

Woche 4: Grundlagen der OOP

Herzlich willkommen zur vierten Woche Ihres Python-Selbststudiums! Sie haben bereits viel über Python gelernt, von den Grundlagen und Datenstrukturen bis hin zu Funktionen und Modulen. In dieser Woche werden wir einen großen Schritt machen und in die Welt der objektorientierten Programmierung (OOP) eintauchen.

Objektorientierte Programmierung ist ein Programmierparadigma, das auf dem Konzept von "Objekten" basiert, die Daten und Methoden enthalten. Sie werden lernen, wie man Klassen definiert, Objekte erstellt und wie diese Objekte mit Datenstrukturen interagieren. Sie werden auch erfahren, wie man Methoden erstellt und auf Attribute zugreift.

Ihre Aufgaben für diese Woche bestehen darin, einfache Klassen zu entwerfen und zu implementieren und Methoden zu entwerfen. Sie werden die OOP-Konzepte, die Sie im Selbststudium gelernt haben, in die Praxis umsetzen und dabei Ihre Fähigkeiten in der objektorientierten Programmierung vertiefen.

Am vierten und fünften Tag werden wir die Konzepte der Woche wiederholen und vertiefen. Wir werden uns auf den Klassenentwurf und den Einsatz von Objekten konzentrieren. Sie werden auch die Gelegenheit haben, ein bestehendes Programm in eine OOP-Version umzuwandeln.

Zusätzlich dazu werden wir einige Designmuster und bewährte Verfahren in der objektorientierten Programmierung erforschen. Sie werden auch die Möglichkeit haben, an einem Code-Review teilzunehmen, einer wichtigen Praxis in der professionellen Softwareentwicklung.

Am Ende dieser Woche werfen wir einen Blick auf erweiterte OOP-Techniken, die Sie in den kommenden Wochen erkunden werden.

Diese Woche wird eine spannende und herausfordernde sein, die Ihr Verständnis von Python erweitert und Ihnen neue Werkzeuge an die Hand gibt, um effektive und gut organisierte Programme zu erstellen. Lassen Sie uns diese Reise fortsetzen und Ihre Python-Kenntnisse weiter ausbauen!

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
 - Konzepte der OOP
 - Klassendefinition
 - Objektanlage
 - Objekte und Datenstrukturen
 - Methodenanlage
 - Attribute und der Zugriff darauf
- Aufgabe:
 - Einfache Klassen entwerfen
 - Klassen implementieren
 - Methoden entwerfen
- Tag 4 und Tag 5:
 - Wiederholung
 - Vertiefung: Klassenentwurf
 - Vertiefung: Objekteinsatz
 - Vertiefung: Umwandlung eines Programms in eine OOP-Version
 - Ergänzung: Designmuster und bewährte Verfahren
 - Ergänzung: Code-Review
 - Ausblick: erweiterte OOP-Techniken

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein.

Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

- Grundkonzepte der objektorientierten Programmierung
 - Verständnis von Klassen und Objekten
 - Die Bedeutung der Datenkapselung
 - Vererbung und Polymorphie
 - Abstraktion und Interfaces
- Klassendefinition und Objektanlage
 - Erstellen von Klassen
 - Instanzieren von Objekten
 - Zugriff auf Attribute und Methoden
 - Verwenden von Konstruktoren und Destruktoren
- Objekte und Datenstrukturen
 - Speichern von Objekten in Listen, Tupeln, Mengen und Wörterbüchern
 - Zugriff auf Objekte in Datenstrukturen
 - Methoden, die auf Datenstrukturen arbeiten
- Methodenanlage
 - Definition von Methoden innerhalb von Klassen
 - Verstehen der self-Variable
 - Zugriff auf Klassen- und Instanzvariablen in Methoden
- Attribute und der Zugriff darauf
 - Verstehen der Unterschiede zwischen Klassen- und Instanzvariablen
 - Erstellen und Verwenden von Klassen- und Instanzmethoden
 - Private und öffentliche Attribute
 - Verwenden von Eigenschaften (getter und setter Methoden)

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Vorbereitung
 - Empfohlene Lernmaterialien bereitstellen (Bücher, Online-Kurse, Videos)
 - Auffrischen der bisherigen Python-Kenntnisse und der Grundlagen der Funktionen und Module
2. Grundlagen der objektorientierten Programmierung
 - Verstehen, was Klassen und Objekte sind und warum sie verwendet werden
 - Lernen, wie man eine Klasse definiert und ein Objekt erstellt
 - Verstehen des Unterschieds zwischen Klassenattributen und Instanzattributen
 - Arbeiten mit Methoden in Klassen und Verstehen der "self"-Notation
 - Lernen, wie man auf Attribute und Methoden von Objekten zugreift
3. Arbeiten mit Klassen und Objekten
 - Erstellen von Klassen, die Geschäftslogiken repräsentieren
 - Anwendung von Objekten und Methoden zur Lösung realer Probleme
 - Lernen, wie man Attribute ändert und Methoden von Objekten aufruft
4. Datenkapselung und Information Hiding
 - Einführung in das Konzept der Datenkapselung und des Information Hiding
 - Lernen, wie man mit privaten und öffentlichen Attributen und Methoden arbeitet
 - Verstehen der Bedeutung von getter- und setter-Methoden
5. Übungsaufgaben
 - Entwerfen und Implementieren von Klassen basierend auf spezifischen Anforderungen
 - Erstellen und Manipulieren von Objekten zur Problemlösung
6. Selbstbewertung
 - Multiple-Choice-Bewertung absolvieren
 - Wichtige Konzepte und Fähigkeiten überprüfen
 - Schwierige Themen gegebenenfalls erneut durcharbeiten
7. Online-Schulungstag

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. **Bankkonto-Verwaltung:** Erstellen Sie eine Klasse BankAccount, die ein Bankkonto repräsentiert. Es sollte Attribute für den Kontostand und den Kontoinhaber haben und Methoden zum Einzahlen, Abheben und Anzeigen des aktuellen Kontostands.
2. **Bücherei-System:** Erstellen Sie eine Klasse Book, die ein Buch in einer Bibliothek repräsentiert. Es sollte Attribute für den Titel, den Autor, das Genre und den Ausleihstatus haben. Erstellen Sie eine weitere Klasse Library, die eine Sammlung von Büchern repräsentiert. Es sollte Methoden zum Hinzufügen von Büchern, zum Ausleihen von Büchern und zur Anzeige aller verfügbaren Bücher haben.
3. **Geometrische Formen:** Erstellen Sie eine Basisklasse Shape mit Methoden zur Berechnung des Umfangs und der Fläche. Leiten Sie daraus Klassen wie Circle, Rectangle und Triangle ab. Implementieren Sie die Methoden zur Berechnung des Umfangs und der Fläche in den abgeleiteten Klassen entsprechend.
4. **Einfaches Studentenverwaltungssystem:** Erstellen Sie eine Klasse Student mit Attributen für Namen, Alter und Kurs. Erstellen Sie eine Klasse School die eine Liste von Studenten verwaltet. Es sollte Methoden zum Hinzufügen und Entfernen von Studenten, zum Abrufen von Informationen über einen bestimmten Studenten und zum Anzeigen aller Studenten in einem bestimmten Kurs haben.
5. **Einfaches Inventarsystem:** Erstellen Sie eine Klasse Item, die ein Inventarelement mit Attributen wie Name, Menge und Preis repräsentiert. Erstellen Sie eine weitere Klasse Inventory, die eine Sammlung von Artikeln verwaltet. Sie sollte Methoden zum Hinzufügen und Entfernen von Artikeln, zur Aktualisierung der Menge eines Artikels und zur Anzeige des Gesamtwerts des Inventars haben.

Überlegen Sie sich bei den Aufgaben 2 bis 5, wie Sie die Daten speichern und wieder laden können. Hierzu können Sie Textdateien, CSV, JSON o.ä. nutzen. Oder vielleicht finden Sie im Internet noch weitere, bessere Möglichkeiten?

Abschluss-Quiz

Das Quiz soll Ihnen einen ersten Hinweis auf Ihren Lernfortschritt geben. Nach unserer Einschätzung sollten Sie diese Fragen alle beantworten können, wenn Sie den Stoff der Woche durchgearbeitet und verstanden haben. Natürlich gibt es noch sehr viel mehr mögliche Fragen, dazu wollen wir auf die Literatur und das Internet verweisen. Geben Sie gerne einmal „python quizzes“ bei Google ein.

1. Wie erstellt man eine Klasse in Python?
 1. `class MyClass: pass`
 2. `MyClass = class():`
 3. `class = MyClass()`
 4. `MyClass = class: pass`
2. Wie initialisiert man eine Klasse in Python?
 1. `def __init__(self): pass`
 2. `def __start__(self): pass`
 3. `def __class__(self): pass`
 4. `def __begin__(self): pass`
3. Wie greift man auf eine Klassenmethode zu?
 1. `object.method()`
 2. `class.method(object)`
 3. `object->method()`
 4. `method.object()`
4. Wie definiert man eine Klassenmethode in Python?
 1. `def method(self): pass`
 2. `method(self): pass`
 3. `self.method(): pass`
 4. `method = def(self): pass`
5. Wie erzeugt man eine Instanz einer Klasse?
 1. `object = MyClass()`
 2. `object = MyClass`
 3. `object -> MyClass()`
 4. `MyClass = object()`
6. Wie greift man auf ein Klassenattribut zu?

1. `object.attribute`
 2. `attribute.object`
 3. `class.attribute(object)`
 4. `object->attribute`
7. Wie definiert man ein Klassenattribut in Python?
1. `self.attribute = value`
 2. `attribute = self.value`
 3. `attribute(self) = value`
 4. `self = attribute.value`
8. Was ist der Zweck der self-Variable in Python-Klassen?
1. Sie verweist auf die Instanz der Klasse
 2. Sie verweist auf die Klasse selbst
 3. Sie verweist auf die Methode, die gerade aufgerufen wird
 4. Sie verweist auf die Elternklasse
9. Was ist eine Instanzmethode in Python?
1. Eine Methode, die auf die Klasse selbst und nicht auf eine Instanz der Klasse angewendet wird
 2. Eine Methode, die auf eine Instanz der Klasse und nicht auf die Klasse selbst angewendet wird
 3. Eine Methode, die keine Argumente akzeptiert
 4. Eine Methode, die nur statische Variablen verwendet
10. Was macht der `__init__`-Methodenaufruf in Python?
1. Er wird aufgerufen, wenn eine neue Instanz der Klasse erstellt wird
 2. Er wird aufgerufen, wenn eine Klasse gelöscht wird
 3. Er wird aufgerufen, wenn eine Methode der Klasse aufgerufen wird
 4. Er wird aufgerufen, wenn eine Klasse importiert wird
11. Wie kann man in Python eine Methode einer übergeordneten Klasse aufrufen?
1. `super().method()`
 2. `parent.method()`
 3. `method.super()`
 4. `method.parent()`
12. Wie kann man eine Methode einer Klasse außerhalb der Klasse aufrufen?

1. `object.method()`
2. `class.method()`
3. `method.object()`
4. `method.class()`

Antworten:

1/1 2/1 3/1 4/1 5/1 6/1 7/1 8/1 9/2 10/1 11/1 12/1

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und erstanden worden sein.

- Buch: Python Crash Course:
 - **Klassen:** Dieses Thema wird in "Kapitel 9: Klassen" behandelt.
- Video: CodeAcademy
 - Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer
- Lab:
 - Lab: Python for Developers – Grundlegendes Python inklusive Datenstrukturen

Anhänge

Aktuell eine Leerseite