

3. Python Mixin (Mehrfachvererbung)

Was ist ein Mixin in Python?

Ein Mixin

- ist eine Klasse, welche Methodenimplementierungen zur Wiederverwendung durch mehrere verwandte, untergeordnete Klassen bereitstellt. Die Vererbung impliziert jedoch keine Ist-ein-Beziehung.
- definiert keinen neuen Typen. Daher ist es nicht zur direkten Instanziierung vorgesehen.
- bündelt eine Reihe von Methoden zur Wiederverwendung. Jedes Mixin sollte ein einziges spezifisches Verhalten haben und eng verwandte Methoden implementieren.

Python Mixin Beispiel

Zunächst wird die Klasse Person implementiert:

```
class Person:
    def __init__(self, name):
        self.name = name
```

Dann wird eine Klasse Employee definiert, die von Person erbt:

```
class Employee(Person):
    def __init__(self, name, skills, dependents):
        super().__init__(name)
        self.skills = skills
        self.dependents = dependents
```

Zuletzt wird eine neue Instanz von Employee erzeugt:

```
if __name__ == '__main__':
    e = Employee(
        name='John',
        skills=['Python Programming', 'Project Management'],
        dependents={'wife': 'Jane', 'children': ['Alice', 'Bob']}
    )
```

Mehrere Mixin Klassen kombinieren

Angenommen, Sie möchten das Employee Objekt zu JSON konvertieren. Dazu können Sie zunächst eine neue Mixin-Klasse definieren, die das Standardmodul json verwendet:

```
import json

class JsonMixin:
    def to_json(self):
```

```
return json.dumps(self.__dict__)
```

Ergänzen Sie dann die Klasse Employee so, dass sie zusätzlich von JsonMixin erbt, wodurch die Methode to_json auf Employee verfügbar wird:

```
class Employee(JsonMixin, Person):  
    def __init__(self, name, skills, dependents):  
        super().__init__(name)  
        self.skills = skills  
        self.dependents = dependents
```

Im Folgenden wird eine neue Instanz der Klasse Employee erstellt und nach JSON konvertiert:

```
if __name__ == '__main__':  
    e = Employee(  
        name='John',  
        skills=['Python Programming', 'Project Management'],  
        dependents={'wife': 'Jane', 'children': ['Alice', 'Bob']}  
    )  
  
    print(e.to_json())
```



© Daniel Fahrni **Quellen:**

- [Python mixin | pythontutorial.net](https://pythontutorial.net/python-mixin/)
- [Python Up Your Code: Mixins | medium.com](https://medium.com/@daniel.fahrni/python-up-your-code-mixins-1234567890)

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m320/learningunits/lu99/theorie/lu8-kapitel_4

Last update: **2024/03/28 14:07**

