

Wichtig:

Im obigen (Aufgabe 1) Klassendiagramm sind alle Attribute als „private“ deklariert. Dies ist ein Vorgriff auf Implementierungsüberlegungen (Kapselung), zu dem man von CASE Werkzeugen mehr oder weniger gedrängt wird, da viele eine Default-Sichtbarkeit annehmen (oft "private"). Um dieses Modell implementieren zu können muss es jedoch um weitere Methoden ergänzt werden, die den Zugriff auf die privaten Felder ermöglichen.

Dies wurde hier bewusst noch nicht gemacht, da es das Modell aufbläht ohne wesentlich Neues hinzuzufügen. In einer solch frühen Phase wäre der Aufwand Zugriffsmethoden (getter & setter) zu modellieren sogar kontraproduktiv: Wenn sich das Modell noch häufig ändert (Felder verändert, gelöscht oder verschoben werden), müssten die entsprechenden Zugriffsmethoden jedes mal mit geändert werden.

Das vermeidet man indem man erst mal so tut als ob man direkt auf die Felder zugreifen könnte und die Zugriffsmethoden erst in der Implementierungsphase hinzufügt -- und auch das eher automatisiert statt manuell (siehe später "Encapsulate Field"-Refactoring).

Merken Sie sich: Wenn wir nicht von den CASE-Werkzeugen dazu angehalten werden würden Sichtbarkeiten zu modellieren, würden wir sie in der Anforderungserfassung und -analyse erst mal völlig ignorieren, da die Erfassung der konzeptuellen Essenz des Systems im Vordergrund steht, nicht die Produktion von ausführbarem Code.

(Das gleiche gilt für alle Klassendiagramme im Blatt 3)