Seiteneffekte ist wenn man von außen darauf zugreifen kann

Liste verändern, refferenz zeigt drauf und dadurch verändert man orginal und das verweißte dann das wird geändert - Seiteneffekte

Konstruktor heißt in Python __init__

Mehrfachvererbung ist in p möglich, in java nicht

Modifikatoren in python wie static gibt es nicht wirklich

es gibt instanz und klassen variablen

membervariablen in java und python, wenn wir in klasse außerhalb definieren wird es zu einer membervariable mit einer instanz, erst wenn wir

static dazuschreiben, dann wird es statisch

wenn eine klasse von ner anderen klasse erbt dann kann man in der erbenden klasse super schreiben, damit sie von mutterklasse erbt

was ist super? - eine Refferenz

Lambda ist speicherschonend da kürzer

Map sind wie for-schleifen und können iterable funktionen ausführen

iterable gehen jede stelle durch und ein objekt ist iterable wenn es zb eine liste ist, oder __iter__ und __next__ müssen implementiert werden dann ist es iterable (iter-anfang, next-nächstes)

In Java ist es keine echte Mehrfachvererbung

I muss die eindutigkeit vergeben, wenn man mehrfach vererbt dann muss man aufpassen wenn man eine struktur mit den selbenmethodennamen nimmt - achtung überlagerung (overload)

bei java muss ich super.super.super eingeben was ich in python nicht muss

In der init methode kann ich schauen wie viele argumente ich habe und nach der anz. der Argumente gehen

Wir könnten noch eine separate methode machen oder vererbung

Wir haben mehrere Konstruktoren

Counter für wie oft der Mensch was spielt und des soll der pc nehmen und er soll random

und das auf verschiedenen schwierigkeitsstufen

wie speichere ich daten? Txt datei oder sqlight datenbank

Daten gezählt, flask api im hintergrund der eine db hat und an den wir daten schicken -> 2 separate programme

Wir wollen eine Programmierumgebung die exakt so abgestimmt ist dass unser programm funktioniert

Bibliotheken können sehr sehr viel aber sie verändern sich laufend

Python rohinstallation erweitert man in einem Ordner auf unserer Festplatte, welche nur in einem Bestimmten Ordner sind

Python beim normalen rechner eingeben in cmd dann kommt systempython

mit virtual invironment "activate" kommt python

Wir haben viruatl environment installiert und bibliotheken importiert

-> es gibt zwei befehle pip (python installer) installiert zusätzlich bibliotheken

pip freeze -> |

wenn jemand meine repository klont: pip install -r

pip install pipreqs

bei pip freeze werden alle imports reingeschrieben

wenn wir pipreqs eingeben schaut es nach wer es wirklich verwendet im code und benutzt nur die erforderlichen (eigentlich eigenen librarys)

geht ordnermäßig vor

ist an unsere hardware gekoppelt, sobald wir an unserem neuen rechner sind müssen wir es erneut aufbauen

wir haben projekt, dann hab ich gitinit, dann gitignore python c*jpg oder so kein plan, pipreqs installieren (oder freeze)

---------- neue Stunde

Namespaces geben Scope an, in dem sie definiert sind (Sichtbarkeit)

-> Sammlung von Variablen

Es gibt global, enclosing, Built-in, und local namespace

Enclosing ist wenn in python in der methode eine andere Methode definiert wird (nested Methods)

local -> enclosing -> global -> built-in

enclosed ist die zweite def in einer def (enclosing)

https://realpython.com/python-namespaces-scope/

Wenn wir eine variable nicht finden bekommen wir eine name-error exception

Alles was ich global definiere ist überall Sichtbarkeit

built in, ist des was python schon direkt vordefiniert hat

Error: kann man nicht handlen

Exception: kann man handlen

python verwaltet refferenzen als Dictionaries

->

global()

{add:adfa,aadfa:hs,nrt:rmrr,...}

global vor variabel schreiben für Sichtbarkeit

x,y,z = 1,2,3

nonlocal isch um global zu umgehen

-Best Practices durchlesen zuhause-

---------- neue Stunde

Wieso brauche ich Bibliotheken?

Wiederverwendbarkeit,

Simplicity,

Wartbarkeit (damit ich wenn ich an einer stelle was mache, dass es benötigte auch geändert wird),

Scoping (Gültigkeitsbereich wenn ich etwas importiere, dann entsteht ein 2. global bereich)

Ein Modul kann selber geschrieben werden (ist nur Python Code), wir können in C module schreiben

Es gibt Module die per se schon module sind aber von haus aus in interpreten eingebettet sind

mod.py - man kann variabeln, methoden, klassen importieren

Wie macht man module: in aktuellen ordner reinlegen oder in aktuellen pfad legen ooooder
schmutzig: aktuellen pfad manipulieren (zu dem pfad directory oder so hinzufügen -> man verändert damit struktur -> wird unkenntlich)

mod. ist aufrufverfahren

https://realpython.com/python-modules-packages/

import * importiert alles

dir() zeigt was ich local definiert habe

ein modul kann man nur einmal laden

---------- neue Stunde

n*logn sortiert, gibt sonst noch Bubblesort

Instanzvariablen (Alles was mit Self anfängt (100 Firmen = 100 Instanzen)) und Klassenvariablen ()

----------neues Jahr----------

In Python kann man nicht nur Objekte sondern auch Funktionen übergeben, also Methoden andere Methoden übergebe.

3 Normalformen

Atomar, alle nicht Primärschlüssel Atribute müssen immer von einem (Schlüssel) Atribut abhängig sein, darf nicht transient erfolgen (abhängigkeit darf nicht über einen nichtschlüssel fungieren)...deutsch?

Die drei Normalformen des Programmierens sind die erste Normalform (1NF), die zweite Normalform (2NF) und die dritte Normalform (3NF).

1NF bezieht sich auf die Struktur der Tabellen in einer Datenbank, und sie besagt, dass jede Zelle in einer Tabelle genau einen Wert enthalten sollte.

2NF besagt, dass jede Nicht-Primärschlüsselfehlvariable in einer Tabelle vollständig von der Primärschlüsselfehlvariable abhängig ist.

3NF besagt, dass es keine transitive Abhängigkeiten in einer Tabelle gibt,d.h. dass jede Nicht-Primärschlüsselfehlvariable in einer Tabelle nicht direkt oder indirekt von einer anderen Nicht-Primärschlüsselfehlvariable abhängig ist.

0.3333333 weil Gleitkommazahl (float) ja in Binär gespeichert wird und 1/3 schwer darzustellen ist und speicher fressen würde

Dateipfad verbessern, verkürzen, auf anderen Systemen zum Laufen bringe; generell von Home ausgehen

spieler zug in global

In Python können Funktionen Argumente auf zwei Arten entgegennehmen:

durch Positionsargumente und Schlüsselwortargumente (auch als "kwargs" bezeichnet).

Positionsargumente werden anhand ihrer Position übergeben.

Diese Argumente müssen in der gleichen Reihenfolge übergeben werden,

wie sie in der Funktionsdefinition angegeben sind. Hier ist ein Beispiel:

```
def print_info(name, age):
    print(f"Name: {name}, Alter: {age}")

print_info("Alice", 25)  # Name: Alice, Alter: 25
```

In diesem Beispiel wird die Funktion print_info mit zwei Positionsargumenten aufgerufen:

"Alice" und 25. Da die Argumente in der gleichen Reihenfolge übergeben werden,

wie sie in der Funktionsdefinition angegeben sind, werden sie der Variablen name und age zugewiesen.

Schlüsselwortargumente (kwargs) werden anhand ihres Schlüssels übergeben.

Diese Argumente können in einer beliebigen Reihenfolge übergeben werden und müssen mit dem Schlüssel angegeben werden.

Hier ist ein Beispiel:

```
def print_info(name, age):
    print(f"Name: {name}, Alter: {age}")

print_info(age=25, name="Alice")  # Name: Alice, Alter: 25
```

In diesem Beispiel wird die Funktion print_info immer noch mit zwei Argumenten aufgerufen,

aber diesmal werden sie als Schlüsselwortargumente übergeben.

Da die Argumente mit ihrem Schlüssel angegeben werden,

können sie in einer beliebigen Reihenfolge übergeben werden und werden trotzdem der Variablen name und age zugewiesen.

Es gibt auch eine Möglichkeit, eine Funktion so zu schreiben,

dass sie sowohl Positions- als auch Schlüsselwortargumente entgegennehmen kann.

Dies kann durch die Verwendung von *args und **kwargs in der Funktionsdefinition erreicht werden.

```
def print_info(*args, **kwargs):
    print(f"Positional args: {args}")
    print(f"Keyword args: {kwargs}")

print_info("Alice", 25, "Bob", 30, name="Charlie", age=35)
```

In diesem Beispiel entgegnet die print_info Funktion sowohl Positions als auch Schlüsselwortargumente.

Die Positionsargumente werden in einem Tuple args gesammelt und

https://www.data-science-architect.de/args-kwargs/

---------- neue Stunde

Decorator wid erst dann aufgerufen wenn

Wenn wir eine Methode haben die zweimal aufgerufen werden soll

Do twice decorator

https://realpython.com/primer-on-python-decorators/

functools dekoriert unsere wrapper (@functools.wraps(func)) speicher eine funktion ab und übergibt wieder

zurück wenn wir mit unserem wrapper do twice methode und verlieren somit den namen der im beispiel übergeben

wird

```
import functools
def do_twice(func):
    @functools.wraps(func)
    def wrapper_do_twice(*args, **kwargs):
```

proxypattern nachlesen sonst greini froga

timing functions

functools kopiert dunder und fügt es danach hinzu

schreibt diese names nochmals im code (im stack) hinzu nach der kopie

decorators und functools ganz sicher im test!

---------- neue Stunde

Was kommt test

tutorials teacher python test

namespaces theorie;

args, kwargs;

(bubble usw) sort nit genau so aber es gibt 2 gruppen von sortieralgorithmen

Aufwandsklasse: n^2 (der schnellste)

(in search in sort (insertieren -> 5 karten in hand, nehme eine karte suche position (zwischen 4 und 8)), er kann wenn zahlen schon in reihenfolge sind hat er bessere Aufwandsklasse (n im besten fall) (welche Aufwandsklasse ein Programm hat sieht man nur am Code) am effektivsten unter 43)

(Selectionsort suche kleinstes element aus zb 10, schiebs vor und hab dann nur noch 8, dann 7,...)

(Bubblesort)

und Aufwandsklasse: n*logn

Äußere schleife ist rekursion, innere schleife bleibt normal

quicksort macht datenstruktur mit pfeilen, pfeile treffen sich dann in der mitte und teilt das array in 2 teile auf (in teilbereiche rekursive varianten) (höhe eines Baumen ist logn));

Es gibt Stabilität und inplace

inplace bedeutet ob sortierverfahren oder algorithmus innerhalb seiner datenstruktur arbeitet--> hintergrund: ich habe 1 mio. speicher und datenstruktur ist 1 mio. groß -> tendenziell immer besser inplace algorithmus zu verwenden

!!!algorith. ist dann inplace wenn es keinen von n abhängigen zusatzspeicher benötigt UND dieser zusätzliche speicher muss dann konstant sein!!!

quicksort arbeitet innerhalb datenstruktur braucht aber zusätzlich eigenen speicher auf Stack

dann gibts stabilität, zb wir haben eine datenstruktur und in der datenstruktur haben wir gleiche elemente (a paar 7ner) -> 7a,7b,7c -> dann sortieren aber merken 7ner sind durcheinander geraten (also nicht stabil)

Hauptkriterium ist aufwandsklasse, dann inplace/stabilität

Pythonaufgabe zur Matura, Jenewein schafft in unter 1h

decorators; random?;

einfach verkettete liste wir haben headzeiger der zeigt auf 0 und dann gibt es zeiger der auf den nextes objekt, dann next auf None

wollen wir 2. objekt hinzufügen

bei ner einfachen muss man alle elemente durchgehen um hinten was hinzuzufügen

doppelt verkettet liste kann man von hinten und von vorne die elemente einarbeiten

wenn etwas konstante aufwandsklasse hat dann Ovon1 (nicht 0)

in nem objekt sind gewisse methoden bereits implementiert man kann mit super irgendwas

repr macht ein objekt mit dem man es sofort anlegen kann

python legt typen dynamisch fest