Physikalisch-Technische Bundesanstalt, Berlin (Adlershof)

Vorlesung: Wissenschaftliche Softwareentwicklung

2023/24

Dr. habil. Till Biskup

— Glossar zu Lektion 13: „Formatierung“ —

Hinweis: Die nachfolgend genannten Begriffe und Definitionen erheben keinen Anspruch auf formale Korrektheit, sondern dienen lediglich dem besseren Verständnis der in der Vorlesung behandelten Themen und sind im jeweiligen Kontext zu sehen. Mehrfache, voneinander abweichende Definitionen in unterschiedlichen Kontexten sind daher möglich. Englische Begriffe werden zwar nach Möglichkeit übersetzt, erscheinen aber ggf. unter ihrem englischen Namen in der Liste. Verweise untereinander sind durch ↑ gekennzeichnet.

Abstraktion Nach Edsger Dijkstra [1] das einzige mentale Werkzeug, das es erlaubt, eine große Vielzahl von Fällen abzudecken. Zweck der Abstraktion ist es nicht, vage zu sein, sondern im Gegenteil ein neues Bedeutungsniveau zu schaffen, das präzise Beschreibungen erlaubt.

Abstraktionsebene Summe aller ↑Abstraktionen eines bestimmten Abstraktionsgrades. ↑Funktionen (bzw. ↑Methoden) sollten immer nur Anweisungen enthalten, die zur gleichen Abstraktionsebene gehören.

Attribut im Kontext der ↑objektorientierten Programmierung eine Variable, die innerhalb einer ↑Klasse definiert wird. ↑Methoden operieren auf den Attributen einer ↑Klasse bzw. dem daraus erzeugten ↑Objekt.

Coding Conventions Innerhalb der Entwicklergruppe eines Projektes zumindest zu einem gegebenen Zeitpunkt festgelegte Konventionen zu bestimmten Aspekten der Formatierung von Quellcode. Oft als Gestaltungsleitfaden (*style guide*) kodifiziert. Dient der Vereinheitlichung und trägt dadurch wesentlich zur Lesbarkeit bei. Wichtiger als der Inhalt ist die konsequente Befolgung der Konventionen und der möglichst breite Konsens über die Inhalte innerhalb der Entwicklergruppe.

Funktion im Kontext der ↑strukturierten Programmierung eine Liste von Anweisungen, die eine bestimmte Aufgabe erfüllt und der Programmiersprache unter einem festen Namen bekannt ist.

Hierarchieebene lokaler Kontext eines Ausdrucks im Quellcode. Ein Schleifenkörper z.B. befindet sich gegenüber dem ihn umgebenden Quellcode auf einer niedrigeren Hierarchieebene. Eine Hierarchieebene wird oft durch entsprechende Einrückung optisch gekennzeichnet.

horizontale Dichte zusammenhängende Teile eines Ausdrucks werden nicht durch Leerzeichen unterbrochen. Ein Beispiel ist die öffnende Klammer nach einem Funktionsnamen. Vgl. als Gegensatz die ↑horizontale Offenheit.

horizontale Offenheit Sinneinheiten eines Ausdrucks werden voneinander durch Leerzeichen getrennt. Ein Beispiel ist der Zuweisungsoperator (=). Vgl. als Gegensatz die ↑horizontale Dichte.

Idiom Linguistik: Spracheigentümlichkeit; Softwaretechnik: Umsetzung (Implementierung) abstrakter Muster bzw. Lösung einfacher Aufgaben (auf niedrigster ↑Abstraktionsebene) in einer konkreten Programmiersprache

Instanzvariable Variable eines ↑Objektes, die entsprechend in einer ↑Klasse definiert wird und

aus allen ↑Methoden der Klasse heraus erreichbar ist.

Klasse *class*, im Kontext der ↑objektorientierten Programmierung die Blaupause für die Erzeugung eines ↑Objektes; Definition der Daten (↑Attribute) und des zugehörigen Verhaltens (↑Methoden).

Konvention innerhalb einer Gruppe oder einem (lokalen) Kontext getroffene (temporäre) Festlegung. Ziel von Konventionen ist die Vereinheitlichung und damit einhergehend die Befreiung von der Notwendigkeit, jedesmal aufs Neue nachdenken zu müssen, wie z.B. gewisse Prozesse durchgeführt oder Objekte benannt werden sollen. Konventionen sind im Gegensatz zu ↑Standards weniger verbindlich und deutlich flexibler sowie *ad hoc* innerhalb einer Gruppe einführbar.

Listing Programmausdruck, gedruckte Wiedergabe von Quellcode eines Programms

Methode im Kontext der ↑objektorientierten Programmierung eine ↑Funktion, die innerhalb einer ↑Klasse definiert wird und auf den ↑Attributen einer ↑Klasse bzw. dem daraus erzeugten ↑Objekt operiert.

Objekt *object*, im Kontext der ↑objektorientierten Programmierung der grundlegende Baustein eines Programms, bestehend aus den Daten (↑Attribute) und dem zugehörigen Verhalten (↑Methoden).

objektorientierte Programmierung (OOP) ein ↑Programmierparadigma, bei dem Daten (Variablen zugewiesene Werte, als ↑Attribute bezeichnet) und Funktionen (↑Methoden), die auf diesen Daten (Attributen) operieren, eine Einheit bilden. Die in den ↑Attributen gespeicherten Daten lassen sich i.d.R. nur vermittelt durch (öffentlich zugängliche) ↑Methoden der ↑Klasse bzw. des daraus erzeugten ↑Objektes ansprechen. Es gibt eine klare Trennung zwischen öffentlicher ↑Schnittstelle und internen Verarbeitungsroutinen. Wichtige Vertreter objektorientierter Programmiersprachen sind Smalltalk, C++ und Java, aber auch Python.

Paradigma nach Thomas S. Kuhn [2] ein Satz allgemein anerkannter wissenschaftlicher Leistungen, der für eine gewisse Zeit einer Gemeinschaft von Fachleuten maßgebende Probleme und Lösungen liefert

Programmierparadigma ein ↑Paradigma der Art zu programmieren. Wichtige Beispiele sind ↑strukturierte Programmierung, ↑objektorientierte Programmierung und funktionale Programmierung.

Refactoring Verbesserung der Qualität des Quellcodes einer Software ohne Einfluss auf ihr von außen erkennbares Verhalten. Diszipliniertes Vorgehen zum Aufräumen von Quellcode, das die Wahrscheinlichkeit, Fehler einzuführen, minimiert.

Standard von einem oft internationalen und anerkannten Gremium definierte Festlegung. Standards sind im Gegensatz zu ↑Konventionen sehr viel starrer und nicht *ad hoc* von einer Gruppe einführbar.

strukturierte Programmierung ein ↑Programmierparadigma, das die Zahl möglicher Kontrollstrukturen auf nur zwei (↑Iteration, ↑Selektion) beschränkt, insbesondere den goto-Befehl eliminiert (E. Dijkstra, [3]). Idealerweise hat ein Codeblock nur jeweils genau einen Ein- und Ausgang. Nach D. Knuth [4, S. x] der systematische Einsatz von Abstraktion, der es ermöglicht, große Programme aus kleine(re)n Komponenten zusammenzusetzen. Wichtige frühe Vertreter strukturierter Programmiersprachen sind C und Pascal. Die meisten heutigen Programmiersprachen (mit Ausnahme der funktionalen Programmiersprachen) unterstützen die strukturierte Programmierung.

vertikale Dichte zusammenhängende Zeilen im Quellcode werden ohne vertikalen Leerraum angeordnet, um den inhaltlichen Zusammenhang zu betonen. Vgl. als Gegensatz die ↑vertikale Offenheit.

vertikale Offenheit Sinnabschnitte im Quellcode werden durch vertikalen Leerraum voneinander getrennt. Vgl. als Gegensatz die ↑vertikale Dichte.

Zeitungs-Metapher Der Code sollte von einfach nach komplex und von abstrakt nach konkret fortschreiten. Übliche Aspekte eines Zeitungsartikels sind eine Überschrift, die möglichst prägnant das Thema vorstellt, eine kurze Zusammenfassung, und insgesamt ein eher kurzer und fokussierter Schreibstil.

Literatur

[1] Edsger W. Dijkstra. The humble programmer. *Communications of the ACM* 15 (1972), S. 859–865.

[2] Thomas S. Kuhn. *Die Struktur wissenschaftlicher Revolutionen*. Frankfurt am Main: Suhrkamp, 1976.

[3] Edsger W. Dijkstra. Go to statement considered harmful. *Communications of the ACM* 11 (1968), S. 147–148.

[4] Donald E. Knuth. *Literate Programming*. Stanford: Center for the Study of Language and Information, 1992.