

# Musterlösung Übung 5

## Entwurf, Prototypen

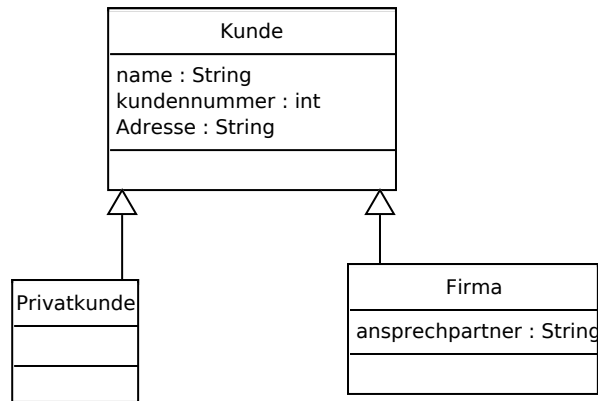
1. Sie haben einem Kunden für ein größeres Softwareprojekt in der Analysephase einen „**Wegwerf-Prototypen**“ zur einmaligen Benutzung erstellt. Überraschenderweise ist der Kunde jedoch mit diesem Prototypen schon vollauf zufrieden, möchte auf den eigentlich geplanten Softwareentwicklungsprozess verzichten und in Zukunft nur noch mit dem Prototypen arbeiten.

Welche Schwierigkeiten könnten sich dadurch ergeben (abgesehen davon, dass das Projekt wohl nicht in vollem Umfang durchgeführt wird, was Ihren Umsatz schmälert)?

- *Die mit Hilfe des Wegwerf-Prototypen in der Entwurfsphase gefundenen Fehler im Design können nicht ohne weiteres eliminiert bzw. verbessert werden, da im Entwurf davon ausgegangen wurde, dass das eigentliche Produkt in einer von Grund auf neu entworfenen Implementierung erst entsteht.*
- *Fehlende Funktionalitäten bzw. vergessene Anforderungen werden ggf. erst spät entdeckt.*
- *Nach Einführung der Software ist eine spätere Migration auf eine neue oder bessere Variante schwierig, da die Datenstrukturen im Wegwerf-Prototypen nur auf diesen schnellen Ansatz hin implementiert wurden.*
- *Generell ist die Fehleranfälligkeit beim Rapid Prototyping hoch.*
- *Da der Wegwerf-Prototyp ggf. nicht auf Wiederverwendung und leichte Erweiterbarkeit hin entwickelt wurde, sind spätere Änderungen an der Software sehr aufwändig (und teuer). Dies ist ein sehr schwerwiegendes Argument beim ansonsten an Erweiterbarkeit und Wiederverwendung orientierten objektorientierten Programmierparadigma.*

2. Installieren Sie, falls noch nicht vorhanden, ArgoUML (<http://argouml.tigris.org/>), und entwerfen Sie ein Klassendiagramm zur Verwaltung von Kunden, wobei Sie sinnvollerweise die allen Kunden gemeinsamen Daten wie **Name** und **Kundennummer** in einer Basisklasse **Kunde** speichern, und detailliertere Kategorien wie **Privatkunde** oder **Firma** als abgeleitete Klasse mit zugehörigen Attributen definieren. Lassen Sie das gezeichnete Klassendiagramm von ArgoUML als Java-Klasse(n) speichern.

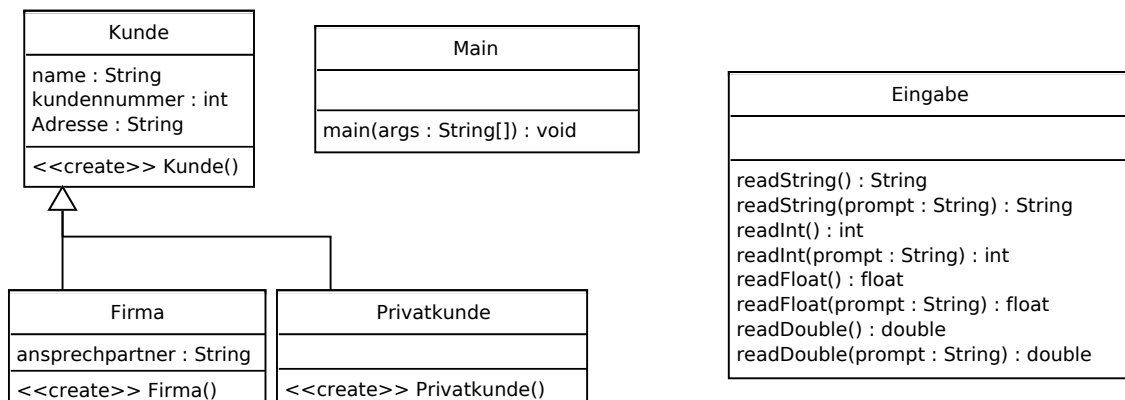
*Siehe Beispiele in uebung05-argouml.*



3. Welche Entwurfsmethode haben Sie in der vorigen Aufgabe angewandt?

*Objektorientiertes Design (OOD).*

4. Erweitern Sie die Klassen aus Aufgabe 2 um Konstruktoren, die die entsprechenden Attribute eines Objektes vom Typ **Privatkunde** oder **Firma** mit Hilfe der Klasse **Eingabe** von der Tastatur einlesen (mindestens Name und Kundennummer). *Siehe Beispiele in uebung05-argouml-erweiterung.*



5. ROBOCODE: Sehen Sie sich den Quelltext des Sample-Roboters **Tracker** in Robocode an. Dort ist als Kommentar ein Hinweis darauf versteckt, wie sich mit einigen kleinen Änderungen seine Bewegungen, und damit seine Trefferquote, enorm steigern lassen. Sie müssen hierfür ggf. seine Basisklasse ändern. Verbessern Sie **Tracker** und vergleichen Sie seine Erfolgsquote mit der Originalversion.

Siehe → *TrackerNeu.java*

6. ROBOCODE: Stellen Sie die beiden für Ihr Roboterteam erstellten Teamroboter (oder Vorlagen hierfür, wenn Sie mit der Programmierung nicht begonnen haben) mit ArgoUML als Diagramm(e) dar. Dies kann auch für Ihre spätere Dokumentation nützlich sein.

*Tipp: Importieren Sie das VERZEICHNIS, in dem Ihre Roboter liegen, als „Datei“, und fügen Sie bei der Frage nach dem Classpath sowohl das Robocode-Verzeichnis, als auch die darin liegende Datei robocode.jar ein. Mit „Drag & Drop“ können Sie die Klassen in das Klassendiagramm übernehmen, und ggf. die Basisklassen ebenfalls übernehmen und Pfeile ziehen.*

