

Python - Programmmentwurf 2022

Michael Möbius

Voraussetzung

- Jeder muss 5 von 6 Übungsaufgaben gemacht und abgegeben haben

Gruppenarbeit

- Gruppen von 4-5 Studenten
- Jeder Student muss seinen Teil beitragen
- Gruppen:
 - <https://docs.google.com/spreadsheets/d/16rNso80S0PA0pcdUhcfilVB-uYvzmrQttl7M37QTL4/edit?usp=sharing>

Kniffel

- <https://de.wikipedia.org/wiki/Kniffel>
- Regeln aus Wikipedia einhalten

Regeln (Aus Wikipedia)

- Vereinfachung auf 2 Spieler

Aufgabe - Kniffel

- Zwei Spieler an einem PC/Konsole
- Spielstand aus Datei einlesen und in Datei abspeichern
 - Format (z.B. Json, xml,...) kann frei gewählt werden
- Spiel Benutzerinterface
 - Jede Gruppe erstellt eigene Ein- und Ausgabe über **Terminal/Konsole**
 - Eine grafische Oberfläche ist nicht erlaubt!
- Wenn möglich objektorientiert programmieren
- Automatisierte Tests (75% Coverage pro Datei)
- Kleine Bibliotheken können nachinstalliert werden (z.B. für Dateizugriffe,.. keine Game-Engine)
- Lauffähig Windows und Linux (Pfade für Dateien, Konsole clear)

Automatisierte Tests

- Es wird **pro Datei** eine Testabdeckung von mindestens 75% gefordert
 - Selbst überprüfen mit “Coverage” Tool
 - Test müssen sinnvoll sein (validieren)
- Tests müssen selbst entwickelt werden
 - Tests fließen in die Bewertung ein

Aufgabe - Erweiterung - 5 Personen

- Gruppen mit 5 Studenten entwickeln zusätzlich einen Computer gegner
- Muss sich auch an Regeln halten
- Muss nicht intelligent sein

Zeitraum

- 2 Monate Zeit
- 01.03.2022 - 30.04.2022 23:59
- Abgabe über Moodle!
 - Nur bei Problemen an dhbw@michaelmoebius.de
 - Als Anhang in der Mail (zip)

Umsetzung in Python

- Aktuelle Python version verwenden
- Python 3.8 - 3.10 erlaubt
- Pylint mindestens 2.4.4

Versionsverwaltung

- Es sollte (kein muss) git verwendet werden
 - **Privates** repository (kein Zugriff für andere Studenten!)
 - Nur lauffähiger Code sollte committed werden
 - Kostenlos z.b. bei Github

Lint

- Alle Lint warnungen sollen behoben werden (Auch kommentare)
- Keine Ausnahmen außer den hier genannten
- Falschmeldungen müssen erklärt werden (Wenn berechtigt, kein Punktabzug)
- Pylint
 - Zeilenlänge 160
 - z.B. mit .pylintrc Datei
 - Keine Dokumentation in Tests nötig
 - In test Dateien:
`# pylint: disable=C`
 - In tests darf auf “protected members” zugegriffen werden
 - test.py: W0212: Access to a protected member _board_arr of a client class (protected-access)
 - `# pylint: disable=protected-access`

Dokumentation

- Siehe Bewertung.doc

Bewertung

- Ihr erstellt eine eigene Abschätzung zu euren Projekt anhand der Tabelle
- Wird mit 40% in die Note Programmieren eingerechnet (Maximal 40 Punkte)
- Die anderen 60 Punkte kommen von Maus/Staudacher

Tipps

- Pylint über Konsole ausführen
- Keine weiteren Pylint Deaktivierungen verwenden
- Tests sind in dieser Arbeit sehr hoch bewertet
- Achtet auf das schließen von Dateien beim Lesen/Schreiben
- Exceptions bei Dateizugriffen abfangen
- Verwendet innerhalb von Klassen und Methoden den “_” für private Felder/Methoden