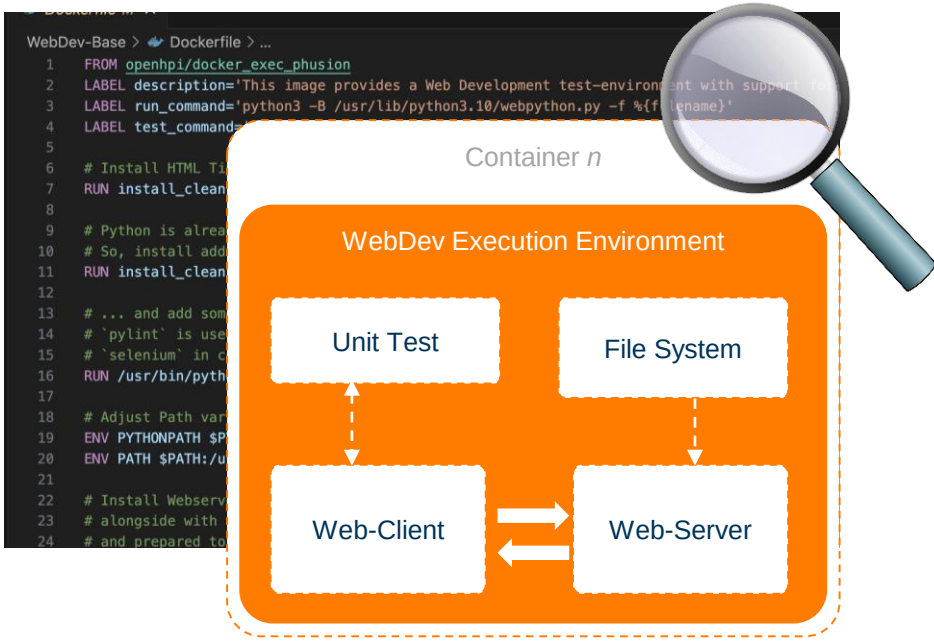
Szenario: Aufgaben (HTML, CSS, JS) über lokalen Web-Server & Web-Client testen (E2E)

Ansatz: Single-Container-Anwendung mit lokaler Client-Server-Architektur und Client-Schnittstelle für Unittests

→ angepasstes Single-Container-Images

→ Start mehrerer Prozesse in einem einzelnen Container

→ Steuerung des Web-Client mittels Selenium-Webdriver



Beispiel: Adapter für Rust

Bessere Wissensvermittlung im Bereich Systemsoftware, aber wie?

Szenario: Programmiersprache Rust

- robuste und effiziente Implementierungen
- hardwarenahe Speicherverwaltung
- nichtfunktionale Tests

```
1 main.rs  2 thread.rs
 1 //! Module for the [`Thread`] trait, which should be implemente
 2 //! type [`MyThread`].
 3
 4 use crate::Dtq;
 5 use std::fmt::Display;
 6
 7 /// Trait for thread behavior.
 8 // `Display` is called a _supertrait_ of `Thread`, which means
 9 // implementing `Thread` has to implement `Display` as well
>> pub trait Thread: Display {
11     /// Simulates a thread working for a maximal interval of `d
12     /// quants.
13     ///
14     /// Returns the duration of effective computation activity
15     fn do_work(&mut self, dur: Dtq) -> Dtq;
16 }
```

Ansatz: Entwicklung eines Rust-spezifischen Adapters für Integration in CodeOcean

- Schnittstelle zum Web-Frontend: Anzahl ausgeführte Tests, Anzahl fehlgeschlagene Tests, Fehlermeldungen
- Ausgaben des Build-Tools cargo extrahierbar
- strukturiertes JSON-Format genutzt, z.B.:


```
{"type": "suite", "event": "ok", "passed": 4, "failed": 0, "ignored": 1,
    "measured": 0, "filtered_out": 0, "exec_time": 0.000473204}
```

Vielen Dank für ihre Aufmerksamkeit!

Möchten Sie mehr wissen?

- Jonas.Genath@tu-ilmenau.de
- Daniel.Fischer@tu-ilmenau.de
- Ulf.Doering@tu-ilmenau.de
- Gunther.Kreuzberger@tu-ilmenau.de
- Peter.Amthor@tu-ilmenau.de