LU02a - Grundkonzepte der funktionalen Programmierung

Die funktionale Programmierung ist ein Paradigma, das in der Informatik immer mehr an Bedeutung gewinnt. Es basiert auf einer Reihe von Prinzipien, die dazu beitragen, den Code vorhersehbarer, wiederverwendbar und leichter zu testen zu machen. In diesem Themenblock konzentrieren wir uns auf zwei Schlüsselkonzepte der funktionalen Programmierung: Immutable Values und Pure Functions.

Immutable Values: Im Kontext der funktionalen Programmierung sind unveränderliche Werte solche, die nach ihrer Erstellung nicht mehr verändert werden können. Diese Unveränderlichkeit trägt zur Vorhersagbarkeit des Codes bei, da sie die Veränderung von Zuständen eliminiert, die oft die Ursache für Fehler und Schwierigkeiten bei der Fehlerbehebung sind.

Pure Functions: Reine Funktionen sind Funktionen, die immer das gleiche Ergebnis für dieselben Eingabewerte zurückgeben und keine sichtbaren Seiteneffekte haben. Sie interagieren nicht mit externen Zuständen oder Daten und ändern nichts außerhalb ihrer eigenen Umgebung. Dies macht sie extrem vorhersagbar und erleichtert das Testen und die Fehlersuche.

Diese Konzepte sind eng miteinander verbunden, da reine Funktionen oft auf Unveränderlichkeit angewiesen sind, um ihre Reinheit zu gewährleisten. Zusammen bilden sie die Grundlage für einen Programmierstil, der die Klarheit und Zuverlässigkeit des Codes fördert.

In den folgenden Abschnitten werden wir uns diese und andere verwandte Konzepte im Detail ansehen, um ein tieferes Verständnis dafür zu entwickeln, wie die funktionale Programmierung zur Entwicklung robuster und effizienter Software beitragen kann.

M323-LU02, M323-AG1

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu02/grundkonzepte?rev=1763026180

Last update: **2025/11/13 10:29**