

Lernnachweis zur Kompetenz B2G

Umgang mit Funktionen als Objekten

Einleitung

Die Kompetenz B2G bezieht sich darauf, Funktionen als Objekte zu behandeln, sie in Variablen zu speichern und weiterzugeben. Dieser Ansatz eröffnet vielfältige Möglichkeiten in der Programmierung, da Funktionen dadurch dynamisch manipuliert und flexibel eingesetzt werden können. In dieser Dokumentation werden wir den Umgang mit Funktionen als Objekten anhand von Python-Beispielen erklären.

Warum ist der Umgang mit Funktionen als Objekten wichtig?

Die Behandlung von Funktionen als Objekte ermöglicht eine höhere Abstraktion und Flexibilität im Code. Funktionen können nun in Variablen gespeichert, als Argumente an andere Funktionen übergeben und aus Funktionen heraus zurückgegeben werden. Dies erleichtert das Schreiben von allgemeinen und wiederverwendbaren Funktionen.

Beispiel: Funktionen in Variablen speichern und weitergeben

```
def square(x):
    return x ** 2

def cube(x):
    return x ** 3

# Funktionen in Variablen speichern
power_function = square
print(power_function(5)) # Ausgabe: 25

power_function = cube
print(power_function(5)) # Ausgabe: 125

# Funktionen als Argumente übergeben
def apply_operation(func, y):
    return func(y)

result = apply_operation(square, 4)
print(result) # Ausgabe: 16

result = apply_operation(cube, 4)
print(result) # Ausgabe: 64
```

In diesem Beispiel werden die Funktionen «square» und «cube» in Variablen gespeichert und als Argumente an die Funktion «apply_operation» übergeben. Dies ermöglicht die dynamische Anwendung verschiedener Funktionen.

Beispiel: Funktionen als Rückgabewerte

```
def get_power_function(exponent):  
    if exponent == 2:  
        return square  
    elif exponent == 3:  
        return cube  
    else:  
        return None  
  
power_func = get_power_function(2)  
print(power_func(4)) # Ausgabe: 16  
  
power_func = get_power_function(3)  
print(power_func(4)) # Ausgabe: 64
```

Hier wird die Funktion «get_power_function» verwendet, um je nach übergebenem Exponenten die entsprechende Potenzierungsfunktion zurückzugeben. Dies ermöglicht eine flexible Auswahl von Funktionen basierend auf Parametern.

Fazit

Der Umgang mit Funktionen als Objekten eröffnet neue Wege für flexible und dynamische Programmierung. Die Möglichkeit, Funktionen in Variablen zu speichern und weiterzugeben sowie sie als Rückgabewerte zu verwenden, ermöglicht eine höhere Abstraktionsebene und fördert die Wiederverwendbarkeit von Code. Die in dieser Dokumentation gezeigten Beispiele verdeutlichen die praktische Anwendung dieser Fähigkeit in der Programmierung.