

## Einführung in die CRC-Analyse

Stefan Joos, Prof. Martin Glinz, 1997

### Ziel

Ziel der CRC-Analyse ist es, Aufgabe und Zweck von Klassen in einem Systems zu definieren und deren Rolle im System festzulegen. Dabei wird entschieden, über welche Kenntnisse und Informationen die Instanzen der Klassen verfügen, d.h. auf welche Art und Weise die Funktionalität und die "Intelligenz" über das System verteilt wird. Eine frühzeitige CRC-Analyse erlaubt einen einfachen Überblick

- über die gesamte Funktionalität eines Systems,
- welche Klassen zur Bereitstellung von Funktionen nötig sind (und um welche Klassen das System gegebenenfalls ergänzt werden muß),

und welche Aufgaben von einer Klasse selber realisiert werden und welche an andere Klassen delegiert werden.

### Definition

**Aufgabe (einer Klasse)** – Die Aufgabe legt fest, welche Funktionalität eine Klasse besitzt und welche Rolle diese dadurch im Gesamtsystem einnimmt.

**Verantwortlichkeit (einer Klasse)** – Verantwortlichkeiten sind Dienstleistungen, die eine Klasse bzw. die Objekte einer Klasse für die Objekte anderer Klassen erbringen. Die Aufgabe einer Klasse ist eine abstrakte Zusammenfassung der Gesamtheit ihrer Verantwortlichkeiten.

**Klassenkarte** – Eine Klassenkarte (engl.: class responsibility card, kurz CRC) ist ein Dokument, das für eine Klasse deren Aufgabe (Zweck) und deren Verantwortlichkeiten innerhalb des Systems definiert. Auf einer Klassenkarte wird zudem beschrieben, an welche Klassen Aufgaben delegiert werden.

### Leitidee

Die Leitidee bei der CRC-Analyse ist die Vorstellung, ein System durch eine Reihe von internen Agenten zu betrachten. Jeder dieser Agenten ist für ein bestimmtes Leistungspaket zuständig und kann weiterhin Leistungen an andere interne Agenten in Auftrag geben. Ein Agent sollte so konzipiert sein, daß er ein möglichst abgeschlossenes Leistungspaket seiner Umgebung zur Verfügung stellt und einen möglichst geringen Kommunikationsaufwand mit anderen Agenten benötigt. Für den kreativen Akt der Zuordnung von Aufgaben zu Agenten sind zwei konträre Strategien vorstellbar:

- Die möglichst gleichmäßige Verteilung der Funktionalität über die beteiligten Agenten hinweg
- Die Konzentrierung der Funktionalität auf wenige Agenten und damit die Aufteilung in "intelligente" und in "dumme" Klassen

Für den jeweiligen Problembereich ist festzulegen, welche Strategie sinnvollerweise verwendet wird.

### Notation

**Klasse:** <Klassenname>

#### Beschreibung

< Kurze, circa 4-10 zeilige Prosabeschreibung des Zwecks und der Aufgabe der Klasse >

#### Verantwortlichkeiten

##### 1. <Bezeichnung der Verantwortlichkeit>

< Beschreibung der Verantwortlichkeit >

-> < Liste der Klassen, an die Leistungen delegiert werden >

##### 2. <Bezeichnung der Verantwortlichkeit>

< Beschreibung der Verantwortlichkeit >

-> < Liste der Klassen, an die Leistungen delegiert werden >

...

### Beispiel:

**Klasse:** Konto

#### Beschreibung

Diese Klasse repräsentiert ein Kundenkonto in der Datenbank der Bank. Alle Zugriffe auf und Modifikationen der Bankkonten müssen über diese Klasse erfolgen.

#### Verantwortlichkeiten

##### 1. Einrichten und Aufheben eines Kontos

Erstellt ein Kundenkonto bzw. löscht ein Konto aus der Datenbank

-> Datenbank, Betreibungsregister

##### 2. Zugriff auf den Kontensaldo

Liefert den aktuellen Saldo, wie er in der Datenbank der Bank festgehalten ist.

-> Datenbank

##### 3. Modifikation des Kontensaldos

Durchführen von Transaktionen zum Einzahlen eines Betrages auf das Konto oder zum Abheben eines Betrages vom Konto. Der Saldo nach einer Kontoabhebung darf nicht unter den Minimalsaldo fallen.

-> Datenbank

##### 4. Modifikation der Benutzerdaten

Veränderung der Parameter eines Kundenkontos (Bsp. Minimalsaldo, Zugriffsrechte)

-> Datenbank