Contents

Chapter 1

Allgemeine Informationen

E-Mail	frank.*******ber@gmx.de
Anzahl Vorlesungen	11
Länge Vorlesung (1x)	255 Minuten

Chapter 2

Objektorientierung

2.1 Grundlegendes

Vor der Objektorientierung wurden Systeme in Prozeduren, Funktionen und Daten zerlegt. In einem solchen System liegen die Prozeduren und Daten getrennt voneinander. Es entstanden große monolithische Systeme die schlecht zu warten und schlecht zu skalieren waren. Änderungen in der Datenstruktur führten zu Änderungen in allen betroffenen Funktionen. Die Lösung zu diesem Problem war der OO-Ansatz, bei dem Prozeduren und Daten zu einem neuen zusätzlichen Strukturelement zusammengefasst wurden. Diese Objekte bringen eine Kapselung innerhalb eines Programms. Objekte sind grundlegende Beschreibungsmöglichkeiten von Dingen aus der realen Welt. Dabei beschreiben die Daten in einem Objekt die Eigenschaften des Äquivalenz in der realen Welt und die Prozeduren (auch Funktionen genannt) die Fähigkeiten. Durch die Zerlegung des Systems zu Objekten müssen Änderungen in der Datenstruktur nur noch in einem kleinen festgelegten Teilbereich angepasst werden (statt das gesamte System nach Abhängigkeiten zu durchsuchen und gegebenenfalls an zu passen). Ein Objekt kann alles abbilden das mit den Menschlichen-Sinnen erfasst werden kann.

2.2 8 - Sätze der Objektorientierung

Vererbung In der Objektorientierung können Eigenschaften und Fähigkeiten durch Vererbung von einem Objekt zu anderen Objekten weitergegeben werden [Beispiel: Das Objekt Person vererbt die Eigenschaften Name, Vorname an das Objekt des Studenten; Ein Student ist also eine Untermenge von Personen].

Polymorphie Durch Polymorphie können in ähnlichen Objekten, bei gleichem Input, unterschiedliches Verhalten ausgelöst werden.

Kapselung . Durch Kapselung wird festgelegt welche Eigenschaften bzw. Fähigkeiten nach außen (außerhalb eines Objekts) sichtbar sind.

Geheimhaltung Bei der Geheimhaltung geht es darum, dass ein Objekt offenbart „was“ es tut, aber nicht wie es dies tut.

Persistenz . Ein Objekt verbleibt, durch die Eigenschaft der Persistenz, in einem Zustand bis es von außen geändert wird.

Nachrichten . Ein Objekt kann via Nachrichten einem anderen Objekt eine Information übermitteln.

Objektidentität Objekte sind durch ihre Objektidentität (einzigartige Kennzeichnung und physikalischer Ort) eindeutig und unterscheidbar.

Klassen Ein OO-System besteht aus Klassen. Zusammenfassung von Objekten mit ähnlichen Eigenschaften und Fähigkeiten.

2.3 Beziehungen zwischen Objekten

2.3.1 Multiplizität

Durch die Multiplizität, die innerhalb von Objekten festgelegt wird, werden die jeweils maximalen Anzahlen an Verbindungen für jede Beziehung angegeben (1 - 1; 1 - n; n - n). Diese Eigenschaften werden in einem UML Diagramm auf den Verbindungen an den jeweiligen Objekten angegeben.

2.3.2 Assoziation

Assoziationen sind Verbindungen zwischen Objekten. Dabei gibt es drei verschiedene Formen. In einem UML Diagramm werden diese Beziehungen durch einen einfachen Strich mit einer Beschriftung (Name) und eine Mengenrelation angegeben.

- α und β kennen sich
- α und β sind mit einander verbunden
- Zu jedem α gibt es ein β

2.3.3 Teil-Ganzes Beziehung (Aggregation)

Wenn ein Obejkt ein Teil von einem größeren Objekt ist oder ein Teil eines Ganzem ist findet diese Verbindung Platz. Im UML Diagramm werden diese Beziehungen mit einer Raute am Ganzen und einem einfachen Strich zum Teilobjekt dargestellt. Ein Objekt kann immer nur in einem Objekt gleichzeitig sein.

- α besteht aus β
- α enthält β
- β ist Teil von α

Komposition

Komposition beschreibt eine zeitliche (logische) Abhängigkeit. Wenn das Ganze nicht mehr existiert, kann auch das Teil nicht mehr existieren. In einem UML Diagramm wird dies mit der ausgemalten Raute dargestellt.

2.3.4 Gerichtete Beziehung

Beschreibt eine unidirektionale Beziehung. α kennt β aber β kennt α nicht. In einem UML Diagramm wird diese Beziehung durch einen Pfeil in die Richtung der Beziehung gekennzeichnet.

2.4 OO-Sicht von Heeg

Laut Heeg beobachten wir Phänomene, die aktuell Teil des Weltgeschehens sind (Ein Phänomen könnte die Vorlesung sein). Aus dieser Beobachtung entstehen Begriffe, die in Klassen umgewandelt werden. Aus diesen Klassen entstehen Objekte die wiederum das Geschehen widerspiegeln (identifizieren, abstrahieren, instanziieren, repräsentieren).

2.5 Objektorientiertes Modell

Das Basismodell besteht aus Objekten und Klassen, die aus Eigenschaften und Fähigkeiten bestehen. Das Basismodell ist Teilmenge des statischen Modells, dass dieses durch Beziehungen und Vererbungen erweitert. Im dynamischen Modell ist das statische Modell enthalten und erweitert dieses um Aktivitäten, Abläufe/Sequenzen, Nachrichten und Zustände. Es gibt verschiedene Betrachtungen von dynamischen Aspekten. Hierzu zählen die Beschreibung der Aktivitäten, Zustände, Abläufe und Nachrichten in gleichnamigen Diagrammen (Aktivitätsdiagramm, Zustandsdiagramm, Ablaufdiagramm). Alles wird anschließend im Modell zur Systemnutzung zusammengefasst das durch den Anwendungsfall und die use-cases ergänzt wird.

2.5.1 Anwendungsfall (Modell zur Systemnutzung)

Ein Anwendungsfall ist eine Situation in der ein Objekt oder eine Menge von Objekten benutzt oder darauf zugegriffen wird. Der Anwendungsfall kann auch andere Objekte zulassen, muss also nicht auf ein spezifisches Objekt bezogen sein (Bsp. Anwendungsfall: Von A nach B kommen; Objekt zur Lösung des Problems: Auto; es gäbe jedoch auch die Möglichkeit zu laufen, fliegen etc.). Ein Objekt kann auch Antwort auf mehrere verschiedene Anwendungsfälle sein.

2.5.2 Zustandsdiagramm

Das Zustandsdiagramm wird repräsentiert durch alle Möglichen Zustände, die mit den Aktionen verbunden werden. Die Zustände werden in rechteckigen Kästen beschrieben und die Aktionen sind die Pfeile die die Übergänge von einem Zustand in den Anderen repräsentieren.

2.5.3 Aktionsdiagramm

Im Aktionsmodell werden alle möglichen Aktionen mit Transitionen dargestellt und mit einander verknüpft. Das Diagramm ähnelt dem UML Format

und verknüpft die in rechteckigen Kästen untergebrachten Aktionen mit einfachen Pfeilen. Außerdem gibt es noch Entscheidungspunkt und Zusammenführungspunkte, die der Übersicht halber eingebaut werden um Aktionen auf zu splitten oder zusammen zu führen (Es muss während der Aktion eine Entscheidung getroffen werden die zu zwei unterschiedlichen Zuständen führt, Zwei gleiche Aktionen zeigen auf den selben Zustand).

2.5.4 Sequenzdiagramm

Das Sequenzdiagramm wird durch alle teilnehmenden Objekte nebeneinander dargestellt. Jedes Objekt erhält eine "Live-Line", die durch einen Strich (meist) nach unten dargestellt wird. Werden nun Nachrichten zwischen den einzelnen Akteuren geschickt, werden diese mit Querstrichen zwischen den "Live-Lines" abgebildet. Diese Querstriche werden in zeitlicher Abfolge untereinander angegeben und mit einer bezeichnenden Beschreibung beschriftet. Das Sequenzdiagramm bildet nur einen Pfad aus dem Aktionsdiagramm ab.

Chapter 3

Softwarearchitektur

3.1 Definition

Begriff der Architektur: "Architektur ist die Wissenschaft von der Gestaltung und Konstruktion von Bauwerken.", somit ist die Softwarearchitektur die Wissenschaft von der Gestaltung und Konstruktion von Software-Bauwerken. Die Architektur, definiert den Rahmen, damit die Anforderungen an das Gebäude bestmöglich realisiert werden können. Dabei definiert die Architektur die Struktur und den Kontext für Design und Umsetzung. Dabei beschreibt sie nur den groben Lösungsentwurf und lässt genaue Details aus. Architektonische Entscheidungen sind grundlegend.

Was braucht man für die Erstellung eines IT-Systems?

- Textuelle Beschreibung
- System Umgebung
- Anwendungsfälle
- Anzahl Nutzer
- benötigte Rechenleistung
- Kosten
- Schnittstellen
- Technologien

3.2 Fähigkeiten eines Architekten

Was sind die Fähigkeiten und Eigenschaften eines Architekten?

- methodische Fähigkeiten
- Wissen: Wie komme ich zur benötigten Architektur? (Entscheiden, Beraten, verteidigen)

- Vermittler zwischen den Interessengruppen (Auftraggeber, Auftragnehmer)
- Fachliche und technische Fähigkeiten (Einarbeitung in das Fachgebiet des Auftraggebers)
- Diplomatisches Geschick
- Kommunikative Grundfähigkeit

3.3 Ablauf und Aufgeben eines Architekten

Der grobe Ablauf und die Aufgaben eines Architekten:

- Anforderungsanalyse (wenn nciht vorhanden, dann noch ermitteln)
- Skizze oder Grobplanung (um eine Vorstellung davon zu schaffen, wie es aussehen könnte, erstes Modell[POC, UML, ...])
- Feinplanung, Konstruktionszeichnungen, weitere Detaillierungen der Architektur, nicht funktionale Aspekte müssen hier beachtet werden (Sicherheit, Performance, ...)
- Beantragung einholen
- Überwachen des Bauvorhabens
- Endabnahme durch Führen

3.4 Spezialisierung in der IT-Architektur

Enterprise-Architekt: Entwirft Geschäftsprozesse und Unternehmensarchitekturen.

System-Architekt: Entwirft ganze Systeme die die Geschäftsprozesse abbilden.

Technologie-Architekt: Entwirft Komponenten innerhalb eines Systems.

Die drei Architekten, stehen in einer hierarchischen Struktur und ”erben” im Übertragenen Sinne die Aufgaben der darüber liegenden Ebene.

3.5 Qualitätskriterien in der IT-Architektur

- Performance, Antwortzeit, Durchsatzrate (Fähigkeit des Systems auf Anfragen in angemessener Zeit zu antworten)
- Robustheit (Eigenschaft eines Systems auch unter ungünstigen Bedingungen zuverlässig zu funktionieren)
- Sicherheit (Fähigkeit des Systems Fremdzugriffen abzuwehren)
- Skalierbarkeit (Fähigkeit eines Systems auf eine sich ändernde Anzahl von Anfragen angepasst zu werden)
- Wartbarkeit (Fähigkeit eines Systems angepasst bzw. erweitert zu werden)

- Bedienbarkeit (Das System ist einfach zu bedienen)
- Portabilität (Die Fähigkeit, das System auf unterschiedlichen Plattformen lauffähig zu machen)
- Verfügbarkeit (Die Wahrscheinlichkeit, dass das System die Anfragen während eines vereinbarten Zeitraums erfüllt)
- Testbarkeit (Die Fähigkeit ein System auf alle Anforderungen zu testen)

3.6 Architekturprinzipien

1. Die die einfachste Architektur, die alle Anforderungen übernimmt
2. Prinzip der losesten Kopplung: Architektur-Bestandteile sollen so wenig wie möglich von einander abhängen.