

Wikipedia-Artikel zum Thema „Quantity“ als Analysemuster in der Softwareentwicklung

Inhaltsverzeichnis:

1. Definition
2. Beispiele und zugehörige Erläuterungen
3. Bedeutung des Analyseusters „Quantity“ für die softwaretechnische Umsetzung
4. Umsetzung bei monetären Einheiten (Währungen)

Dieser Artikel soll ein Verständnis für die softwaretechnische Behandlung von messbaren Grössen entwickeln, die uns im täglichen Alltag begegnen.

1. Definition

Der Begriff „Quantity“ ist mit dem deutschen Wort „Menge“ gleichgestellt. Er beschreibt die zahlenmäßige Erfassung von bestimmten Dingen in unserer realen Welt. Darunter versteht man einen abgegrenzten messbaren Raum von Objekten eines bestimmten Typs. Dieser Raum wird durch den Namen des Typs dargestellt und durch die Objekte in abzählbare Einheiten gleicher Größe eingeteilt. Diesen Einheiten weist man zur Darstellung der Unterscheidung eindeutige Namen zu. Zur Beschreibung der Menge wird der Einheit ein Zahlenwert vorangestellt. Die sich daraus ergebenden Messwerte setzen sich daher immer aus einem Betrag und einer dazugehörigen Einheit zusammen.

2. Beispiele und zugehörige Erläuterungen

(1)

Wir betrachten zum besseren Verständnis folgenden Aussagesatz:

Die Temperatur beträgt 38 Grad-Celsius.

Diese Aussage beinhaltet 2 Begriffe und einen Zahlenwert. Die Begriffe müssen klar getrennt werden. Der Zahlenwert ist bereits eindeutig. Wir betrachten daher jeden Begriff als eigenen Typen und

lassen dann beide Typen in eine Beziehung treten. Die Beziehung stellt dann den Zusammenhang zwischen den Typen dar und wird durch ein Verb gekennzeichnet. Der Zahlenwert (Betrag) wird dann dem jeweils passenden Typ zugeordnet. In diesem Falle stellt der Begriff „Temperatur“ den Raum dar, in dem sich die Objekte befinden. Die einzelnen Objekte selbst werden durch die Einheit mit dem Namen „Grad-Celsius“ dargestellt, von denen im Beispiel genau 38 gezählt werden können.

(2)

Ein wichtiger Gesichtspunkt bei der Typenzuordnung ist die sinnvolle bzw. korrekte Verwendung der Begriffe zur Vermeidung von fehlerhaften Interpretationen. Vertauschungen von Begriffen müssen daher unbedingt vermieden werden.

Ein anderes Augenmerk sollte auch auf die zwingende Notwendigkeit bei der Verwendung des Analysemodells „Quantity“ gelegt werden. Für jeden Einzelfall ist dieser Aspekt dann getrennt zu betrachten.

Die folgende Aussage soll die Zusammenhänge verdeutlichen:

Das Buch besteht aus 250 Seiten.

Eine Vertauschung der Begriffe „Buch“ und „Seite“ im Satz würde hier keinen Sinn ergeben. In der Softwaretechnik gibt es dann einen Oberbegriff „Buch“ mit einem Attribut „Seitenzahl“. Es ist hierbei nicht erforderlich, die Anzahl der Seiten als „Quantity“ zu programmieren.

3. Bedeutung des Analysemodells **Quantity** für die softwaretechnische Umsetzung

Der technische Einsatz dieses Analysemodells ist natürlich aufgrund der Vorgabe der Meßbarkeit bestimmter Merkmale dem Gebiet der Naturwissenschaften vorbehalten. Zum Zwecke der Standardisierung wurde hierfür ein sogenanntes „Internationales Einheitensystem“ entwickelt, das die Basisgrößen sowie Ihre zugehörigen Basiseinheiten und zugeordnete Kurzzeichen nach internationalem Standard beinhaltet. In der nachfolgenden Liste sind die wichtigsten SI-Basisgrößen zusammengestellt.

Basisgröße	SI-Basiseinheit	Zeichen
Länge	Meter	m
Masse	Kilogramm	kg
Zeit	Sekunde	s
Elektrische Stromstärke	Ampere	A
Thermodynamische Temperatur	Kelvin	K
Stoffmenge	Mol	mol
Lichtstärke	Candela	Cd

4. Umsetzung bei monetären Einheiten (Währungen)

Bestimmte Datentypen werden vom Typen-System der Programmiersprache über den Compiler zur Verfügung gestellt. Solche Datentypen sind beispielsweise Integer, Real, String und Datum. Da der Compiler diese Datentypen bereits versteht, können diese bereits für die Implementierung von Attributen innerhalb einer Klasse verwendet werden.

Andere Datentypen wiederum kennt der Compiler nicht oder sie setzen sich aus mehreren Grundattributen zusammen. Diese Typen implementiert man als abstrakte Datentypen in neuen Klassen und stellt dann über Beziehungen entsprechende Verbindungen zu den Klassen her, die auf diese Datentypen zugreifen müssen.

Bei der Darstellung von Währungen steht man vor der Entscheidung, diese Objekt-Eigenschaft entweder als Attribut in der Objekt-Klasse zu implementieren oder einen neuen Datentypen als eigene Klasse zur Aufnahme der entsprechenden Attribute zu generieren.

Es gibt verschiedene Währungen, denen bestimmte Namen eindeutig zugeordnet sind. Aufgrund dieser Tatsache ist bereits die Notwendigkeit zur Bildung einer eigenen Klasse erkennbar. In der praktischen Umsetzung wird dann eine eigene Klasse für den Grundtyp des Geldes mit einem sinnvollen Klassennamen erstellt (z. B. „Money“ oder „Geld“). An diese Klasse wird dann über eine Assoziation eine Unterklasse angehängt, die dann die verschiedenen Währungseinheiten dem Grundtyp „Money“ eindeutig zuordnet.