

# Grundlagen der Python-Programmierung

## Praxisprojekte

Zeitraum: 16.01.2024 – 22.01.2024

Präsentation am 22.01.2024 ab 13:15

Für dieses Praxisprojekt stehen drei verschiedene Optionen mit verschiedener Auslegung zur Verfügung. Entscheidet euch als Gruppe für eins dieser drei Projekte und bearbeitet dies gemeinsam! (Wenn ihr als Gruppe schneller durchsteigt und es wirklich nichts mehr zu perfektionieren gibt, dann geht **danach** ein weiteres oder alle an, diese gern dann auch individuell.)

Für alle Auswahlmöglichkeiten gilt:

- Fertigt eure Auswertungen in Python an. Nutzt alle Werkzeuge und alles Wissen, das ihr bereits habt. Es bleibt euch überlassen, ob ihr Skripte oder Jupyter-Notebooks benutzt.
- Dokumentiert eure Vorgehensweise und bereitet euch darauf vor, eure Schritte in der Präsentation zu erläutern. Für dieses Projekt ist eine PowerPoint-Präsentation nicht zwingend erforderlich, aber von Vorteil. Solltet ihr euer Programm live präsentieren und Code erklären wollen, dann achtet darauf, dass diese Demonstrationen möglichst kurz, informativ und vielleicht interaktiv sind. Zeile-für-Zeile-Demonstrationen sind nicht erwünscht, die Beleuchtung von einzelnen spannenden Stellen im Code dagegen zulässig.
- Ihr könnt rein "imperativ" programmieren und den Fokus auf die Datenanalyse und Verarbeitung legen. In allen Projekten kann aber auch die Modularisierung von Code eingesetzt und geübt werden um z.B. neue Funktionen und Klassen schreiben, welche weiterverwendet werden können
- Bringt ein, was ihr im Softskill-Modul gelernt habt, sprecht euren Projektplan ab, verteilt Aufgaben und organisiert euch!

Das Projekt gilt als Lernkontrolle. Ihr werdet in Gruppen benotet. Die Note setzt sich zusammen aus den Kategorien:

1. Datenauswertung/Programmierung
2. Projektdesign/-architektur
3. Visualisierung – Präsentation

Ladet am Präsentationstag bitte trotzdem jeder das Ergebnis des Projekts als Abgabe in Moodle und auf GITHUB.

Viel Erfolg!

### **Option 1 – Immobilien-Preisrechner**

(Fokus: Automatische Berechnung einer komplexen Preis-Formel; User-Eingabe; Interfaces; String Formatierung; Datenvalidierung)

### **Option 2 – Gasrechner**

(Fokus: Automatische Berechnung von Ausgaben per Nutzereingabe, beliebig komplex, mit GUI)

### **Option 3 – Kundendaten-Register**

(Fokus: Verschachtelte Datenstrukturen / JSON bearbeiten und speichern; Stammdatenverwaltung; Dateneingabe und Validierung)

## **Option 1 – Immobilienpreis-Rechner**

Gegeben ist eine Excel-Datei, die bestimmte Informationen zur Berechnung eines Schätzwertes für einen Immobilienpreis enthält. Studiere die Excel-Datei und die darin ausgeführte Berechnung. Der Preisfinder beurteilt anhand von 10 einfachen Fragen wie teuer eine Immobilie tatsächlich ist.

Folgende Fragen werden dazu gestellt:

- Wie viel m<sup>2</sup> Grundstück?
- Wie viel m<sup>2</sup> Wohnfläche?
- Handelt es sich um ein Architektenhaus?
- Soll ein Makler das Haus verkaufen?
- Steht das Haus oder Teile davon unter Denkmalschutz?
- Welches Baujahr hat das Haus?
- Ist das Haus in der Stadt oder auf dem Land?
- Wie ist die Ausstattung?
- Wie ist die Hausart?
- In welchem Bundesland steht das Haus?

Deine Firma möchte die Berechnung automatisieren und beauftragt dich, einen Prototypen für ein Programm zu schreiben, das an Kunden weitergegeben werden kann. Der Kunde soll durch Eingabe der Daten seiner Immobilie automatisch den Schätzpreis angezeigt bekommen.

Macht euch Gedanken darüber, auf welche Art der Nutzer seine Daten eingeben soll (über Input, die Konsole, eine grafische Benutzeroberfläche...) und setzt euch mit diesen "Interfaces" auseinander. Überlegt, wie ihr die für die Berechnung benötigten Daten aus der Excel-Datei in Python (per Hand oder automatisch) übertragen oder anderweitig speichern könnt. Die Validierung der eingegebenen Daten ist von großer Bedeutung, um sicherzustellen, dass nur korrekte und gültige Daten für die Berechnung verwendet werden.

Ihr dürft frei wählen, wie genau ihr das Programm ausgestaltet und organisiert. In der Präsentation sollte die Bedienung und Funktionsweise eures Programms erläutert werden.

## Option 2 – Gasrechner

Freiere Alternative zu diesem Projekt: Baut den von uns begonnen Gasrechner weiter aus! Dieser soll vom Nutzer Arbeitspreis, Grundpreis, Abschlagszahlung, Anzahl der Abschlagszahlungen (11 oder 12 gängig) sowie die Kilowattstunden vom Vorjahr (Vorjahresverbrauch) entgegennehmen und ausrechnen, was bei gleichbleibendem Verbrauch in diesem Jahr nachzuzahlen wäre oder zurückerstattet wird.

Wenn ihr das abgeschlossen habt, dann überlegt euch eine schöne „Verpackung“ für den Nutzer in Form einer grafischen Oberfläche, die den Umgang mit diesem Programm angenehm macht. Wenn das steht, dann könnt ihr noch einen Schritt weitergehen z.B. Preiserhöhungen mit berücksichtigen. Beispiel: Im Juli wurde abgerechnet, dann findet im November eine Preiserhöhung statt. Insgesamt gibt es elf Abschlagszahlungen, sodass August bis Oktober zu altem Preis und die restlichen Monate bis Juni zu neuem Preis berechnet werden müssten. Wenn auch das nicht genug ist, gern den Gasrechner zu einem Gas- und Stromrechner ausbauen.

## Option 3 – Kundendaten-Register

Du bist beauftragt, für deine Firma einen Prototypen eines Programms zum Management von Stammdaten zu entwickeln. Mit deinem Python-Programm sollen Informationen zu registrierten Kunden in einem flexiblen und austauschbaren Format gespeichert und bearbeitet werden können.

Pflichteinträge für jeden Kunden beinhalten seinen Namen, sein Alter, eine E-Mail Adresse, sowie einen oder mehrere Wohnorte und eine einzigartige ID. Die Daten jedes Kunden sollten aber auch beliebig mit neuen, optionalen Feldern wie z.B.: Hobbies, Haustiere, oder Ähnlichem erweitert werden können.

Die Kundendaten sollten persistent gespeichert sein – also nicht mit Ende der Python Sitzung gelöscht werden, sondern auf der Festplatte gesichert und in neuen Sitzungen wiederhergestellt werden. Mache dir deshalb Gedanken über geeignete Datenstrukturen und Dateiformate zur Speicherung der Kundendaten und über die Möglichkeiten des Datenimports und -exports.

Überlegt auch, wie ihr euren Code organisiert. Ihr könnt zum Beispiel neue Funktionen zum Hinzufügen, Löschen, Bearbeiten und Speichern von Kundendaten erstellen, oder sogar mit objektorientierter Programmierung (OOP) arbeiten und etwa Klassen für Kunden und KundenDatenBanken erstellen, welche alle Funktionalitäten beinhalten.

Der Nutzer sollte an zentraler Stelle Kundendaten Hinzufügen, Aktualisieren und Löschen können. Macht euch daher Gedanken zur Form der Dateneingabe (bzw. des interfaces). Soll der Nutzer seine neuen Angaben in einem Notebook, in der Kommandozeile, oder in einer graphischen Oberfläche machen? Bedenkt dazu die Sicht des Anwenders, aber auch eure eigenen Fähigkeiten in der Umsetzung.

In der Präsentation sollte die Bedienung und Funktionsweise eures Programms erläutert werden.