

SOFTWARETECHNIK

Aufgabenblatt 8

Aufgabe 8.1

Aufgabe 8.2

a)

Eine Komponente im Software Engineering ist eine abgeschlossene Einheit eines Software-Systems, die meist eine logisch getrennte Aufgabe erfüllt und die dazu benötigten Inhalte zusammenfasst. Komponenten zeichnen sich durch eine hohe Kohäsion nach innen (Zusammengehörigkeit der Bestandteile der Komponente) und lose Kopplung nach außen (geringe Abhängigkeit von anderen Komponenten) aus. Komponenten existieren in einem Kontext: sie werden von anderen Komponenten verwendet und verwenden ihrerseits auch andere Komponenten. Dazu ist es nötig, Schnittstellen explizit zu definieren.

Neben der Schaffung einer logischen Struktur bringen Komponenten den Vorteil der Wiederverwendung und Austauschbarkeit. Ein Softwaresystem könnte beispielsweise auf einen Kunden zugeschnitten werden, indem eine bestimmte Komponente ausgetauscht wird.

b)

Software Komponenten...

- sind von ihrer Umgebung möglichst unabhängig (\Rightarrow Austauschbarkeit)
- haben explizite Schnittstellen, verringern dadurch Kopplung
- fördern Wiederverwendbarkeit und Qualität ("best in class", Produktivitätssteigerung, geringerer Schulungsaufwand etc.)
- können unabhängig versionsverwaltet werden

c)

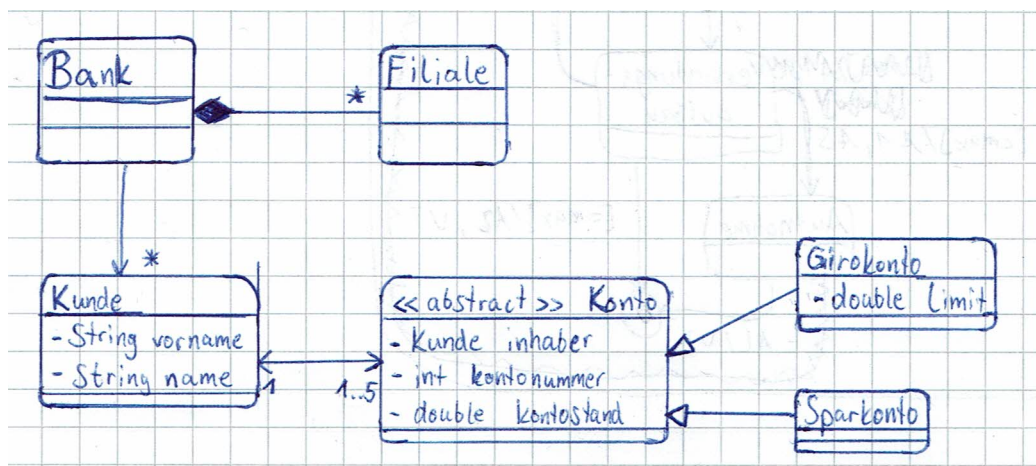
Mit Glue Code wird derjenige Programmcode bezeichnet, der die Komponenten untereinander verbindet. Dies kann auf verschiedenen Arten erfolgen (z.B. durch Scripting-Sprachen) und ist recht einfach auszutauschen.

Aufgabe 8.3

a)

Die Idee hinter der Code-Generierung ist, durch das automatisierte Generieren von Programmcode(teilen) aus einer Modellsprache sowohl Zeit einzusparen als auch eine höhere Qualität in Hinblick auf die Strukturierung und Lesbarkeit zu erzielen.

b)



Nun folgt die textuelle Beschreibung der geforderten Elemente des Klassendiagramms:

```
classdiagram BankSystem {  
    class Kunde {  
        String vorname;  
        String name;  
    }  
    abstract class Konto {  
        Kunde inhaber;  
        int kontonummer;  
        double kontostand;  
    }  
    class Girokonto extends Konto {  
        double limit;  
    }  
    association [1] Kunde <--> Konto [1..5];  
}
```