Ein Überblick über Programmierparadigmen

Gabasch Tamara

letzte Aktualisierung: 3. Juni 2015

Inhaltsverzeichnis

1 Einleitung

In den folgenden Seiten möchte ich Ihnen einen kurzen Überblick über die Vier wichtigsten Programmierparadigmen geben. Das bedeutet, dass die Paradigmen nicht detailliert erklärt werden, sondern nur kurze Beschreibungen der Funktionsweise gegeben werden.

Anfangs wird der Begriff „Paradigma“ erklärt, zweites die „Imperative Programmierung“. Als drittes wird auf die „Funktionale Programmierung“ eingegangen und viertes auf das „Logische Programmierung“. Als letztes Paradigma wird die „Objektorientierte Programmierung“ erklärt. Für jeweils genauere Beschreibungen finden sich in den einzelnen Abschnitten Hinweise.

2 Was ist ein Paradigma

„Ein Paradigma ist ein symbolisches Modell oder Diagramm, das es uns erleichtert, die wesentlichen Merkmale eines Prozesses zu verstehen“[1]

Ein Paradigma ist einfach gesagt ein Konzept oder Muster, dass der Programmiersprache verschiedene Prinzipien zugrunde liegt an die sie sich halten soll.

3 Imperative Programmierung

Wie der Name schon sagt, handelt es sich dabei um eine Sprache die aus einer Reihe von Befehlen besteht. Sie ähnelt stark den Maschinensprachen bzw. Assemblersprachen und scheint dadurch für den Menschen oft etwas schwerer zu verstehen.[2]

Erster BEFEHL und dann nächster BEFEHL

Obiger Satz beschreibt kurz wie die Imperative Programmierung aufgebaut ist. Ein Programm muss den ersten Befehl erledigen bevor es den zweiten Befehl erledigen kann und soll. In einen imperativen Programm ist somit die Ausführungsreihenfolge durch Kontrollstrukturen festgelegt und muss auch eingehalten werden.

Für mehr Einblick in dieses Paradigma, empfiehlt sich der Besuch der Vorlesung „Einführung in die Programmierung“ von Univ.-Prof. PhD. Justus Piater.

Wichtigsten Sprachen: Fortran, Pascal, C

4 Funktionale Programmierung

Im Gegensatz zur Imperativen Programmierung ist die funktionale Programmierung einfacher gestaltet. Sie hat ihren Ursprung in der Mathematik, genau gesagt aus dem Lambda-Kalkül, und besteht nur aus Funktionen. Dieses Paradigma wurde Entwickelt mit der Absicht alle möglichen mathematischen Probleme darstellen zu können, was auch der Grund ist, dass viele Mathematiker es Benutzen. [3]

Das Lambda-Kalkül ist eine mathematische Schreibweise für Funktionen, mit dem wir Beweisen können ob ein Programm korrekt ist, denn jedes geschriebenen Programm kann in Lambda-Kalkül umgewandelt werden.

Mehr Informationen dazu bekommen Sie in der Vorlesung „Funktionale Programmierung“ von Dr. Cezary Kaliszyk

Wichtigste Sprachen: LISP, OCAML, Haskell

5 Logische Programmierung

Die Logische Programmierung unterscheidet sich wohl am Meisten von den anderen Drei. Es versucht aus gegebenen Fakten und Regeln die Lösung zu finden. Der Benutzer muss sich dabei nicht um den Lösungsweg kümmern, denn die Lösung wird einfach aus den gegebenen Wissen berechnet. Durch das gegebene Wissen, können auch mehrere Aussagen gelten. Sollte dies der Fall sein, dann versucht die Maschine durch Ausprobieren alles Möglichkeiten (Backtracking) eine Lösung zu finden.[4]

Wer genaueres zur Logischen Programmierung wissen möchte, kann die Vorlesung „Logische Programmierung“ von Assoz. Prof. Dr. Georg Moser besuchen.

Wichtigste Sprache: Prolog

6 Objektorientierte Programmierung

Die Objektorientierte Programmierung ist das wohl bekannteste und beliebteste Paradigma der letzten paar Jahrzehnten. Es wird versucht den Ausgangspunkt in der Betrachtung der Realität sowie der menschlichen Art und Weise zu denken und zu sprechen zu nachempfinden und wirkt dadurch für den Menschen verständlicher.[5]

Im Gegensatz zu den vorherigen Paradigmen teilt hier der Programmierer den Code in kleinere Teile, Klassen genannt auf, die miteinander kommunizieren. Diese Aufteilung soll es den Programmierer einfacher machen nicht den Überblick zu verlieren, wenn die Programme sehr groß werden.

Damit ein Programm als Objektorientiert gilt, sollte es folgende Beschreibung von Alan Kay befolgen [6]:

1. Alles ist ein Objekt,
2. Objekte kommunizieren durch das Senden und Empfangen von Nachrichten (welche aus Objekten bestehen),
3. Objekte haben ihren eigenen Speicher (strukturiert als Objekte),
4. Jedes Objekt ist Instanz einer Klasse (welche ein Objekt sein muss),
5. Die Klasse beinhaltet das Verhalten aller ihrer Instanzen (in der Form von Objekten in einer Programmliste),
6. Um eine Programmliste auszuführen, wird die Ausführungskontrolle dem ersten Objekt gegeben und das Verbleibende als dessen Nachricht behandelt

Eine längere und detailliertere Erklärung zu diesen Paradigma findet man in der Vorlesung „Programmiermethodik“ von Dr. Eva Zangerle.

Wichtigste Sprachen: Smalltalk, Java, C++, Eiffel

7 Zusammenfassung

Jedes der besprochenen Paradigmen hat seinen passenden Anwendungsbereich. Einige davon scheinen den Menschen schwerer verständlich, andere hingegen einfacher. Doch je nachdem was für ein Programm man schreibt und wie das Ergebnis ausschauen soll, kann man es sich durch obige Paradigmen einfacher machen.

Literatur

[1] Wulf „Zimbardo 1995“

[2] Justus Piater „Einführung in die Programmierung“

[3] Christian Sternagel und Harald Zankl.Skriptum zur Vorlesung Functional Programming

[4] Danielo König „Programmierung: Womit anfangen?“

[5] Karl Greuter „Seminararbeit über das Thema Paradigmen der objektorientierten Programmierung“

[6] Alan Kay „The Early History of Smalltalk“