



Teil 7: Einführung in OOP



OOP (Objektorientierte Programmierung mit Python)

Einführung in Python

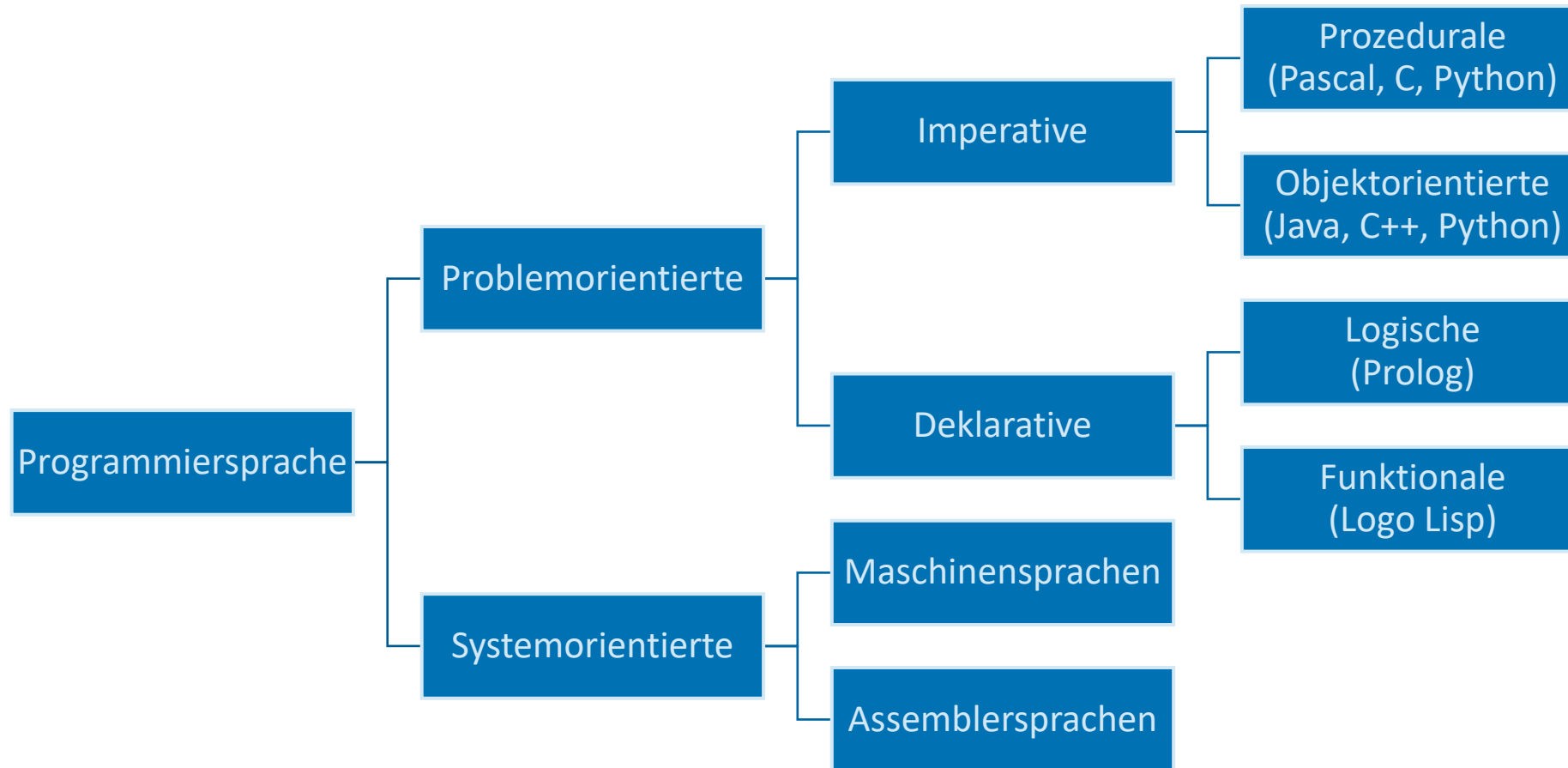
Lernziele:

- Was ist OOP?
- Programmierparadigmen
- Die Idee von OOP
- Grundbegriffe in OOP
- OOP in Python

Was ist OOP?

- Eine Art der Herangehensweise in der Programmierung, ein Programmierparadigma(grundsätzliche Denkweise)
- OOP als Programmierparadigma ist in einigen Programmiersprachen zwingend vorgeschrieben, in Anderen nicht.
- In Python ist objektorientierte Programmierung nicht vorgeschrieben aber möglich.
- Alternative Programmierungsstile in Python sind deklarative Programmierung und prozedurale Programmierung.
- Aus dem Grund wird Python auch als Multiparadigmensprache bezeichnet.

Programmierparadigmen



Programmierparadigmen

- Imperative Programmierung

Die imperative Programmierung ist der klassische Ansatz der Programmierung. Ein Programm wird nach einer Folge von Anweisungen der Reihenfolge nach abgearbeitet (Beispiel: Basic etc.)

- Deklarative Programmierung

Die deklarative Programmierung ist ein Programmierparadigma, bei der Beschreibung des Problems im Vordergrund steht. Der Lösungsweg wird dann automatisch entwickelt.

- Prozedurale Programmierung

Die Bezeichnung ist (laut Wikipedia) nicht eindeutig beschrieben. Sie gilt aber als Erweiterung des Imperativen Paradigmas, um den Ansatz Algorithmen in überschaubare Teile zu zerlegen, die anhand definierter Schnittstellen aufrufbar sind.

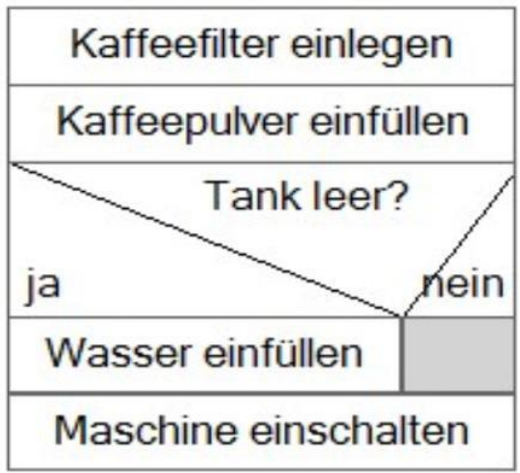
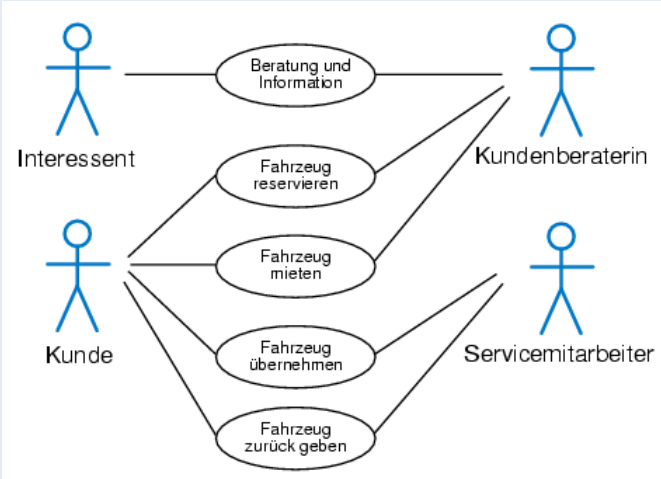
Programmierparadigmen

Zerlegung in Teilprobleme

- Die imperative Programmierung ist der klassische Ansatz der Programmierung. Ein Programm wird nach einer Folge von Anweisungen der Reihenfolge nach abgearbeitet (Beispiel: Basic etc.)
- Diese Teilprobleme werden Funktion, Unterprogramm, Routine oder Prozedur genannt, und können von verschiedenen Stellen beliebig aufgerufen werden.
- Dieses Programmier-Paradigma wird auch in Python viel genutzt, vor allem in Skripten. Aber auch Module für Python sind nicht automatisch und immer objektorientiert programmiert.

Prozedurale Programmierung vs. OOP

Vergleich von prozeduraler Denkweise gegenüber objektorientierte Sichtweise

Prozedural	OOP
<p>Das Problem wird analysiert und in Einzelschritte zerlegt.</p> <p>Das Programm läuft die Einzelschritte durch bis zum gewünschten Endresultat.</p>	<p>Das Problem wird danach analysiert, wer die Akteure sind, was diese für Aktionen ausführen, und was die Daten sind.</p> <p>Aus Akteuren werden Objekte, denen Eigenschaften zugeschrieben werden und die mit Hilfe von Methoden Aktionen ausführen und somit Daten erzeugen.</p>
	

Die Idee von OOP

- Die Grundidee besteht darin, die Architektur einer Software an den Grundstrukturen desjenigen Bereichs der Wirklichkeit auszurichten, der die gegebene Anwendung betrifft.
- Ein Modell dieser Strukturen wird in der Entwurfsphase aufgestellt. Es enthält Informationen über die auftretenden Objekte und deren Abstraktionen, ihre Typen.
- Die Umsetzung dieser Denkweise erfordert die Einführung verschiedener Konzepte:
 - wie Klassen,
 - Vererbung und
 - Polymorphie.

Die Idee von OOP

Was erhofft man sich von OOP?

- bessere Wiederverwertbarkeit von Code
 - was wiederum zu kürzerem Code führt
 - und auch zu besser lesbarem Code führen kann
 - kürzere Entwicklungszeit
 - eine Senkung der Fehlerrate
- verbesserte Erweiterbarkeit, Anpassbarkeit

Die Idee von OOP

Die Umsetzung von OOP:

- In der objektorientierten Programmierung werden Daten und deren Funktionen (Methoden), - d.h. Funktionen, die auf diese Daten angewendet werden können - in einem Objekt zusammenzufassen und nach außen gekapselt, so dass Methoden fremder Objekte diese Daten nicht direkt manipulieren können.
- Objekte wiederum werden in Klassen definiert.
Die Objekte werden auch Instanzen genannt.

Grundbegriffe in OOP

Klassen:

- Eine Klasse ist eine formale Beschreibung über die Beschaffenheit eines Objektes. Diese beinhaltet die Eigenschaften (Attribute) des Objektes. Außerdem wird das Verhalten des Objektes mit Hilfe von Methoden der Klasse bestimmt.
- Es geht also um 2 Dinge:
 - was können wir über das Objekt sagen, was wissen wir über das Objekt?
 - was können wir mit dem Objekt in einem Programm tun bzw. was kann es tun?

Schließlich werden mit Hilfe dieses Bauplans Objekte erzeugt.

Grundbegriffe in OOP

Vererbung:

- Vererbung bedeutet im Grunde, dass Entwickler Unterklassen definieren können, die alle Eigenschaften der übergeordneten Klasse besitzen.
- Dieser Teil der OOP wurde in der objektorientierten Programmierung erst 1976 eingeführt, war also nicht Teil der ursprünglichen Idee.

Links zu Python und OOP

- <https://docs.python.org/3/tutorial/classes.html>
- https://www.w3schools.com/python/python_classes.asp
- <https://docs.microsoft.com/en-us/learn/modules/python-object-oriented-programming/>
- https://openbook.rheinwerk-verlag.de/python/21_001.html#u21
- <https://www.python-kurs.eu/klassen.php>

VIELEN DANK!

