



© Sergey8grus & Leszek Glasner, Shutterstock.com

Tanzen statt krabbeln

Moderne und zukunftsweisende Programmierung ist objektorientiert. Ist das so? Wo liegen die Vorteile der objektorientierten Programmierung gegenüber der prozeduralen? Ist Objektorientierung (OO) immer der beste Weg? Und wie neu ist Objektorientierung wirklich?

Von Sonja Telscher, G.I.B

Eines ist sicher: OO-Programmierung ist absolut up to date und im Trend. OO-Sprachen wie Java, Self, C++ und C# sind vom Markt nicht mehr wegzudenken. Neu ist die Idee der OO jedoch nicht. Bereits in den 60er-Jahren wurden die ersten OO-Sprachen entwickelt. „Smalltalk“ und „Simula 67“ waren Vorreiter und Wegbereiter für die moderne objektorientierte Programmierung und beeinflussten die nachfolgenden Sprachen.

Auch in der SAP-Welt gibt es die Objektorientierung bereits seit den 90ern, doch erst in den letzten zehn Jahren kam Dynamik in die Anwendungen. Weshalb erst so spät und warum laufen bis heute so viele Abap-basierte Programme prozedural? „Die Programme sind ausgereift und haben sich bewährt“, erklärt Felix Grab, Entwickler bei G.I.B. „Es macht keinen Sinn, eine komplexe Anwendung, die im Laufe von Jahrzehnten einen perfekten Reifegrad erreicht hat, nur um der Modernität Willen neu aufzusetzen. Neben einem großen Programmieraufwand gäbe es auch ein erhöhtes Risiko, eine fehlerfreie Anwendung, die in zahlreichen Konzernen und Unternehmen läuft, durch eine schicke objektorientierte Anwendung abzulösen, die mit ‚Kinderkrankheiten und Bugs‘ behaftet ist.“

OO ein SAP-Thema

Doch der Anspruch an die Programme und deren Funktionsumfang steigt. Themen wie Echtzeit und Mobile gewinnen immer mehr an Bedeutung. Um den wachsenden Anforderungen gerecht zu werden, gewinnt auch im SAP-Umfeld die objektorientierte Programmierung immer mehr an Bedeutung. „Mit der OO bieten sich ganz neue Funktionsmöglichkeiten. Das Coding ist wesentlich übersichtlicher. Fehler können schneller identifiziert und behoben werden. Zudem sind OO-Programme wesentlich schlanker und schneller“, erklärt Grab. In objektorientierten Programmen sind Methoden und Attribute konkreten Objekten zugeordnet. Diese Objekte befinden sich als „Kapseln“ im Programm und können je nach Bedarf angesteuert werden, während in der prozeduralen Programmierung die jeweiligen Funktionen und Prozeduren immer wieder eingefügt werden müssen. „Stellen Sie sich ein prozedurales Programm wie eine Reparaturanleitung für Ihr Auto vor. Bestimmte Arbeitsabläufe sind identisch, egal ob Sie den Luftfilter oder



Sonja Telscher ist Head of Marketing bei G.I.B.

den Ölfilter wechseln. Dennoch wird jeder Schritt in beiden Kapiteln mit allen redundanten Abläufen und Einzelheiten beschrieben.“ Die Folge sei, dass sich bestimmte Arbeitsabläufe von Kapitel zu Kapitel wiederholen. In der Objektorientierung würde jeder Teilschritt der Reparatur auf Wiederverwendbarkeit geprüft und vom eigentlichen Einsatzgebiet („Luftfilter wechseln“) abgekoppelt. Die Reparaturanleitung wäre dann nicht mehr ein Buch, sondern ein dreidimensionaler Raum, in dem verschiedene Kapseln mit unterschiedlichen Objekten und Methoden schweben. Ein Anleitungsfehler könnte sich in der „Buchversion“ an vielen Stellen wiederholen, da dieselbe Tätigkeit immer wieder beschrieben wird. Das erschwert die Fehlersuche und erhöht den Aufwand bei Änderungen in der Anleitung. Im dreidimensionalen OO-Raum kann der Anleitungsfehler nur in einer einzigen Kapsel liegen. Es reicht also aus, diese Kapsel zu identifizieren und den Fehler zu beheben. Auch Ergänzungen und Änderungen müssten nur in dieser Kapsel durchgeführt werden.

Die Kunst der Abstraktion

Die Kunst der OO-Programmierung liegt also darin, den Sachverhalt so weit zu abstrahieren, dass eine komplexe Gesamtanforderung (Auto reparieren) in wenige Kapseln zerlegt wird, und natürlich darin, diese Kapseln untereinander kommunizieren zu lassen. Ulf Bauer, Entwickler bei G.I.B., sieht die Beteiligung der Fachseite an der Konzeptentwicklung als zentralen Punkt an. „Der richtige Bauplan lässt sich nur erstellen, wenn das zu lösende Problem verstanden, zerplückt, abstrahiert und aggregiert wird. Als Entwickler muss man sich unbedingt vom Automatismus befreien, jede Anforderung bereits im Kopf zu codieren“, warnt Bauer. Um ein gutes OO-Programm zu erstellen, müsse man sich von der Frage leiten lassen, was immer bleibt, was sich ändern kann und was sich definitiv ändern wird. „Unser OO-Haus sollte im Bauplan möglichst wenige tragende Wände besitzen. Damit bleibt für die Raumgestaltung maximale Flexibilität“, erklärt Bauer. „In der Objektorientierung krabbeln wir nicht mehr eine enge Gasse entlang, wir tanzen in großen Räumen.“



Bitte beachten Sie auch den Community-Info-Eintrag ab Seite 83

