



Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Studierendenprojekt

GradeCalc

Grundlagen Programmierung

EINLEITUNG



GradeCalc ist ein **Kommandozeilenprogramm** in **Python**, mit dem Studierende ihre **Note verwalten, berechnen und umrechnen** können.



Das Ziel von unserem Projekt ist es, ein **praxisnahes, übersichtliches Programm** zu entwickeln, das typische Probleme im Studienalltag für die Notenübersicht löst.

ZIEL DES PROJEKTES

HAUPTZIELE

Benutzerverwaltung mit **Login und Registrierung**

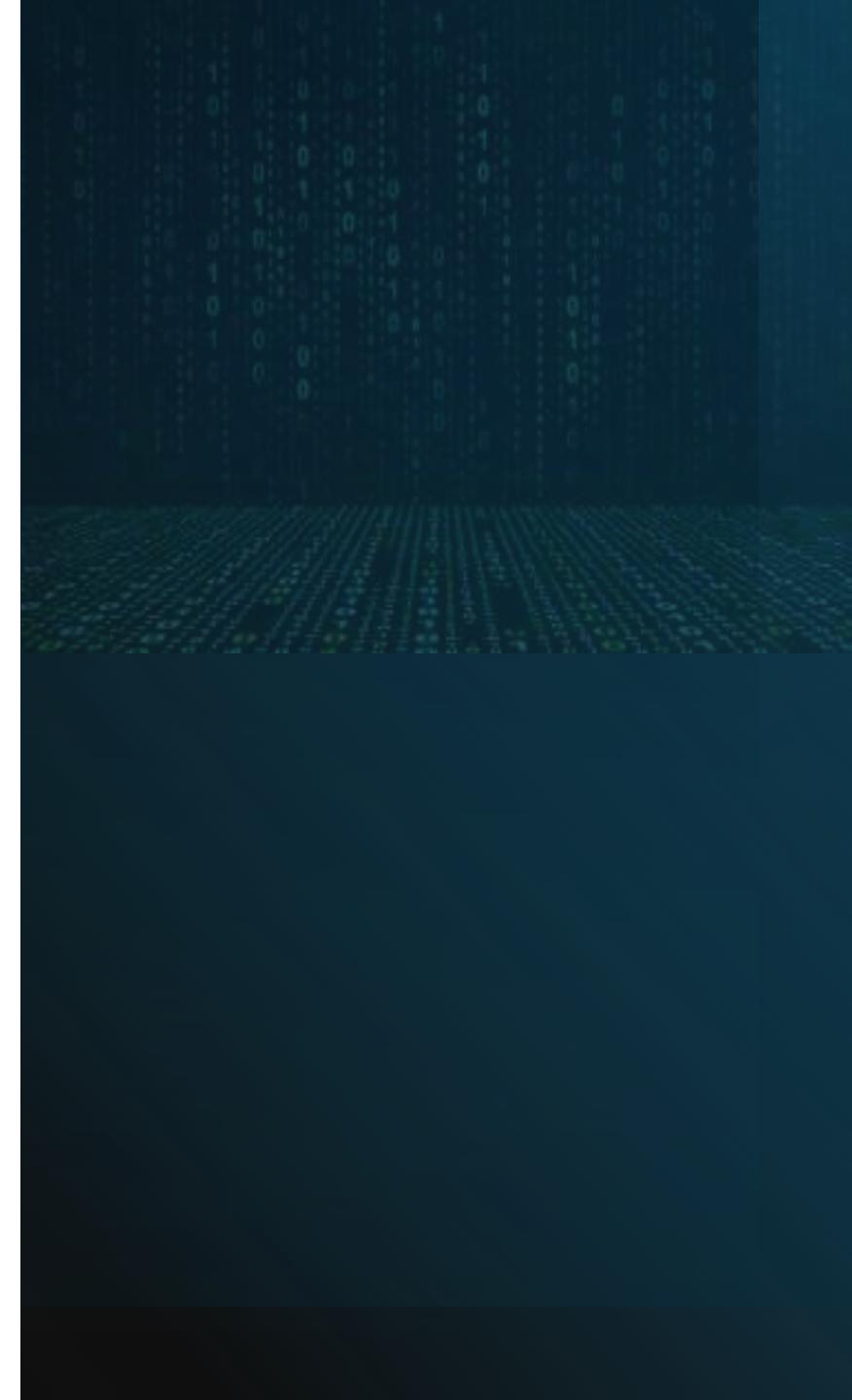
Noten **einfach erfassen und speichern**

Durchschnitte berechnen, mit und ohne Gewichtung

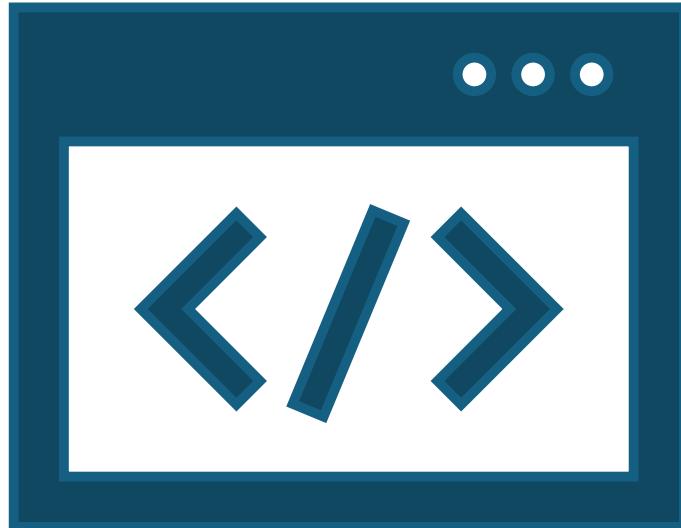
Noten aus Punkten berechnen

Notenskalen umrechnen (Schweiz, Deutschland, USA)

Saubere Fehlerbehandlung bei falschen Eingaben



AUFBAU UND STRUKTUR



Das Programm ist **modular aufgebaut**, das heisst:

- Jedes Modul hat eine **klare Aufgabe**
- Hauptmenü, Logik und Datenhaltung sind getrennt
- Wiederverwendbare Hilfsfunktionen (z.B. Farben, Stilmittel, säubern des Terminals)

Wichtige Module sind zum Beispiel:

- main_menu – Startmenü (Gastmodus)
- user_menu – Menü für eingeloggte Benutzer
- durchschnitt_berechnen
- notenrechnung
- notenskala
- Module für **Noten – und Modulverwaltung**

DATENHALTUNG

Die Benutzerdaten werden in einer JSON – Datei gespeichert

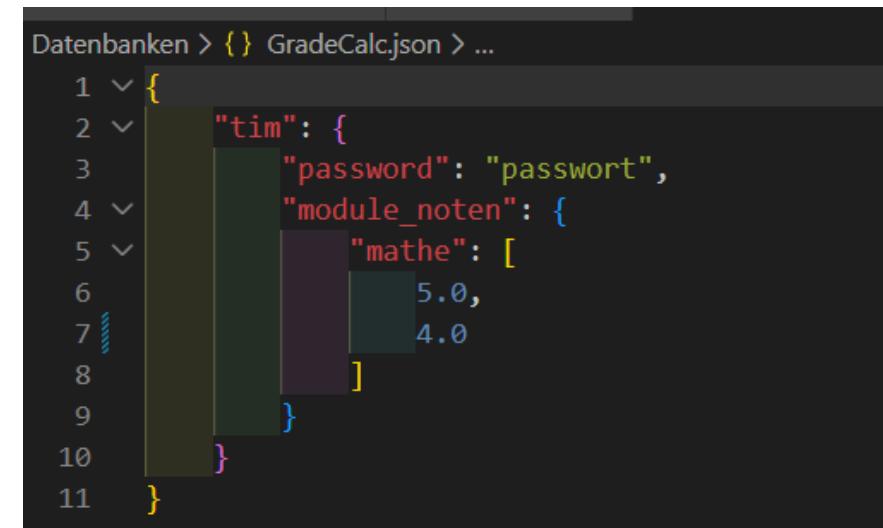
Struktur:

Jeder Benutzer hat:

- Ein Passwort
- Ein Benutzername
- Mehrere Module
- Jedes Modul enthält eine Liste von Noten

Warum JSON?

- Leicht lesbar
- Keine externe Datenbank nötig
- Perfekt für kleine bis mittlere Datenmenge
- Direkte mit Python kompatibel
- Ohne Datenbank overhead



```
Datenbanken > {} GradeCalc.json > ...
1  {
2   "tim": {
3     "password": "passwort",
4     "module_noten": {
5       "mathe": [
6         5.0,
7         4.0
8       ]
9     }
10  }
11 }
```

BENUTZERVERWALTUNG

Zwei Modi:

GASTMODUS

BENUTZER – MODUS (NACH LOGIN)

Durchschnitt berechnen

Modul hinzufügen/löschen

Noten aus Punkten berechnen

Noten speichern und entfernen

Notenskala

Durchschnitt pro Modul berechnen

Alle Funktionen von Gastmodus sind verfügbar

NOTENBERECHNUNG NACH PUNKTE



Hier berechnen wir eine Note aus erreichten Punkten anhand der maximalen Punktzahl

Welche Funktion willst du ausführen? 2

=====

Notenberechnung nach Punkten

=====

Maximal mögliche Punktzahl: 100

Erzielte Punktzahl: 75

Sehr gut deine Note **4.75** ist genügend!

Zurück zum Hauptmenü? (j/n):

NOTENSKALA UMRECHNUNG



Wir bieten eine Umrechnung von

- Schweiz - Deutschland
- Schweiz - USA



Wichtig

- Notenskala sind **nicht international standardisiert**
- Unsere Lösung nutzt **lineare Funktionen**
- Inputs werden auf **gültige Wertbereiche begrenzt**
- Wir kennzeichnen die Umrechnung bewusst als **Näherung**, nicht als offizielle Regel

```
=====
Notenskala umrechner
=====

Gib deine CH Note ein: 5

In welches Notensystem möchtest du umrechnen?

1) Deutschland
2) USA

Auswahl: 1

CH-Note: 5.00
DE-Note: 2.49
Zurück zum Hauptmenü? (j/n): █
```

FEHLERBEHANDLUNG & ROBUSTHEIT

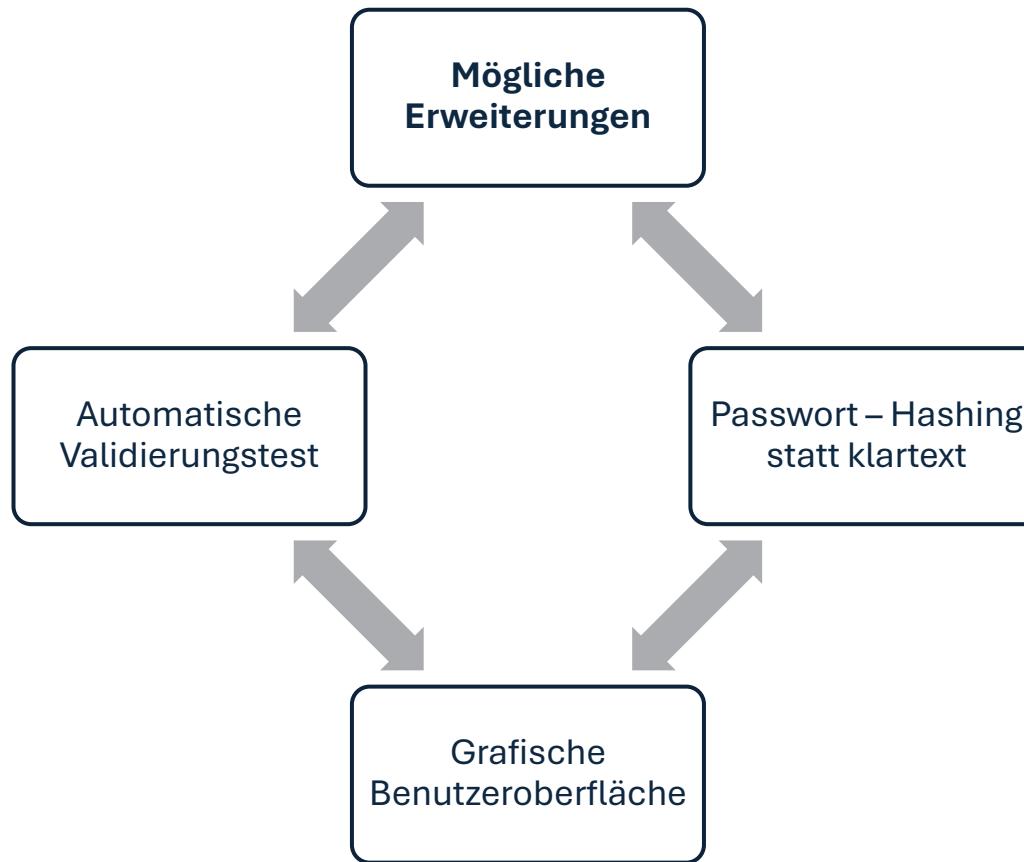
Besonderen Werte haben wir gelegt auf:

- Vermeidung von Programmabstürzen
- Klare Fehlermeldungen
- Wiederholung bei falscher Eingabe

Risikostellen:

- Keine Noten vorhanden
- Ungültige Inputs
- Leere Module
- Anmeldung ohne eine vorhandene Datenbank

VERBESSERUNGSMÖGLICHKEITEN



PROJEKTMANAGEMENT



Python Konsolenprogramm GradeCalc – Team Projekt

Für unser Projekt **GradeCalc** haben wir das Vorgehen geplant und umgesetzt. Dabei haben wir zuerst festgelegt, **was** das Programm können soll, **bis wann** es fertig sein muss, **wie** wir vorgehen, **wer** welche Aufgaben übernimmt, **welche Mittel** wir verwenden und **welche Risiken** auftreten können. Zusätzlich haben wir Massnahmen zur **Qualitätssicherung** eingeplant.

Was soll gemacht werden?

- Benutzer sollen sich **registrieren und anmelden** können.
- Noten sollen **dauerhaft gespeichert** werden (auch nach Programm-Neustart).
- Nutzer sollen **Module** anlegen und löschen können.
- Zu jedem Modul sollen **Noten** hinzugefügt oder entfernt werden können.
- Das Programm soll **Durchschnitte berechnen**, sowohl ohne Gewichtung als auch mit Gewichtungen.
- Zusätzlich soll eine **Notenberechnung nach Punkten** möglich sein.
- Ausserdem soll eine **Notenskala-Umrechnung** (z.B. Schweiz → Deutschland, Schweiz → USA) angeboten werden.
- Das Programm soll **robust gegen falsche Eingaben** sein und nicht abstürzen.

Bis wann?

Sonntag, 14.12.2025 23:55 Uhrzeit

intern Meilensteine:

- **M1:** Grundgerüst und Menüstruktur (Main-Menü / User-Menü)
- **M2:** Login/Registrierung und stabiler JSON-Speicher
- **M3:** Modul- und Notenverwaltung (hinzufügen/entfernen)
- **M4:** Berechnungen und Umrechnungen (Durchschnitt, Punkte→Note, Notenskala)
- **M5:** Tests, Bugfixing, README und Präsentation finalisieren

Wie?

Schrittweise Vorgehen

- **Anforderungen festlegen**
Wir haben zuerst definiert, welche Funktionen unbedingt enthalten sein müssen (z.B. Login, Speichern, Berechnungen).
- **Aufgaben in Teilbereiche aufteilen**
Das Projekt wurde in einzelne Module aufgeteilt, z.B. Menüführung, Benutzerverwaltung, Datenhaltung, Berechnungen.
- **Reihenfolge planen**
Wir haben bewusst zuerst die Basis implementiert (Menüs und Login), danach die Module/Notenverwaltung und erst anschliessend die Berechnungen und Umrechnungen.
Dadurch konnten wir früh testen, ob der Datenfluss funktioniert.
- **Schrittweise Umsetzung und Testen**
Jede neue Funktion wurde einzeln implementiert und direkt getestet, bevor die nächste Funktion hinzugefügt wurde.

PROJEKTMANAGEMENT

Python Konsolenprogramm GradeCalc – Team Projekt

Wer macht was?

Damit wir parallel arbeiten konnten, haben wir die Aufgaben klar untereinander aufgeteilt:

Tim :

- Anmelden.py (Login/Registrierung)
- Deine_Module.py (Anzeige der Module und Noten)
- User_menu (Menü für eingeloggte Benutzer)
- Module_hinzufügen_entfernen (Module verwalten)

Irfan

- Notenberechnung (Punkte → Note)
- Noten_hinzufügen_entfernen (Noten verwalten)
- durchschnitt_berechnen (Durchschnitt normal und gewichtet)
- ausserdem das **README** (Dokumentation)

Josselyn:

- ANSI (Farben)
- notenskala (Noten-Umrechner)
- Main_menu / Main_programm (Startmenü und Navigation)
- sowie die Struktur und Pflege der **GradeCalc JSON-Datei**

Mit welchen Mitteln?

Wir haben folgende Mittel genutzt:

Python als Programmiersprache

Visual Studio Code zum Schreiben und Testen des Codes

Git und **GitHub** zur Versionsverwaltung und Zusammenarbeit

JSON-Datei zur Datenspeicherung, da sie leicht lesbar und für ein Schulprojekt gut ist

Terminal/Konsole zum Ausführen und Testen des Programms

ChatGPT Hilfsmittel

Was kann schiefgehen?

Im Projekt haben wir typische Risiken berücksichtigt:

- **Falsche Eingaben** (z.B. Buchstaben statt Zahlen, Noten ausserhalb 1–6)
→ Lösung: try/except, Range-Checks, Wiederholungsschleifen
- **Programmabstürze wegen Datei-Problemen** (fehlende oder kaputte JSON-Datei)
→ Lösung: Datei prüfen/erstellen, sauberes Laden/Speichern
- **Login-Probleme durch Gross-/Kleinschreibung**
→ Lösung: Eingaben von **USERNAME** immer mit Kleinschreibung (z.B. .lower ())
- **Umrechnung der Notenskalen** ist nicht überall gleich standardisiert
→ Lösung: Umrechnung als lineare Approximation deklarieren, Grenzen definieren
- **Merge-Konflikte und doppelte Änderungen** bei Teamarbeit
→ Lösung: klare Aufgabenteilung, regelmässig push/pull, saubere Commits

Qualität sichern

Zur Qualitätssicherung haben wir folgende Punkte eingesetzt:

- **Tests mit Risikosteile**, z.B. 0 Punkte, maximale Punkte, leere Notenlisten
- **Fehlerbehandlung**, damit das Programm bei falschen Eingaben nicht abstürzt
- **Kommentare im Code** für bessere Verständlichkeit
- **Docstrings** für bessere Verständlichkeit von Funktionen und Klasse
- ein **README**, das Installation und Nutzung erklärt



Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Studierendenprojekt

TIM
IRFAN
JOSSELYN



BESTEN DANK