

Lernnachweis B3G

Die Fähigkeit, einfache Lambda-Ausdrücke zu schreiben, die einzelne Operationen durchführen, wurde durch meine vertiefte Auseinandersetzung mit der funktionalen Programmierung gestärkt. Ein herausragendes Beispiel ist die Verwendung von Lambda-Ausdrücken für das Erstellen von Funktionen zur Transformation von Daten.

Codebeispiel:

```
# Lambda-Ausdrücke für einfache Operationen
```

```
square = lambda x: x ** 2
```

```
uppercase = lambda s: s.upper()
```

```
# Anwendung der Lambda-Ausdrücke
```

```
result_square = square(5)
```

```
result_uppercase = uppercase('hello')
```

```
print(f'Quadrat von 5: {result_square}')
```

```
print(f'Uppercase von "hello": {result_uppercase}')
```

Hier werden Lambda-Ausdrücke genutzt, um Funktionen für einfache Operationen wie Quadrieren und Uppercase-Transformation zu erstellen.

Reflexion:

Die Anwendung von Lambda-Ausdrücken bietet eine kompakte Möglichkeit, einfache Funktionen zu definieren, ohne eine separate Funktionserklärung zu verwenden. Die Reflexion über die Verwendung von Lambda-Ausdrücken hat mein Verständnis für die Klarheit und Eleganz funktionaler Programmierung vertieft.

Die Diskussionen in der Entwicklergemeinschaft über die Anwendung von Lambda-Ausdrücken in verschiedenen Kontexten förderten meine Fähigkeit zur präzisen und effektiven Nutzung dieser Ausdrücke. Die Kreativität bei der Integration von Lambda-Ausdrücken in meinen Code eröffnete mir neue Wege zur Lösung spezifischer Probleme.

Zukünftige Schritte:

Um meine Fähigkeiten weiter zu vertiefen, plane ich, an Projekten teilzunehmen, die den Einsatz von Lambda-Ausdrücken erfordern, um effiziente und klare Lösungen zu schaffen. Die kontinuierliche Anwendung und Erweiterung dieser Kompetenz wird meine Fähigkeiten in der funktionalen Programmierung weiter stärken und meine Kreativität bei der Lösung verschiedener Probleme fördern.