LU01b - Strukturierte Programmierung

Definition

Strukturierte oder auch imperative Programmierung ist ein Programmierparadigma, das Anweisungen nutzt, um den Computer zu steuern. Es besteht aus einer Abfolge von Befehlen, die nacheinander ausgeführt werden, um einen bestimmten Zustand zu erreichen oder ein Problem zu lösen. Diese Methode ist ähnlich zu der Art und Weise, wie traditionelle Kochrezepte oder Bauanleitungen strukturiert sind.

Grundprinzipien

Im Mittelpunkt der imperativen Programmierung steht die Veränderung des Zustands eines Systems durch die Anwendung von Anweisungen oder Befehlen. Diese Befehle werden in einer klaren und definierten Reihenfolge ausgeführt.

1. Abfolge: Befehle werden in einer bestimmten Reihenfolge geschrieben und in dieser Reihenfolge ausgeführt.

2. Kontrolle: Der Programmfluss kann mit Schleifen und Verzweigungen wie IF-THEN-ELSE-Strukturen gesteuert werden.

3. Zustandsänderung: Variablen speichern Daten und können im Laufe des Programms verändert werden. Jede Anweisung kann den Zustand des Programms verändern.

Beispiele

Ein einfaches Beispiel für imperatives Programmieren ist ein Stück Code, das eine Liste von Zahlen aufnimmt und die Summe davon berechnet.

```
summe = 0
zahlen = [1, 2, 3, 4, 5]
for zahl in zahlen:
    summe += zahl
print(summe) # Ausgabe: 15
```

Hier wird eine klare Abfolge von Befehlen gegeben, die nacheinander ausgeführt werden, um das gewünschte Ergebnis zu erzielen.

Fokus auf Befehle und Anweisungen

Befehle und Anweisungen

In der imperativen Programmierung sind Befehle und Anweisungen die fundamentalen Bausteine. Ein Befehl ist eine Aufforderung an den Computer, eine bestimmte Operation durchzuführen, während eine Anweisung eine spezifische Aufgabe definiert, die ausgeführt werden soll.

Befehle

Befehle sind direkte Anforderungen an den Computer, um eine bestimmte Aktion auszuführen. Sie können einfache Operationen wie Addition oder Multiplikation sein oder komplexere wie das Lesen oder Schreiben einer Datei.

Anweisungen

Anweisungen sind präzise Instruktionen, die einen bestimmten Prozess oder eine Aufgabe beschreiben. Sie werden durch die Kombination von Befehlen und Kontrollstrukturen wie Schleifen oder Bedingungen gebildet. Anweisungen werden auch Algorithmen genannt.

Unterschied zu anderen Paradigmen

Im Gegensatz zur deklarativen Programmierung, die beschreibt, „was“ getan werden soll, konzentriert sich die imperativ auf das „wie“. Es gibt eine klare Abfolge von Schritten, die gefolgt werden müssen, im Gegensatz zur funktionalen oder logischen Programmierung, die möglicherweise mehr auf Beziehungen zwischen Entitäten fokussiert.

Schlussfolgerung

Die imperativ Programmierung ist eine der am weitesten verbreiteten Methoden und liegt vielen der heutigen beliebtesten Programmiersprachen zugrunde. Der Schwerpunkt auf klaren Befehlen und Anweisungen macht sie sowohl leistungsstark als auch flexibel, erfordert aber auch eine sorgfältige Kontrolle und Verwaltung des Zustands des Programms. Dies kann sowohl ein Vorteil als auch eine Herausforderung sein, je nach spezifischem Anwendungsfall oder Problem, das gelöst werden muss.

M323-LU01, M323-AG1