

Teamarbeit ...

- ▶ EPR 5 ist eine Aufgabe, die im **Zweierteam** zu erledigen ist.

▶ Ihre Aufgaben

1. Eine Partner*in **in** Ihrer Übungsgruppe finden!
Es gibt keine Ausnahmen! Ungerade Anzahl von Teilnehmern in der Übungsgruppe → **Tutor per email ansprechen!**

2. Für die Abgabe die **__author__**-Variable richtig setzen.

```
__author__ = "1234567, Nachname1, 7654321, Nachname2"  
oder  
__author__ = "1234567, Müller, 7654321, Meyer"
```

3. Dann **gemeinsam** die Aufgabe bearbeiten!

Einführung in die Programmierung

Prof. Dr. Franziska Matthäus
Prof. Dr. Matthias Kaschube
Dr. Karsten Tolle
Lukas Müller

Übungsblatt ÜE-05

Ausgabe: 24.11.2021
Abgabe: 11.12.2021
16:00 Uhr

Ein Spiel

Hinweis:

- Es sind grundsätzlich Rechenwege anzugeben, es sei denn es findet sich ein expliziter Hinweis, dass dies nicht nötig ist.
- Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.
- Digitale Abgaben, die nicht im Format **.pdf** oder **.txt** für Texte oder **.py** für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrerer Dateien müssen diese als **.zip** zusammengefasst werden.
- Achten Sie darauf die Variable **__author__** in allen Quellcode Dateien korrekt zu setzen (am Anfang des Quellcodes):
`__author__ = "<Matr-Nr>, <Nachname>"` bzw. bei Zweierteams
`__author__ = "<Matr-Nr_1>, <Nachname_1>, <Matr-Nr_1>, <Nachname_1>"`
- Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein.
- Abgaben, die per Hand geschrieben und eingescannt werden, sind **nicht** erlaubt (bzw. geben 0 Punkte und werden nicht korrigiert).
- Beim Programmieren und Kommentieren halten Sie sich die Regeln im Programmierhandbuch, siehe Moodle-Kurs (<http://www.python.org/dev/peps/pep-0257>). Im Zweifelsfall gilt PEP 8.

Hinweis: Dieses Aufgabenblatt ist mit einer Bearbeitungszeit von zwei Wochen dafür ausgelegt in einem Zweierteam gelöst zu werden. Neben der Implementierung wird bei der Bearbeitung Wert auf die folgenden Punkte gelegt: Dokumentation (auch die Benutzerführung sollte erklärt werden (Benutzerhandbuch), was bei dem BMI-Beispiel durch fehlende Komplexität nicht nötig war), Docstrings (<http://www.python.org/dev/peps/pep-0257>), Strukturierung und Einhalten des Style-Guides.

Vorgehen (1)

- › **Pair Programming** ist eine zentrale Technik aus dem eXtremeProgramming(XP) – einem Beispiel für agile Techniken.
- › Beim Pair Programming sitzen zwei Entwickler*innen gleichberechtigt an einem Rechner und arbeiten gemeinsam an einer Aufgabe.
- › Die zwei Entwickler nehmen unterschiedliche Rollen ein, welche oft mit „**Pilot**“ und „**Navigator**“ bezeichnet werden.
 - › Der „**Pilot**“ schreibt den Code, während
 - › der „**Navigator**“ die Korrektheit des Codes und des Lösungsansatzes überwacht und parallel über Verbesserungen am Design nachdenkt.
- › Der Navigator kann sich zum Beispiel auch **Testfälle überlegen** (wenn das nicht schon vorher gemacht ist!)



Vorgehen (2)

- › Weil beide Entwickler **gleichberechtigt** sind, gibt es keine feste Aufgabenteilung. Deshalb wechselt der „Pilot“ z.B. jede Stunde zum „Navigator“ und der „Navigator“ wird zum „Piloten“.
- › Achtung: Auch in den Paaren werden beim nächsten Mal die Partner gewechselt (Aber dazu später im Jahr.) **So wird sichergestellt, dass jedes Teammitglied alle Teile des Projektes kennenlernt. Das wird im Code Review überprüft ... sonst "keine Punkte,,**
- › Das funktioniert am besten, wenn die Programmierkenntnisse der Partner*innen etwa gleich sind!
- › *" Das mache ich nicht! – das ist ja viel zu aufwendig!"*
VORSICHT! ... und über etwas zu reden, dass ich nicht wirklich kenne ist **auch blöd und unakademisch! ... bitte einfach mal ausprobieren**😊



Ziele des Pair Programming

- › **Steigerung der Software Qualität**
- › Die Kontrollfunktion der/des zweiten ProgrammiererIn sollen "problematische" (z.B. trickreiche) Lösungen vermeiden.
- › Das Pair Programming dient der Verbreitung von Wissen über den Quellcode.
- › Durch das regelmäßige Rotieren der Partner wissen beide über den Quellcode bescheid: keine Expertenrollen.
- › Bei uns zusätzlich: **schnelleres Erkennen einfacher Fehler** – fachliche Reflektion wird unterstützt.



Vorteile des Pair Programming (1)

vergl. Wikipedia Paarprogrammierung

- › Insgesamt weniger Fehler im Code: **ca -15 %**
- › Kleinere Programme: **ca. 20% kleiner**
- › Höhere Disziplin
- › Besserer Code: weniger Sackgassen → verständlicherer Code, höhere Qualität
- › Belastbarer Flow: (aus der Psychologie), d.h. sehr hohe Motivation – Unterbrechungen werden auf diese Art besser abgewehrt.
- › Freude an der Arbeit:
90 % der Entwickler sprechen von einer erfreulicheren Arbeit.



Vorteile des Pair Programming (2)

vergl. Wikipedia Paarprogrammierung

- › Geringeres Risiko, weil das gesamte Projektteam Wissen über den Gesamtcode hat → geringeren Projektrisiko hinsichtlich der Mitarbeiterfluktuation und Mitarbeiterabwesenheiten.
- › Wissensvermittlung: Jeder hat Wissen, das andere nicht haben. Paarprogrammierung ist ein **Anlass dieses Wissen zu teilen**.
- › Teambildung: Die Programmierer*innen lernen sich gegenseitig schneller kennen, wodurch die Zusammenarbeit verbessert werden kann.
- › Weniger Unterbrechungen: Paare werden seltener unterbrochen als jemand, der alleine arbeitet.



Nachteile des Pair Programming

vergl. Wikipedia Paarprogrammierung

- › Produktivität und Kosten: Geringere Geschwindigkeit bei der Programmierung: Ein Zweierteam schafft ca. 170% im Vergleich zu 200% bei zwei Personen → **die Produktivität pro Person reduziert sich also pro Programmier*in um 15%.**
- › (Befürworter der Paarprogrammierung behaupten aber, dass die Produktivität nicht sinke, sondern im Gegenteil sogar deutlich steige. Grund dafür sei, dass die gesteigerte technische und fachliche Qualität genau dort wo die Produktivität erhöhen, wo während der Softwareentwicklung am meisten meiste Zeit verbracht wird: Beim Fehlerfinden und -beheben, sowie beim Lesen von Code.)
- › Urheberrecht
- › Haftung

