

INTOCODE: Python II

Allan Christmas Maheri
e-mail: allan-christmas.maheri@hs-hannover.de

24.07.2023 - 04.08.2023
09:00 - 13:00

25. Juli 2023

Inhalt

Inhalt

- ▶ Rückblick
- ▶ Klassen & Objekte
- ▶ Vererbung und Polymorphie
- ▶ Fehlerbehandlung
- ▶ Prüfungsvorbereitung



Rückblick

Rückblick

- ▶ Listen
- ▶ Tupel
- ▶ Dictionary



Aufgabe 1

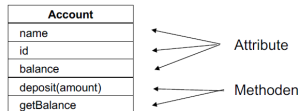
Schreibt ein Programm, welches einen String zu einen Dictionary konvertiert. Die Keys sollen dabei die Buchstaben sein und die Values die Positionen wo sich der Buchstabe im Alphabet befindet.

Motivation

- ▶ Funktionen sind unzureichend für die Verwaltung strukturierter Daten, da sie nur begrenzte Möglichkeiten bieten, komplexe Zusammenhänge abzubilden.
- ▶ Beispiel: Wie würde man ein Konto verwalten, wenn man nur Funktionen verwendet?
 - ▷ Eine typische unstrukturierte Lösung könnte so aussehen: ["Max", 22424, 52.23] # name, id, kontostand
- ▶ Die Verwendung von Klassen und Objekten führt zu einem besseren Code-Design, einer verbesserten Lesbarkeit und erleichtert die Wartung und Erweiterung der Software.

Objekte

- ▶ Objekte ermöglichen die Erstellung von Datenstrukturen und Funktionen, die zusammengehören und in einer einzigen Einheit organisiert sind.
- ▶ Klassen ermöglichen die Erstellung von Instanzen (Objekten), die individuelle Eigenschaften haben können, aber auf gemeinsamen Attributen und Methoden der Klasse zugreifen können.
- ▶ Klassen können mithilfe von Klassendiagrammen visuell beschrieben werden, um die Struktur und Beziehungen zwischen den Klassen darzustellen.





Definition

Hands-on-Session





Objekte erzeugen

Hands-on-Session



Objekte verändern

Hands-on-Session



Aufgaben 1

- ▶ Schreibt eine Klasse Person mit den Attributen name, gewicht und groesse
- ▶ Außerdem soll eine Funktion getBMI in der Klasse implementiert werden, welche den BMI der Person zurückgibt.
- ▶ Schreibe eine Methode innerhalb der Klasse Person, die das Gewicht der Person um einen bestimmten Wert erhöht

Aufgaben 2

- ▶ Erstelle eine Funktion außerhalb der Klasse Person, die zwei Person-Objekte als Parameter akzeptiert und vergleicht, welche Person einen höheren BMI hat.
 - ▷ Die Funktion soll den Namen der Person zurückgeben, die den höheren BMI hat.
 - ▷ Falls beide Personen den gleichen BMI haben, soll die Funktion Beide haben den gleichen BMI zurückgeben.

Aufgaben 3

- ▶ Schreibt die Klasse *Auto* mit den Attributen *marke*, *preis*, *modell* und *baujahr*. Implementiert für alle Attribute die entsprechenden Getter- und Setter-Methoden. (z.B. `setMarke` oder `getMarke`)
- ▶ Erstellt verschiedene Instanzen dieser Auto-Klasse mit Autos verschiedener Automarken, Preise, Modelle und Baujahre. Probiert anschließend die Getter- und Setter-Methoden aus, um die Werte der Attribute abzurufen und zu aktualisieren

Aufgaben 4

- Schreibt eine Methode in der Auto-Klasse, die den Preis des Autos basierend auf einem prozentualen Rabatt aktualisiert
 - ▷ Die Methode soll den neuen Preis zurückgeben.
 - ▷ Testet diese Methode mit einem Beispiel.