

Situation: Die letzten Vorbereitungen...

Kjeld Matthiessen, der Anwendungsentwickler der InSoft GmbH, ist schon ganz ungeduldig. Als Informatiker weiß er natürlich, dass die ganzen Vorbereitungen für eine erfolgreiche Projektdurchführung notwendig sind, dennoch möchte er immer am liebsten gleich loslegen.



Am heutigen Vormittag spricht er Dorothee Kemper, die für die Systemanalyse des Projektes zuständig ist, an.

Kjeld Matthiessen: „Hi Doro! Wann kann ich denn endlich mit der Programmierung des CarRent-Projekts beginnen?“

Dorothee Kemper: „Ach Kjeld, Du bist mal wieder so ungeduldig wie ein kleines Kind in der Weihnachtszeit. Aber ich kann Dich beruhigen: Du kannst bald loslegen.“

Kjeld Matthiessen: „Kann ich Dir denn noch irgendwie helfen?“

Dorothee Kemper: „Ich muss eigentlich nur noch die benötigten Klassen- und Objektdiagramme erstellen, damit wir Herrn Kohler von der CarRent GmbH die Systemarchitektur veranschaulichen können. Danach kannst Du mit Hilfe der Diagramme mit der Codierung beginnen.“

Kjeld Matthiessen: „Das ist doch schnell erledigt – gib mir mal die Anwendungsfallbeschreibungen!“

Auszug aus dem Pflichtenheft ...

Die Fahrzeuge der Autovermietung sollen erfasst werden. Wichtige Merkmale sind: KFZ-Kennzeichen, Fahrzeuggestellnummer, Farbe, Kilometerstand, Tankvolumen, Fahrzeugklasse (Mittelklasse, Kleinwagen, etc.), Karosserieform (Cabrio, SUV, Sportwagen, Coupe, etc.), durchschn. Kraftstoffverbrauch und natürlich Marke, Modell, Motorisierung und spezielle Ausführungskennzeichen (Klimaanlage,

Zudem sollen die wesentlichen Daten der Fahrzeugmieter verwaltet werden. (...) ... und die Mietvorgänge erfasst.

(...) Bei der Mietwagenbuchung wird passend zu den Wünschen des Kunden ein freies Auto für den angegebenen Zeitraum reserviert.

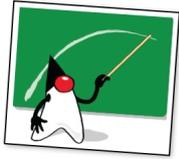
(...) Nach Rückgabe des Fahrzeugs wird der Kilometerstand festgehalten, die gefahrenen Kilometer ermittelt und eine Rechnung geschrieben. Der Rechnungsbetrag setzt sich zusammen aus den Tagespauschalen des jeweiligen Fahrzeugtyps und ggf. einem Zuschlag bei Überschreiten der Freikilometer. Zudem wird dokumentiert, ob das Fahrzeug mängelfrei zurückgegeben worden ist.

Entwicklung eines Klassendiagrammes

Arbeitsauftrag:

1. Überlege, welche Informationen für den Anwendungsfall "PKW erfassen" benötigt werden. Halte diese auf einem Blatt Papier fest.
2. Lese Dir den Informationstext kurz durch und unterstreiche die wesentlichen Sachverhalte.
3. Entwickle ein Klassendiagramm für den Anwendungsfall "PKW erfassen". Berücksichtige dabei die Informationen zu Klassendiagrammen und ihre Dokumentation in der UML.

Information zu Klassendiagrammen und ihre Dokumentation mit der UML



Klassen- und Objektdiagramme mit der Unified Modelling Language (UML) Das Klassendiagramm

Klassendiagramme sind der zentrale Diagrammtyp der UML. Sie beschreiben die Klassen eines Systems, ihre Eigenschaften, Methoden und ihre Beziehungen. Klassendiagramme werden in allen Phasen des Software Engineerings eingesetzt. Ein Klassendiagramm modelliert den Teilausschnitt der realen Welt, der zur Lösung eines bestimmten Problems benötigt werden.

Klassenname
Attribute
Methoden

Ein Klassendiagramm besteht aus dem fett gedruckten Namen der Klasse sowie einem Bereich für die Attribute (Eigenschaften) und einem Bereich für die ausführbaren Methoden der Klasse.

Ein Klassendiagramm herleiten

Konto
-kontonummer : int
-kontoInhaber : String
-saldo : double
+ausgabe() : String

Vor der Modellierung der Struktur von Softwaresystemen müssen aus den Dokumentationen der Systemanforderungen (z. B. aus den Anwendungsfalldiagrammen und -beschreibungen) die benötigten Objekte identifiziert werden. Bei der anschließenden Ermittlung der Attribute müssen – damit das System Speicherplatz dafür zur Verfügung stellt, der Datentyp eines jeden Attributs festgelegt werden.

Primitive Datentypen

In Java existieren acht elementare Datentypen, sogenannte primitive Datentypen:

Datentyp	Größe	Wertebereich	Standard
boolean	1 Bit	true / false	false
char	16 Bit	U+0000 ... U+FFFF	\u0000
byte	8 Bit	-128 ... +127	0
short	16 Bit	-32.768 ... +32.767	0
int	32 Bit	-2.147.483.648 ... +2.147.483.647	0
long	64 Bit	-9.223.372.036.854.775.808 ... +9.223.372.036.854.775.807	0
float	32 Bit	±1,4E-45 ... ±3,4E+38	0.0
double	64 Bit	±4,9E-324 ... ±1,7E+308	0.0

Bei dem "Datentyp" `String` (Achtung großer Anfangsbuchstabe) handelt es sich schon um eine eigene Klasse, die eine Zeichenkette von `char` implementiert.

Zudem sollte stets folgende Frage beantwortet werden: Was muss die Klasse über sich selbst wissen, damit sie arbeiten kann?

Die Sichtbarkeit von Attributen und Methoden wird wie folgt gekennzeichnet:

+	für public	(engl. öffentlich)	unbeschränkter Zugriff
-	für private	(engl. privat)	nur die Klasse selbst kann es sehen
#	für protected	(engl. geschützt)	Zugriff nur von der Klasse sowie von Unterklassen
~	für package		innerhalb des Pakets sichtbar

Abschließend sind die ausführbaren Funktionen, die Methoden einer Klasse zu definieren. Die Methoden sind die Schnittstellen einer Klasse, über die sie die Dienste, die sie implementiert, allen anderen Klassen des Systems anbietet.

Das Objektdiagramm

<u>konto1 : Konto</u>
kontonummer : int = 12345
kontoinhaber : String = "Thomas Bucker"
saldo : double = 1352,74

Falls statt einer Klasse ein konkretes Objekt dargestellt werden soll, verwendet man ein Objektdiagramm. Dieses stellt den aktuellen Zustand, d. h. die Belegung der Attribute, eines Objektes einer Klasse dar. Dazu wird im Diagramm vor dem Klassennamen der Name des darzustellenden Objektes (der Instanz) gefolgt von einem Doppelpunkt eingetragen, diese Angaben werden darüber hinaus komplett unterstrichen.