

Übungsskript Python

Softwaretechnologie und Datenmanagement
Berufsschule IT Unterstufe

Robert Wittek
Dr. Birgit Worat

2022/2023

Inhaltsverzeichnis

1 Python IDEs	2
2 Konsolenein-/ausgabe, Variablen und Datentypen	4
3 Arithmetische Operatoren	8
4 Kontrollstrukturen: Einfache Alternativen, Bedingte Verarbeitungen	12
5 Kontrollstrukturen: Mehrfache Alternativen	14
6 Kontrollstrukturen: Zählschleifen	16
7 Kontrollstrukturen: Kopf- und Fußgesteuerte Schleifen	17
8 Datenstrukturen: Listen	19
9 Persistente Datenspeicherung: csv-Dateien	21
10 Expertenaufgaben	23
11 Anhang	26

1 Python IDEs

1.1 PyCharm

Im Unterricht wird die Entwicklungsumgebung PyCharm CE verwendet. Sie ist auf den Workstations der Schule bereits installiert.

Wir empfehlen Ihnen, sich diese Software nach Möglichkeit auch auf Ihrem eigenen PC zu installieren, um eine Programmierumgebung zum Lernen zur Verfügung zu haben. Sie können auch gerne ein eigenes Notebook mit zum Unterricht bringen und dort verwenden.

Installationsanleitung (für private Windows-PC)

1. Laden Sie den Installer von PyCharm unter <https://www.jetbrains.com/de-de/pycharm/download/> herunter. Nur die Community Edition steht kostenfrei zur Verfügung.
2. Führen Sie den heruntergeladenen Installer (`pycharm-community-<version>.exe`) aus.
3. Sie müssen zustimmen, dass die App Änderungen auf Ihrem Gerät vornehmen darf.
4. Klicken Sie auf **Next** >.
5. Im Dialog „Choose Install Location“ klicken Sie ebenfalls auf **Next** >, wenn Sie mit dem Standardverzeichnis einverstanden sind.
6. Im Dialog „Installation Options“ empfiehlt es sich, die Optionen **Add Open Folder as Project** und für die Assoziationen mit `.py`-Dateien zu aktivieren, um Projekte und Dateien komfortabel aus dem Explorer mit PyCharm öffnen zu können.
7. Im Dialog „Choose Start Menu Folder“ klicken Sie wieder auf **Install**, wenn Sie mit dem vorgeschlagenen Eintrag einverstanden sind. Die Installation beginnt nun.
8. Nach erfolgreicher Installation klicken Sie auf **Finish**. Das Programm ist nun installiert.
9. Bei der Erstellung des ersten Projekts lädt PyCharm einen Python-Interpreter nach und installiert ihn. Sie müssen zustimmen, dass diese Änderungen auf Ihrem Gerät vorgenommen werden dürfen.

1.2 Alternativen ohne PC

Falls Ihnen kein PC zur Verfügung steht, gibt es Alternativen in Form von Apps für mobile Geräte und Webanwendungen. Diese sind im Allgemeinen allerdings nicht so komfortabel und leistungsfähig, wie eine integrierte Entwicklungsumgebung auf einem PC.

iOS-App Zum Beispiel: Pythonista (<http://omz-software.com/pythonista/>, abgerufen am 24.06.2022)

Webanwendung Zum Beispiel: <https://www.programiz.com/python-programming/online-compiler/>, <https://ideone.com>, <https://repl.it>, jeweils abgerufen am 24.06.2022

2 Konsolenein-/ausgabe, Variablen und Datentypen

2.1 Fachliche Inhalte

- print, input
- Variablen
- Standarddatentypen: Integer, Float, Boolean, String
- Konvertieren mit str(), int(), float()
- Konkatenieren von Strings mit +

2.2 Kontext

Das Einkaufszentrum CenterMall hat mehrere Stellen für Mitarbeiter im Sicherheitsdienst ausgeschrieben. Sie unterstützen die Personalabteilung bei der Erfassung und Auswahl geeigneter Bewerber*innen.

2.3 Handlungsschritte im SWE-Prozess

Anforderungen

Die Personalabteilung hat folgende Anforderungen festgelegt:

- Das Programm soll für eine*n Bewerber*in folgende Daten erfassen: Bewerbernummer (eindeutige Nummer - wird aufsteigend vergeben), Name, Vorname, Geschlecht, Straße, Hausnr., Postleitzahl, Stadt, Geburtsdatum, Gehaltsvorstellung, Einträge im Führungszeugnis vorhanden (ja/nein).
- Die eingegebenen Daten sollen anschließend zur Kontrolle ausgegeben werden.

Analyse

- a) Recherchieren Sie mit Hilfe des Pythonskripts wie Daten eingelesen, gespeichert und ausgegeben werden können. Fassen Sie Ihre Ergebnisse in Stichpunkten zusammen.
- b) Es ist gute Praxis, sich bereits vor der Implementierung Testfälle zu überlegen. Ein Beispiel ist dazu in Tabelle 1 bereits angegeben. Ergänzen Sie weitere Testfälle, die möglichst viele verschiedene Szenarien abdecken.

Entwurf

Python stellt verschiedene Datentypen zur Speicherung von Werten zur Verfügung.

- c) Recherchieren Sie, welche Datentypen für die verschiedenen Daten der Bewerber*innen (Name, Vorname, usw.) am besten geeignet sind und über welche Eigenschaften sie verfügen. Füllen Sie dazu Tabelle 2 aus.

Implementierung

- d) Setzen Sie die Anforderungen in ein Python-Programm um. Konvertieren Sie die Eingaben, falls erforderlich, in den gewünschten Datentyp. Konvertieren Sie die Variablen zur Ausgabe zurück in den Datentyp String. (*Hinweis:* Die Konvertierung nach bool für die Vorstrafen ist etwas komplizierter. Nutzen Sie entweder stattdessen einen String (ja/nein) oder recherchieren Sie selbständig nach der richtigen Lösung.)

Test

- e) Testen Sie Ihr Programm mit dem zuvor erstellten Testfall. Korrigieren Sie eventuell aufgetretene Fehler.

2.4 Erweiterungen

- a) Recherchieren Sie nach einem geeigneteren Datentyp in Python für das Geburtsdatum. Geben Sie in Stichpunkten mindestens zwei Vorteile dieses Datentyps an. Ändern Sie anschließend Ihr Programm so ab, dass es diesen Datentyp nutzt.
- b) Informieren Sie sich darüber, warum der Datentyp `float` für die Speicherung von Währungsbeträgen nicht gut geeignet ist und recherchieren Sie eine Alternative. Fassen Sie Ihre Ergebnisse in Stichpunkten zusammen. Ändern Sie Ihr Programm dann so ab, dass es diese Alternative statt `float` nutzt.
- c) Wenn Sie bereits mit Kontrollstrukturen in Python vertraut sind, erweitern Sie das Programm um eine Überprüfung auf ungültige und unplausible Eingaben, z. B. negative oder viel zu hohe Werte bei der Gehaltsvorstellung. In diesen Fällen soll eine Fehlermeldung ausgegeben und das Einlesen der Eingabe wiederholt werden.

Eingabedaten	erwartete Ausgabe
12, Peters, Laura, w, Beispielpfad, 3, 00012, Beispieldorf, 07.07.1977, 38000.00, nein	Bewerber: 12, Name: Peters, Vorname: Laura, Geschlecht: w, Straße: Beispielpfad, Hausnummer: 3, PLZ: 00012, Stadt: Beispieldorf, Geburtsdatum: 07.07.1977, Gehaltsvorstellung: 38000.00, Vorstrafen: False

Tabelle 1: Testfälle für das Skript zum Einlesen der Bewerberdaten

Attribut	mögliche Werte	geeigneter Datentyp	Wertebereich des Datentyps	Darstellung der Werte im Programm (mit Bsp.)
Bewerbernr.				
Vorname				
Nachname				
Geschlecht				
Straße				
Hausnr.				
PLZ				
Stadt				
Geburtsdatum				
Gehaltsvorstellung				
Vorstrafen				

Tabelle 2: Geeignete Datentypen für Bewerberdaten

2.5 Übungen

- a) Markieren Sie im unten stehenden Quellcode aus Listing 1 die folgenden Elemente:
1. eine Variable
 2. eine Ausgabe
 3. eine Typumwandlung von Integer nach String
 4. eine Typumwandlung von String nach Integer
 5. die Funktion zur Eingabe von Daten von der Konsole
 6. den Verknüpfungsoperator
 7. einen Kommentar
- b) Erläutern Sie in Stichpunkten, was der Quellcode aus Listing 1 macht.

```
1      # Eingabe der Daten
2      name = input("Bitte gib Deinen Namen ein: ")
3      beruf = input("Welchen Ausbildungsberuf hast Du? ")
4      jahre = int(input("Wie viele Jahre dauert Deine Ausbildung? "))
5
6      # Ausgabe der Nachricht
7      print("Hallo " + name + ",")
8      print("herzlich Willkommen am HNBK!")
9      print("Viel Erfolg für Deine Ausbildung zum " + beruf
10           + " in den kommenden " + str(jahre) + " Jahren.")
```

Listing 1: Begrüßung

3 Arithmetische Operatoren

3.1 Fachliche Inhalte

- Arithmetische Standardoperatoren: +, -, *, /, %
- Unterschiede bei der Division von int- und float-Werten
- Grundlagen der Erstellung von Testfällen, insb. Festlegung von Äquivalenzklassen
- Debugging, insb. Verstehen von Fehlermeldungen

3.2 Kontext

Ein Mieter des Einkaufszentrums CenterMall ist das Weinfachgeschäft „Feine Weine“. Das Geschäft möchte in den Onlineversandhandel einsteigen und evaluiert gerade verschiedene Algorithmen zur Versandkostenkalkulation. Dabei soll SystemsForAll unterstützen.

3.3 Handlungsschritte im SWE-Prozess

Anforderungen

Entwickeln Sie einen Prototyp des Moduls zur Versandkostenkalkulation, das nach folgenden Vorgaben arbeitet:

- Eingabedaten: Es sollen der Name des Weins, der Preis pro Flasche, und die Anzahl der bestellten Flaschen erfasst werden.
- Ausgaben:
 - Zusammenfassung der Daten über die Bestellung
 - Gesamtpreis brutto und netto
 - Ausgabe der Anzahl der für den Versand erforderlichen 6er-Kartons
 - Ausgabe der freien Plätze im letzten 6er-Karton

Analyse

- a) Definieren Sie schriftlich Testfälle zur Überprüfung Ihres Programms. Bilden Sie Äquivalenzklassen, um mit möglichst wenig Testfällen eine hohe Testabdeckung zu erreichen.
- b) Recherchieren Sie, wie in Python grundlegende arithmetische Operationen durchgeführt werden und fassen Sie Ihre Ergebnisse in Stichpunkten zusammen.

Entwurf

- c) Erstellen Sie einen Entwurf des Versandkostenalgorithmus in der Notation *Struktogramm* oder *Programmablaufplan*.

Implementierung

- d) Setzen Sie den Algorithmus in Python um.

Test

- e) Testen Sie gegenseitig Ihre Implementierung. Tauschen Sie Ihr Programm mit dem einer anderen Zweiergruppen und testen sie das fremde Programm gegen Ihre Testspezifikation. Geben Sie anschließend Ihr Testprotokoll an die andere Gruppe zurück und beheben Sie anhand des von der anderen Gruppe erhaltenen Testprotokolls alle gefunden Fehler in Ihrem Programm.

3.4 Erweiterungen

- a) Erweitern Sie das Programm so, dass die Daten von drei Weinen eingegeben werden können.
- b) Erweitern Sie das Programm so, dass eine Aufteilung in 6er-, 12er- und 18er-Kartons berechnet wird.
- c) Erweitern Sie das Programm um eine formatierte Ausgabe der Bestellinformationen.
- d) Wenn Sie bereits wissen, wie Schleifen in Python funktionieren, ändern Sie das Programm so, dass eine beliebige Anzahl von Weinen bestellt werden kann.

3.5 Übungen

```
1 n = 23
2 print(n+1)
3 print(n)
4 n+=1
5 print(n)
6
7 n*=7
8
9 print(n)
10 print(n / 100)
11 print(int(n/100))
12 print(n % 100)
```

Listing 2: Ergebnisse raten

- a) Was gibt das Programm aus Listing 2 bei Ausführung aus? Notieren Sie die Ausgabe schriftlich, *ohne das Programm am Computer auszuführen*.
- b) Schreiben Sie ein Programm, das ausgibt, welche Uhrzeit 1000 Minuten nach Mitternacht ist.

- c) Erweitern Sie Ihr Programm aus **b)**, so dass die Anzahl von Minuten nach Mitternacht eingegeben werden kann.
- d) Schreiben Sie ein Programm, das ausgibt, welcher Wochentag der 200. Tag des Jahres ist, wenn der 23. Tag des Jahres ein Sonntag ist. Tipp: Vergeben Sie Zahlen für die Wochentage.
- e) Bankautomaten zahlen den gewünschten Betrag gewöhnlich mit möglichst wenig, d. h. möglichst großen Geldeinheiten aus. Schreiben Sie ein Programm, dass eine solche Auszahlung simuliert: Nach Eingabe des gewünschten Auszahlungsbetrags soll die Auszahlung mit möglichst wenig Geldeinheiten der Größen 50 €, 20 €, 10 €, 5 €, 2 € und 1 € erfolgen.

Wichtig: Verwenden Sie für diese Übung ausschließlich die Kontrollstruktur Sequenz.

Beispiellauf:

Gewünschter Betrag: 191

Auszahlung erfolgt als

50-Euro-Scheine: 3

20-Euro-Scheine: 2

10-Euro-Scheine: 0

5-Euro-Scheine: 0

2-Euro-Münzen: 0

1-Euro-Münzen: 1

- f) Erweitern Sie das Programm aus **e)** zu einem Kassenautomat zur Wechselgeldrückgabe. Das Programm soll nach Eingabe des zu zahlenden Betrags und des tatsächlich gezahlten Betrags im ersten Schritt die zu viel gezahlte Differenz berechnen und anzeigen. Im zweiten Schritt sollen die zur Auszahlung verwendeten Geldeinheiten angezeigt werden. Wie in Aufgabe **e)** sollen dabei möglichst wenig Geldeinheiten verwendet werden. Im Unterschied zur Aufgabe 1 sollen allerdings auch Centbeträge ausgezahlt werden.

Wichtig: Verwenden Sie für diese Übung ausschließlich die Kontrollstruktur Sequenz.

Beispiellauf:

Zu zahlender Betrag: 47,21

Gezahlter Betrag: 50,00

Wechselgeld: 2,79

Das Wechselgeld wird ausgezahlt als

50-Euro-Scheine: 0

20-Euro-Scheine: 0

10-Euro-Scheine: 0

```
5-Euro-Scheine: 0
2-Euro-Münzen: 1
1-Euro-Münzen: 0
50-Cent-Münzen: 1
20-Cent-Münzen: 1
10-Cent-Münzen: 0
5-Cent-Münzen: 1
2-Cent-Münzen: 2
1-Cent-Münzen: 0
```

4 Kontrollstrukturen: Einfache Alternativen, Bedingte Verarbeitungen

4.1 Fachliche Inhalte

- Bedingungen, Logische Operatoren, komplexe Bedingungen mit and, or, not
- Alternativen in Python mit if und else

4.2 Kontext

Nachdem durch das Programm aus Abschnitt 2 die Daten der Bewerber*innen erfasst wurden, soll nun ein Programm zur automatischen Vorauswahl implementiert werden. Das Programm soll Bewerber*innen, die gewisse Kriterien nicht erfüllen, automatisch aussortieren. Die übriggebliebenen Bewerber*innen werden zu einem Vorstellungsgespräch eingeladen.

4.3 Handlungsschritte im SWE-Prozess

Anforderungen

Die Personalabteilung hat folgende Mindestanforderungen an die Bewerber*innen:

- Es dürfen keine Vorstrafen vorhanden sein.
- Sie sollen jünger als 50 Jahre sein.
- Sie sollen aus den PLZ-Regionen 40, 42, 45, 46, 47, 48, 58 oder 59 stammen.

Das Programm soll so erweitert werden, dass nach Eingabe der Daten ausgegeben wird, ob die Person zu einem Vorstellungsgespräch eingeladen oder abgelehnt wird. Im Fall einer Ablehnung sollen die Gründe, die zur Ablehnung geführt haben, ausgegeben werden.

Analyse

- a) Definieren Sie Testfälle für diese Bedingungen.

Entwurf

- b) Recherchieren Sie, wie die Bedingungen der Personalabteilung in Python formuliert werden können. Informieren Sie sich dazu über die Themen Vergleichsoperatoren, logische Operatoren und String-Funktionen.
- c) Formulieren Sie anschließend die Bedingungen der Personalabteilung schriftlich in Python-Syntax.
- d) Erstellen Sie ein Struktogramm, das mit Hilfe dieser Bedingungen den Algorithmus zur Prüfung der Bewerber*innen realisiert.

Implementierung

- e) Erweitern Sie das Programms aus Abschnitt 2 um die Vorauswahl der Bewerber, indem Sie das Struktogramm aus **d)** in Python implementieren. Informieren Sie sich dazu über die Umsetzung von bedingten Verarbeitungen und einfachen Alternativen in Python.

Test

- f) Testen Sie nun das Programm einer anderen Zweiergruppe mit Hilfe Ihrer Testspezifikation. Erstellen Sie ein Testprotokoll und geben Sie es der anderen Gruppe zurück. Lassen Sie ebenfalls Ihr eigenes Programm testen und beheben Sie anschließend alle gefundenen Fehler.

4.4 Übungen

- a) Ermitteln Sie die Ausgabe des folgenden Programms, indem Sie schriftlich die Teilausdrücke schrittweise auswerten, ohne das Programm am Computer auszuführen

```
1 a = 10;
2 b = -3;
3 c = False;
4 s = "foo";
5
6 print(a < 5 or b >=0)
7 print(a >= 10 and c and b == -3)
8 print(s=="foo" and b < 0 or (c and a > 100))
```

- b) Schreiben Sie ein Programm, das einen Nettopreis vom Benutzer einliest und ihn danach fragt, ob der ermäßigte oder der volle Mehrwertsteuersatz aufgeschlagen werden soll. Je nach Eingabe wird der entsprechende Bruttopreis ausgegeben.
- c) Schreiben Sie ein Programm, das 3 Dezimalzahlen von der Konsole einliest. Geben Sie das Maximum und das Minimum dieser 3 Zahlen auf der Konsole aus.
- d) Schreiben Sie ein Programm, das 3 Bool-Werte von der Konsole einliest. Es soll **True** ausgegeben, wenn mindestens 2 Werte wahr sind, ansonsten gibt es **False** aus.
Können Sie das Programm auch ohne eine if-Anweisung schreiben?

5 Kontrollstrukturen: Mehrfache Alternativen

5.1 Fachliche Inhalte

- Alternativen in Python mit `if`, `elif`, `else`

5.2 Kontext

Die Mietpreise pro m² im Einkaufszentrum sind von verschiedenen Faktoren abhängig. Es soll ein Programm entwickelt werden, mit dem sich Interessenten die verschiedenen Mietpreise anzeigen lassen können.

5.3 Handlungsschritte im SWE-Prozess

Anforderungen

Die Miete pro m² berechnet sich wie folgt: Zunächst wird ein Grundpreis ermittelt, der vom Stockwerk abhängt. Im Erdgeschoss beträgt die Quadratmetermiete 60 €, im 1. Stock 50 € und im 2. Stock 40 €. Je nach Lage im Stockwerk wird zudem folgender Aufschlag auf die Quadratmetermiete erhoben: Ladenlokal der Kategorie 1: 20 €, Kategorie 2: 10 €, Kategorie 3: kein Aufschlag.

Bei ungültigen Eingaben soll das Programm eine aussagekräftige Fehlermeldung ausgeben.

Analyse

- a) Identifizieren Sie die in den Anforderungen enthaltenen Anwendungsfälle und notieren Sie diese in einem UML Use-Case-Diagramm.
- b) Erstellen Sie basierend auf den Use-Cases schriftlich Testfälle.

Entwurf

- c) Entwerfen Sie den Algorithmus zur Umsetzung der Anwendungsfälle in einer der Notationen Struktogramm, Programmablaufplan oder Pseudocode. Verwenden Sie als Kontrollstruktur u. a. eine mehrfache Alternative.

Implementierung

- d) Implementieren Sie den Algorithmus in Python unter Verwendung der Schlüsselwörter `if`, `elif` und `else`.

Test

- e) Testen Sie nun das Programm einer anderen Zweiergruppe mit Hilfe Ihrer Testspezifikation. Erstellen Sie ein Testprotokoll und geben Sie es der anderen Gruppe zurück. Lassen Sie ebenfalls Ihr eigenes Programm testen und beheben Sie anschließend alle gefundenen Fehler.

5.4 Übungen

- a) Schreiben Sie ein Programm, das zu einer eingegebenen Zahl den entsprechenden Monat ausgibt (1 - Januar, 2 - Februar, usw.).
- b) Ein Möbelhaus berechnet die Auslieferungspreise für Möbel folgendermaßen. Setzen Sie die Logik jeweils in Python um:
 - i) Wenn die Entfernung mehr als 30 km beträgt, ist der Lieferpreis 30 Euro.
 - ii) Wenn die Entfernung mehr als 30 km beträgt, ist der Lieferpreis 30 Euro, sonst kostet es 10 Euro.
 - iii) Wenn die Entfernung ≤ 30 km ist, ist der Lieferpreis 10 Euro.
Wenn die Entfernung > 30 km und ≤ 70 km ist, ist der Lieferpreis 30 Euro.
Wenn die Entfernung > 70 km und ≤ 100 km ist, ist der Lieferpreis 50 Euro.
Wenn die Entfernung > 100 km ist, ist der Lieferpreis 100 Euro.
 - iv) Ergänzen Sie **iii)** wie folgt: Das Möbelhaus hat seine Kunden in drei Typen kategorisiert.
Wenn der Kundentyp „A“ ist, dann wird der Lieferpreis um 10 Euro reduziert.
Wenn der Kundentyp „B“ ist, dann wird der Lieferpreis um 20 Euro reduziert.
Wenn der Kundentyp „C“ ist, dann wird der Lieferpreis um 30 Euro reduziert.
Dabei darf der Lieferpreis nicht negativ werden.

- c) Schreiben Sie ein Programm, das zu einer eingegebenen Zahl einen Würfel mit einer entsprechenden Augenzahl anzeigt. Wird eine Zahl außerhalb des Wertebereichs von 1-6 angegeben, soll ein Würfel mit einem Fragezeichen als Auge ausgegeben werden.

So könnten die Würfel zum Beispiel aussehen:

```

-----
|   | | * | | * | | * * | | * * | | * * | |   |
| * | |   | | * | |   | | * | | * * | | ? |
|   | | * | | * | | * * | | * * | | * * | |   |
-----

```

6 Kontrollstrukturen: Zählschleifen

6.1 Fachliche Inhalte

- Schleifen in Python mit `for` und `range`

6.2 Kontext

Für Verwaltungsaufgaben soll eine Liste aller Verkaufsflächen eines Stockwerks erstellt werden.

6.3 Handlungsschritte im SWE-Prozess

Anforderungen

Die Verkaufsflächen pro Stockwerk sind beginnend mit der Zahl 12 durchnummeriert. Sie haben das Format `Stockwerk.Nummer`. Es gibt 30 Flächen pro Stockwerk. Es soll eine Liste für den 1. Stock erstellt werden. Sie soll genau dieses Format haben:

```

Verkaufsflächen Etage 1
-----
Verkaufsfläche 1.12
Verkaufsfläche 1.13
Verkaufsfläche 1.14
...
Verkaufsfläche 1.41

```

Analyse

Die Anforderungen sind so trivial, dass sie keiner weiteren Analyse bedürfen.

Entwurf

- a) Entwerfen Sie einen Algorithmus in einer der Notationen Struktogramm, Programmablaufplan oder Pseudocode. Verwenden Sie die am besten geeignete Schleife für die wiederholten Ausgaben.

Implementierung

- b) Implementieren Sie den Algorithmus als Python-Programm

Test

- c) Testen Sie Ihr Programm durch Vergleich der Ausgabe mit der in den Anforderungen vorgegebenen Ausgabe.

6.4 Erweiterungen

- a) Um lediglich Ausschnitte des Etagenplans sehen zu können, soll eine Start- und Endraumnummer angegeben werden können.
- b) Auf der Ostseite des Mittelgangs liegen die Räume mit den ungeraden Nummern. Schreiben Sie das Programm so um, dass es nur diese Räume ausgibt.
- c) Erweitern Sie das Programm so, dass es die Räume aller 3 Etagen (0, 1 und 2) ausgibt. Lösen Sie das Problem durch Einführung einer weiteren Schleife und nicht durch dreifaches Kopieren des vorhanden Codes.

6.5 Übungen

- a) Schreiben Sie ein Programm, das die Summe der Zahlen von 1 bis n berechnet:

$$\sum_{i=1}^n i = 1 + 2 + 3 + \dots + n$$

Der Wert n wird vom Benutzer eingegeben.

- b) Schreiben Sie ein Programm, das die Summe $\sum_{i=m}^n i$ berechnet. Die Werte m und n werden vom Benutzer eingegeben.

7 Kontrollstrukturen: Kopf- und Fußgesteuerte Schleifen

7.1 Fachliche Inhalte

- Schleifen in Python mit `while`

7.2 Kontext

Das Einkaufszentrum hat ein Gewinnspiel veranstaltet. Zur Auswertung der Teilnehmerdaten soll gezählt werden, aus welchen Postleitzahlbereichen die Teilnehmer kommen.

7.3 Handlungsschritte im SWE-Prozess

Anforderungen

Die auf den Teilnahmekarten angegebenen Postleitzahlen sollen mit Hilfe des Programms erfasst werden. Das Ende der Eingabe soll durch Eingabe eines speziellen Werts signalisiert werden. Danach soll die Anzahl der Postleitzahlen, aufgeschlüsselt nach ihrer jeweiligen Startziffer, ausgegeben werden, z. B.

PLZ-Bereich	Anzahl
-----	-----
0	7
1	2
3	1
...	
9	8

Analyse

- Identifizieren Sie die in den Anforderungen enthaltenen Anwendungsfälle und notieren Sie diese in einem UML Use-Case-Diagramm.
- Definieren Sie einen Testfall, indem Sie eine Liste von Postleitzahlen festlegen und dazu manuell die erwartete Ausgabe ermitteln.

Entwurf

- Begründen Sie schriftlich, warum eine Zählschleife (**for**-Schleife) für die Umsetzung dieser Anforderungen ungeeignet ist. Nennen Sie die am besten geeignete Schleife und begründen Sie Ihre Aussage.
- Entwerfen Sie einen Algorithmus zur Umsetzung der Anwendungsfälle mit der in **c)** identifizierten Kontrollstruktur in einer der Notationen Struktogramm, Programmablaufplan oder Pseudocode.

Implementierung

- Setzen Sie den Algorithmus in Python um.

Test

- f) Testen Sie Ihr Programm mit Hilfe des festgelegten Testfalls.

7.4 Übungen

- a) Es soll ein Programm zur Passwortabfrage erstellt werden.
- i) Das Programm fragt den Benutzer so lange nach einem Passwort, bis die Eingabe „Nixdorf“ erfolgt.
 - ii) Erweitern Sie das Programm um eine Fehlermeldung, die ausgegeben wird, falls das eingegebene Passwort falsch war.
 - iii) Begrenzen Sie die Anzahl der ungültigen Versuche auf 3. Danach wird das Programm mit einer Fehlermeldung beendet.
- b) Die Bevölkerung im Land A beträgt 60 Millionen und die Bevölkerung im Land B beträgt 4 Millionen. Die Bevölkerung im Land A nimmt jährlich um 0,5% ab, die Bevölkerung im Land B nimmt jährlich um 2,5% zu.
- i) Nach wieviel Jahren hat Land B mindestens 10% der Bevölkerung von Land A erreicht?
 - ii) Nach wieviel Jahren ist die Bevölkerung von Land B mindestens so groß wie in Land A?

Geben Sie die Bevölkerungszahlen mit zwei Nachkommastellen aus.

8 Datenstrukturen: Listen

8.1 Fachliche Inhalte

- Verwendung von Listen
- Schleifen über Listen mit for-in (mit und ohne Index)
- Listenfunktionen: len, sort

8.2 Kontext

Die Besucherzahlen des Einkaufszentrums sollen erfasst und ausgewertet werden.

8.3 Handlungsschritte im SWE-Prozess

Anforderungen

Es soll ein Programm entwickelt werden, das die Besucherzahlen des Einkaufszentrums für eine beliebige Anzahl von Tagen erfasst und folgende Auswertungen vornimmt: Berechnung des Durchschnitts, des Maximums und Minimums. Außerdem sollen die eingegebenen Werte sowie ihre Gesamtanzahl zur Kontrolle ausgegeben werden.

Analyse

- a) Identifizieren Sie die in den Anforderungen enthaltenen Anwendungsfälle und notieren Sie diese in einem UML Use-Case-Diagramm.
- b) Erstellen Sie Testfälle indem Sie Eingabedaten und die dazugehörigen erwarteten Ausgabedaten manuell ermitteln.

Entwurf

- c) Erstellen Sie einen Algorithmus zur Umsetzung der Anwendungsfälle in einer der Notationen Struktogramm, Programmablaufplan oder Pseudocode. Verwenden Sie zur Speicherung der Eingabewerte ein Array.

Implementierung

- d) Setzen Sie den Algorithmus in ein Python-Programm um. Verwenden Sie für das Array eine Liste.

Test

- e) Testen Sie nun das Programm einer anderen Zweiergruppe mit Hilfe Ihrer Testspezifikation. Erstellen Sie ein Testprotokoll und geben Sie es der anderen Gruppe zurück. Lassen Sie ebenfalls Ihr eigenes Programm testen und beheben Sie anschließend alle gefundenen Fehler.

8.4 Erweiterungen

- a) Erweitern Sie das Programm um die Berechnung und Ausgabe des Medians der Besucherzahlen. Der Median ist der Wert an der mittleren Position der nach Größe sortierten Liste. Bei einer geraden Anzahl von Werten, ist er der Mittelwert der beiden mittleren Elemente.

Um sich die Arbeit zu erleichtern, recherchieren Sie danach, wie eine Liste sortiert werden kann.

8.5 Übungen

- a) In einem Fitnessstudio können die Mitglieder an einem Fitnesstest teilnehmen, bei dem verschiedene Fähigkeiten wie Kraft, Ausdauer und Beweglichkeit getestet werden. Am Ende können 0 bis 4 Punkte erreicht werden, die folgende Bedeutung haben:

Punkte	Auswertung
0	sehr schlechte Fitness
1	schlechte Fitness
2	durchschnittliche Fitness
3	gute Fitness
4	sehr gute Fitness

Schreiben Sie ein Programm, bei dem der Benutzer seinen Punktwert eingeben kann und die textuelle Auswertung ausgegeben wird. Benutzen Sie dafür keine Kontrollstrukturen wie if, sondern speichern Sie die Texte in einer Liste.

Sorgen Sie außerdem dafür, dass keine ungültigen Zahlen eingegeben werden können (Kontrollstrukturen erlaubt).

- b) Ein Langstreckenwanderer notiert sich für jeden Tag die gewanderte Strecke in km und die benötigte Zeit in hh:mm. Berechnen Sie die durchschnittlich gewanderte Strecke pro Tag und die Durchschnittsgeschwindigkeit und geben Sie die Informationen aus.
- c) Eine Gruppe von Langstreckenwanderern möchte ihre Werte untereinander vergleichen. Wer ist der schnellste? Wer ist der langsamste? Welche Durchschnittsgeschwindigkeit hat die Gruppe erzielt?

9 Persistente Datenspeicherung: csv-Dateien

9.1 Fachliche Inhalte

- Importieren externer Module am Beispiel des csv-Moduls
- Öffnen und Schreiben von Dateien (with-Notation)
- Grundfunktionen des csv-Moduls
- Umgang mit strukturierten Daten

9.2 Kontext

Ein Einzelhändler des Einkaufszentrums hat seine Artikeldaten in eine csv-Datei exportiert. Diese soll eingelesen, verändert und wieder gespeichert werden.

9.3 Handlungsschritte im SWE-Prozess

Anforderungen

Es liegt eine csv-Datei mit der Spaltenstruktur [Artikelname, Lieferant, Lagerbestand, Verpackungseinheit, Einkaufspreis, Verkaufspreis] vor. Sie können die Datei auf der Moodle-Seite zu diesem Kurs finden. In der ersten Zeile stehen die Spaltenüberschriften. Die Datei soll eingelesen werden, um folgende Aktionen auszuführen:

1. Ausgabe aller Artikel mit einem Lagerbestand von weniger als 500 auf der Konsole und in einer zweiten csv-Datei.
2. Schreiben einer neuen Datei mit um 3% erhöhten Verkaufspreisen.
3. Schreiben einer neuen Datei aus der alle Artikel eines bestimmten Lieferanten entfernt wurden.

Analyse

- a) Erstellen Sie ein UML Use-Case-Diagramm, das Use-Cases zu allen drei Anforderungen enthält.
- b) Erstellen Sie Testfälle, indem Sie zu allen drei Anforderungen jeweils manuell eine csv-Datei mit der erwarteten Ausgabe erstellen. Gehen Sie davon aus, dass die Beispieldatei auf Moodle als Eingabe verwendet wird.

Entwurf

- c) Recherchieren Sie wie
 - das csv-Modul zum Lesen von csv-Dateien importiert wird,
 - eine csv-Datei zeilenweise eingelesen werden kann,
 - eine csv-Datei zeilenweise geschrieben werden kann.

Fassen Sie Ihre Ergebnisse schriftlich zusammen, um sie der Klasse präsentieren zu können.

Implementierung

- d) Erstellen Sie zu den drei Anforderungen jeweils ein Python-Programm, das diese Anforderungen erfüllt.

Test

- e) Testen Sie Ihre Programm durch Vergleich der tatsächlichen Ausgabe mit der erwarteten Ausgabe.

9.4 Erweiterungen

- a) Auf der Moodle-Seite befinden sich zusätzlich Artikeldaten in den Formaten XML und JSON. Recherchieren Sie nach geeigneten Modulen zum Einlesen dieser Daten und passen Sie Ihr Programm so an, dass es statt der csv-Datei entweder die XML- oder JSON-Datei verwendet.
- b) Implementieren Sie eine Menüstruktur, unter der Sie die Funktionen der drei separaten Programm zusammenfassen.
- c) Schreiben Sie eine Erweiterung, mit der die Daten von beliebig vielen weiteren Artikeln vom Benutzer eingelesen werden und an die csv-Datei angefügt werden.
- d) Visualisieren Sie die Lagerbestände der Artikel mit Hilfe eines Säulendiagramms. Sie können dazu z. B. die Bibliothek Matplotlib (<https://matplotlib.org>, abgerufen am 24.06.2022) verwenden.

9.5 Übungen

- a) Schreiben Sie ein Programm, das zwei csv-Dateien zu einer zusammenfügt.

10 Expertenaufgaben

10.1 Passwort

- a) Definieren Sie zunächst Regeln für ein sicheres Passwort. Tipps (jeweils abgerufen am 24.06.2022):
 - https://www.uni-due.de/zim/it-sicherheit/sicheres_passwort.php
 - <https://www.sicher-im-netz.de/dsin-muster-passwortkarte>
 - <https://www.verbraucherzentrale.de/wissen/digitale-welt/datenschutz/sichere-passwoerter-so-gehts-11672>

- b) Schreiben Sie ein Passwort-Prüfprogramm, das gemäß den Regeln aus **a)** feststellt, ob das Passwort akzeptiert wird.
- c) Schreiben Sie einen Passwort-Generator, der gemäß den Regeln aus **a)** ein sicheres Passwort erzeugt.

10.2 Spiel programmieren

Programmieren Sie das Spiel „Schiffe versenken“. Regeln finden Sie unter https://de.wikipedia.org/wiki/Schiffe_versenken (abgerufen am 24. 06. 2022).

10.3 Spiel programmieren

Programmieren Sie ein „Klick-mich“-Spiel. Ein Beispiel finden Sie unter <https://www.topster.de/klick-topsterschaft> (abgerufen am 24. 06. 2022).

10.4 Sportaktivitäten

Für die Gesundheit der Firmenmitarbeiter werden kostenlose Sportaktivitäten angeboten. Diese sind in Tabelle 3 dargestellt. Die Mitarbeiter sollen nach geeigneten Kursen suchen und sich eintragen können. Definieren Sie selber eine maximale Teilnehmerzahl, Kursdauer und ähnliches.

Aktivität	Termin
Mountainbike fahren	Montag bis Freitag ab 18 Uhr
Klettern	dienstags 20 bis 22 Uhr, Vorkenntnisse erforderlich
Aerobic Stufe I (Anfänger)	montags 17 bis 18 Uhr
Stufe II (mit Vorkenntnissen)	mittwochs 9 bis 10 Uhr
Stufe III (Experten)	freitags 20 bis 21 Uhr
Zumba	donnerstags 20 bis 21 Uhr
Billard	freitags ab 18 Uhr
Bowling	montags ab 19 Uhr
Yoga	mittwochs 18 bis 20 Uhr

Tabelle 3: Mögliche Sportaktivitäten für Firmenmitarbeiter

10.5 Zahlentheorie

In der Zahlentheorie sind glückliche Zahlen folgendermaßen definiert: Glückliche Zahlen sind natürliche Zahlen, die mit einem bestimmten Siebprinzip erzeugt werden. Man beginnt mit

einer Liste der positiven natürlichen Zahlen. Dann geht man die Zahlen der Liste durch, beginnend mit $x = 2$, und streicht jeweils jede x -te Zahl. Beim Abzählen der zu streichenden Zahlen werden die schon gestrichenen nicht mitgezählt, sondern nur die noch in der Liste stehenden. Auch beim Durchgehen der Liste, um das nächste x zu erhalten, werden die gestrichenen übergangen. Schreiben Sie ein Programm, das glückliche Zahlen anzeigt.

11 Anhang

11.1 Individuelles Lernen an der Berufsschule IT

Liebe Auszubildende,

das Arbeitsmaterial der Berufsschule IT finden Sie auf <https://www.moodle-hnbk.de>.

Loggen Sie sich über  ein.

Lernportale

Alle Kurse → Berufsschule IT → Alle Ausbildungsjahre

- Lernportal Softwaretechnologie und Datenmanagement – LF 5 und Lernportal IT mit Aufgaben inklusive Lösungen, Lernspielen, Quiz, Selbsttests
- Lernportal Englisch mit Vokabeltrainer
- Lernportal WG (Wirtschafts- und Geschäftsprozesse — in Arbeit bzw. auf Papier im Selbstlernzentrum)

SD-Skripte

Alle Kurse → Berufsschule IT → 1. Ausbildungsjahr → SD → SD Skripte LF 5

Fachlich-sprachliche Förderung

Alle Kurse → Allgemeines → Sprachförderung HNBK

Für Nicht-Muttersprachler haben wir in Moodle einen Bereich für die fachlich-sprachliche Förderung angelegt. Dieser ist momentan im Aufbau.

Mathematik-Lernsoftware

Die Mathematik-Lernsoftware KLSOFT finden Sie auf den Workstations im Bereich Mathematik. Diese können Sie als Kopie nutzen.