

# **Programmmentwurf**

## **TINF22B5**

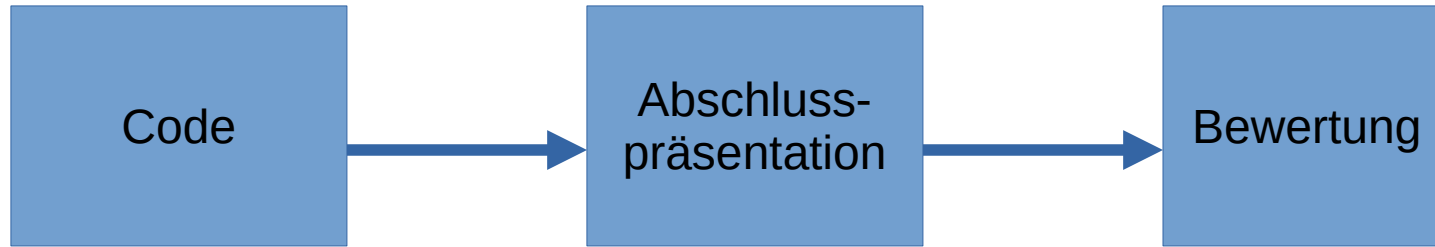
Umfang und Bewertung

# Programmentwurf

## 1.1.8 Programmentwurf (PE)

Ein Programmentwurf umfasst die Bearbeitung einer Aufgabenstellung mit der Auswahl geeigneter Methoden, der Formulierung der verwendeten Algorithmen in einer Programmiersprache, das Testen und Überprüfen der Ergebnisse auf Richtigkeit und die Programmdokumentation.

[https://www.dhbw.de/fileadmin/user\\_upload/Dokumente/Amtliche\\_Bekanntmachungen/2019/14\\_2019\\_StuPrO\\_Technik\\_inklusive\\_Dritter\\_Aenderungssatzung.pdf](https://www.dhbw.de/fileadmin/user_upload/Dokumente/Amtliche_Bekanntmachungen/2019/14_2019_StuPrO_Technik_inklusive_Dritter_Aenderungssatzung.pdf)



- Es wird eine Applikation entwickelt, die bestimmte Vorgaben erfüllen muss (s. „Protokoll Benotung“)
- $\geq 2.000$  Zeilen Code
  - +1000 Zeilen je weiterer Person
- Java, C#, Python (typisiert)
- $\geq 20$  Klassen
  - +5 Klassen je weiterer Person
- keine GUI, nur textbasierte Ausgabe

# Source Code

- Versionskontrolle (git)
- vollständiges Repository abgeben (hochladen)
- vollständiger Source Code
  - kompilierbar
  - testbar
  - ausführbar
- “Works on any (reasonable) machine.”
  - mein PC: aktuelles Fedora-Linux
- 3rd-Party-Libraries **nach** Absprache erlaubt
- Frameworks (z.B. Spring) sind nicht erlaubt

# Source Code (Fortsetzung)

- Build Tools (Maven, Gradle, Make, ...) sind erlaubt (und gewünscht)
- Docker zur Auslieferung ist möglich, wenn der Container unter Linux läuft
- die Applikation sollte mit einem Klick / einem Script vollständig starten
- alles UTF-8 codiert
- Dokumente (falls benötigt) immer als PDF

# Abschlusspräsentation

- Präsentation der Applikation
  - Starten und 1 – 2 Usecases zeigen
  - Begründung für die eingesetzten Technologien
  - Einhaltung der Rahmenbedingungen (Code–Zeilen, ...)
- 30-45min Fragen zur Applikation inkl. allgemeinem Hintergrundwissen
- (optional) Folien dürfen mitgebracht/genutzt werden
  - wird empfohlen v.a. bzgl. der Zeit
  - z.B. um UML-Diagramme vorzubereiten, Code-Stellen aufzuarbeiten, etc.

# Themenbeispiele

- Kalender
- Eventmanager
- Textadventure
- Musikplayer
- Email-Client
- Chat (P2P, Server/Client)
- Filesharing (P2P, Server/Client)
- Backup-Manager
- **Eigene Ideen einreichen!**
- First come, first serve

# Gruppenarbeit

- Gruppenarbeit ist prinzipiell erlaubt
- max. 3er Gruppen
  - die Einzelleistung muss sichtbar sein
- alle Commits müssen im Repository vorhanden sein (um nachvollziehen zu können, wer was umgesetzt hat)
  - es muss also jeder seine Teile mit seinem Account “comitten”
- die Gruppen werden zusammen bewertet



# Gruppenarbeit (Fortsetzung)

- jeder kann (und wird) in der mündlichen Vorstellung zu allem befragt
- bei sichtbaren Unstimmigkeiten werden die Studierenden einzeln mündlich befragt und es kann dann eine voneinander abweichende Bewertung geben

# Abgabe

- vollständige Abgabe muss bis einschließlich 4.5. statt finden
  - Abgabe heißt: Zugang zum Git Repository, das den Code enthält
  - der Zugang kann direkt zu Beginn geschickt werden
- Die Abschlusspräsentation findet in KW19 statt (5.5. - 9.5.)

# Einreichung der Themen

- Themen bis einschließlich 27.10. per Mail an [maurice.mueller@dhbw.mamem.de](mailto:maurice.mueller@dhbw.mamem.de)
- Stichpunktartig die Technologie und die Eigenschaften auflisten
  - z.B.
    - HTTP Server zur ToDo-Listen-Verwaltung
    - Java, Gradle
    - REST-API mit XYZ
    - Feature 1
    - Feature 2
    - ....

# Ohne Rückfragen erlaubte Libraries

- C#
  - MSTest
  - Newtonsoft JSON Serializer
- Java
  - JUnit
  - JAssert
  - Jackson
  - GSON