LU01a - Deklarative vs. Imperative Programmierung

Einführung

Programmierung kann auf verschiedene Weisen verstanden und ausgeführt werden. Die Art und Weise, wie ein Programm geschrieben wird, kann in zwei Hauptkategorien unterteilt werden: Imperative Programmierung und Deklarative Programmierung. Beide haben ihre eigenen Vor- und Nachteile und werden für verschiedene Arten von Problemen verwendet.

Imperative Programmierung

Imperative Programmierung wird oft als die traditionellste Art der Programmierung betrachtet. Sie ist eine Art Anweisungsliste oder Rezept, das dem Computer sagt, was er tun soll.

1. **Schritt-für-Schritt-Ansatz**: Der Programmierer gibt eine genaue Sequenz von Anweisungen, die in einer bestimmten Reihenfolge ausgeführt werden sollen.
2. **Statusverwaltung**: Der Zustand des Programms wird häufig durch Variablen verändert, die Werte speichern und ändern.
3. **Kontrollstrukturen**: Schleifen und bedingte Anweisungen werden verwendet, um die Kontrolle über den Programmablauf zu steuern.
4. **Beispiel**: Ein Sortieralgorithmus wie Bubble Sort, der jeden Schritt im Detail beschreibt.

Deklarative Programmierung

Deklarative Programmierung ist eher eine Art, dem Computer zu sagen, was er erreichen soll, ohne zu beschreiben, wie er es tun soll.

1. **Was, nicht Wie**: Der Schwerpunkt liegt auf dem, was erreicht werden soll, nicht auf dem, wie es erreicht werden soll.
2. **Keine Seiteneffekte**: Die Funktionen haben keine Nebeneffekte, d.h., sie ändern nichts in der Welt außerhalb der Funktion.
3. **Höhere Abstraktion**: Oft wird ein höheres Abstraktionsniveau verwendet, das komplexere Operationen in einfacheren Ausdrücken darstellt.
4. **Beispiel**: Eine SQL-Abfrage, die beschreibt, was man von einer Datenbank möchte, ohne die Schritte zur Erlangung dieser Informationen zu beschreiben.

Vergleich

- **Kontrolle vs. Abstraktion**: Während die imperative Programmierung eine präzise Kontrolle über das „Wie“ der Problemlösung bietet, ermöglicht die deklarative Programmierung eine höhere Abstraktion über das „Was“.

- **Lesbarkeit**: Deklarative Programme sind oft kürzer und leichter zu verstehen, während imperative Programme detaillierter sind.
- **Effizienz**: Imperative Programme können oft effizienter sein, da der Programmierer die genaue Kontrolle hat. Deklarative Programme sind möglicherweise weniger effizient, aber schneller zu schreiben und zu warten.
- **Fehleranfälligkeit**: Der detaillierte Charakter der imperativen Programmierung kann fehleranfälliger sein, da mehr Möglichkeiten für menschliche Fehler bestehen.

Schlussfolgerung

Während die imperative Programmierung den Programmierern die vollständige Kontrolle über den Prozess gibt, bietet die deklarative Programmierung eine elegante und oft intuitivere Methode zur Lösung von Problemen. Die Wahl zwischen diesen Stilen hängt oft von der Art des Problems, den Anforderungen an die Effizienz und die Vorlieben des Programmierers ab. In der modernen Softwareentwicklung werden oft Elemente beider Stile kombiniert, um eine effiziente und wartbare Lösung zu schaffen.

M323-LU01, M323-AG1

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu01/deklarativimperativ?rev=1763383005

Last update: **2025/11/17 13:36**