



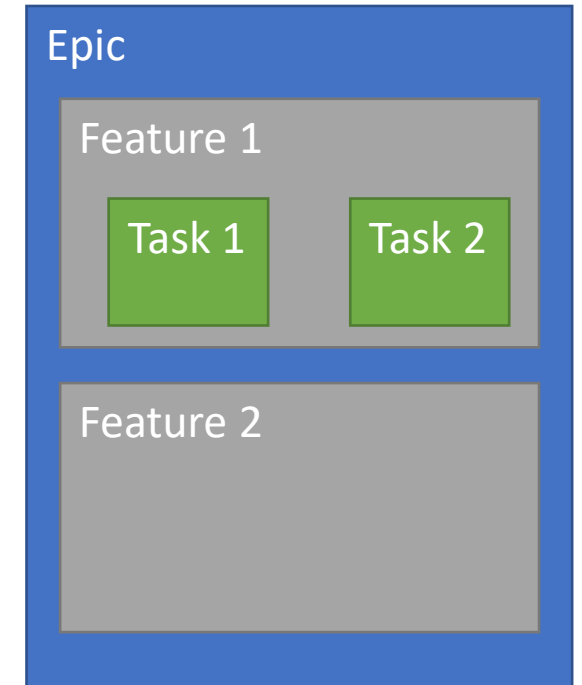
Tutorium

Programmieren mit Python

Überblick:

- Allgemeine Tipps
- Unterlagen
- Aufgaben
- Visualisierung
- Datenbank
- Objektorientierung
- GIT

- Das Problem / die Aufgabe analysieren und in kleinere Pakete zerlegen und abarbeiten
- KISS Prinzip (vor allem für Anfänger/Programmierer mit wenig Erfahrung, zu Beginn einfach halten, nicht zu komplex, erst sollte der Code laufen und dann können Optimierungen eingebaut werden)
- Vom „Einfachen“ zum „Schweren“



Ihr bekommt folgende Unterlagen für die Hausarbeit als Startpunkt:

- 4 Trainingsdatensätze
 - 1 Testdatensatz
 - 50 Ideale Funktionen
-
- Datensätze per Mail oder MS Teams bei Stephan Führer (Tutor) anfragen
 - Kein Ticket nötig ect.
 - Datensätze werden als .zip-Datei verschickt

1. Das Kriterium zur Selektion idealer Funktionen für den Training-Datensatz ist die Minimierung der Summe aller quadratischen y-Abweichungen (Least-Square).

-> Zur Auswahl / Bestimmung eines „Best-Fits“ zu einem Trainingsdatensatz muss die Summe der quadratischen Abweichung der y-Werte zwischen einem Trainingssatz und einer idealen Funktion gebildet werden

-> Diese Summe bildet dann einen Vergleichswert zum Vergleich des Trainingsdatensatzes mit anderen idealen Funktionen

-> So kann für alle 4 Trainingsdaten der „Beste-Fit“ gefunden werden

2. Dein Programm muss den Test-Datensatz B zur Validierung der Selektion benutzen. Hierbei soll für jedes x-y-Paar im Test-Datensatz überprüft werden, ob die Werte zu den vier idealen Funktionen passen.

-> Nach der Bestimmung der 4 idealen Funktionen sollen diese noch mit dem Testdatensatz validiert werden

-> Eine zusätzliche Überprüfung der gefundenen 4 idealen Funktionen

Aufgabe 2a

- a. Benutze ein Kriterium, welches sicherstellt, dass die maximale Abweichung zwischen der vorher ermittelten idealen Funktion und den Testwerten nicht die maximale Abweichung zwischen den Trainingsdaten (A) und den vier idealen Funktionen aus (C) um mehr als den Faktor *Wurzel aus zwei* ($\sqrt{2}$) übersteigt.

-> max. Abweichung zwischen (einer) idealen Funktion und Testdatensatz

M

-> max. Abweichung zwischen allen 4 Trainingsdaten und den dazugehörigen idealen Funktionen

N

$$M < (\sqrt{2}) * N$$

-> Für jede ideale Funktion M bestimmen und überprüfen!

Aufgabe 2b

- b. Sollten die Testdaten an die von Dir gefundenen vier Funktionen anpassbar sein, speichere für jeden Testdatensatz die entsprechenden Abweichungen ab.

-> Ist $M < (\sqrt{2}) * N$ dann ist die ideale Funktion bestätigt und kann abgespeichert werden
-> Für jede ideale Funktion eine eigene Tabelle erstellen in der Datenbank (es ist dabei den Studenten überlassen welche SQL Datenbank verwendet wird / keine Einschränkungen)

Spalte 1 und 2: Testfunktion / Datensatz der Testfunktion		Berechnete Abweichung aus y der Testfunktion und y der idealen Funktion	1 von 4 y Daten einer idealen Funktion
X (Test Funktion)	Y1 (Test Funktion)	Delta Y (Abweichung)	Nummer der Idealen Funktion (z.B. Funk37)
x1	y11	y21 = (y11 - y31) ²	y31
...
xn	y1n	y2n	y3n

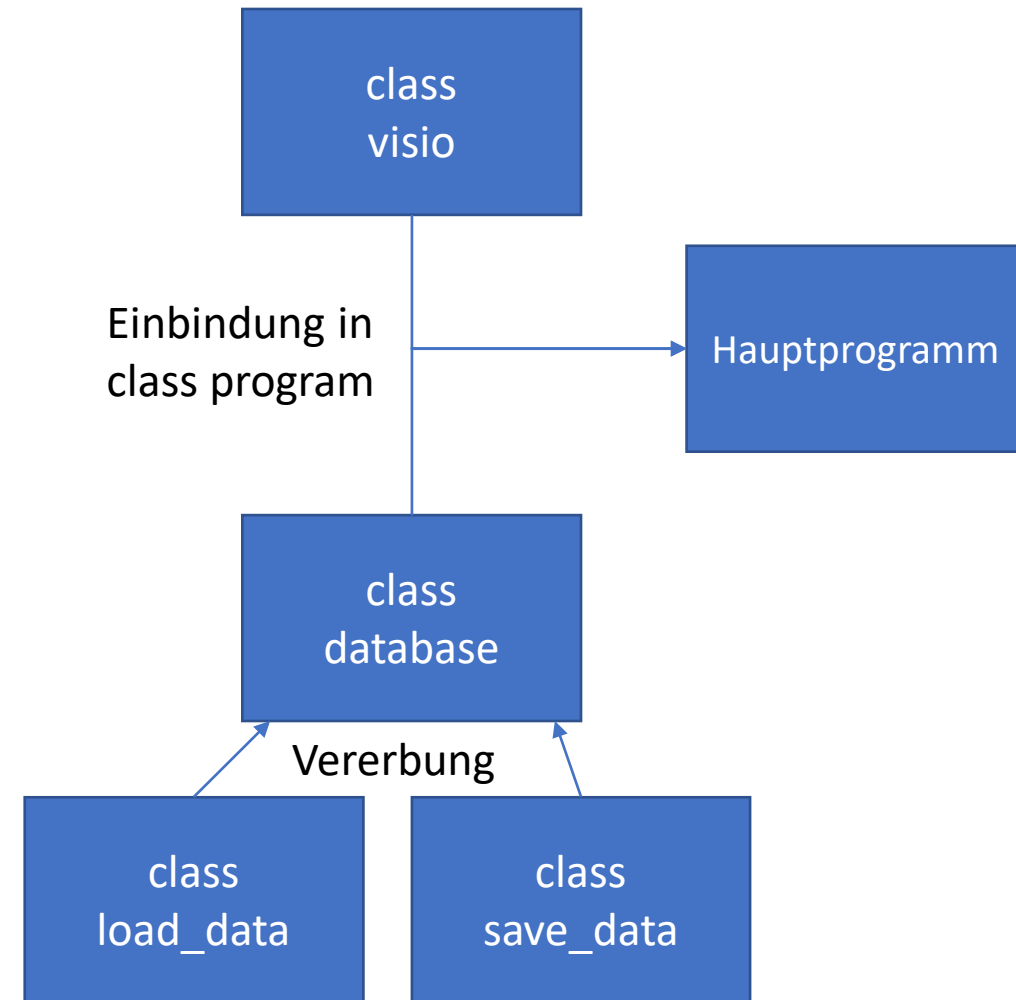
- Die Visualisierung kann mit einer beliebigen Library umgesetzt werden
- Wichtig ist die Ergebnisse zu Visualisieren (Plot als Line/Scatter Chart) und dies dann auch in der Hausarbeit einbauen
- Zwischenergebnisse oder andere Visualisierungen sind den Studenten überlassen



- Es steht den Studenten frei eine beliebige Datenbank für die Hausarbeit zu wählen (MySQL, MSSQL, SQLite, Oracle....)
 - Es gibt dabei auch keine Beschränkungen mit welchen Python Libraries / SQL Clients die Studenten arbeiten
- > Für die Abgabe ist nur ein kompiliertes SQL File nötig

Informationen zur Objektorientierung:

- Das Programm sollte modular aufgebaut sein
- Es sollte verschiedene Klassen haben
- Die Struktur / Aufbau bleibt den Studenten überlassen
- Ein Beispiel:



Das Arbeiten mit einer Versionsverwaltung ist für das spätere Berufsleben sehr wichtig und sollte deswegen auch in der Hausarbeit umgesetzt werden

- Arbeiten mit Branches (Master/Development)
- Muss nicht zu detailliert sein (keine Feature-Branches notwendig, aber schön zu sehen)
- Es geht hier nur um das Anwenden / Verwenden von Versionsverwaltungen
- Das Vorgehen / Arbeiten mit GIT sollte auch in der Hausarbeit beschrieben werden



Nach Fertigstellung eurer Arbeit gebt ihr bitte ab:

- Hausarbeit in PDF-Format Hochladen (Turnitin)
- Sourcecode Repository incl. Datenbank-Datei (SQL-File) als .zip-Datei an den Tutor per Mail

WICHTIG: DIE ERGEBNISSE MÜSSEN KLAR HERVORGEHOBEN WERDEN IN DER HAUSARBEIT!

Korrektur und Notenvergabe wird vom Tutor vorgenommen.

Bei Hilfe gerne per Mail oder MS Teams schreiben. Falls nicht gleich eine Antwort kommt auch noch einmal nachfragen!

Wir arbeiten aktuell an:

- Aufgabenstellung / Formulierung der Hausarbeit
- SQL Teil im Lehrbrief