

PK

Quellcode als Kommunikationsmedium

Kommentare und Namen

Programmierer kommunizieren miteinander über Kommentare wobei Qualität der Kommentare entscheidend (nicht Umfang):

was man für Verwendung wissen muss
statt Beschreibung/Wiederholung der Syntax

Namen unterstützen die Intuition

Auffindbarkeit von Information von großer Bedeutung:

Faktorisierung soll intuitiv sein und
wichtigste Informationen soll an Schnittstellen stehen

Wo Information zu finden ist

Klasse, Interface:

Zweck, Grobstruktur, Besonderheiten

Methoden, Konstruktoren:

Parameter, Verhalten, Ergebniswerte, Bedingungen für Aufruf

Objektvariablen:

Zweck, Eigenschaften für Konsistenz

Kommentare als Zusicherungen

Vorbedingungen auf Methoden und Konstruktoren:

was vor Aufruf gelten muss, hauptsächlich Parameterwerte

Nachbedingungen auf Methoden und Konstruktoren:

was nachher gilt; was gemacht und zurückgegeben wird

Invarianten auf Variablen, Klassen, Interfaces:

müssen in konsistenten Zuständen stets erfüllt sein

History-Constraints auf Variablen, Klassen, Interfaces:

Entwicklung von Objekten im Laufe der Zeit

Aufgabe: Unterscheidungsmerkmale

Such Sie in Gruppen zu zwei bis drei Personen Antworten auf folgende Fragen:

1. Wie kann man Vorbedingungen von Nachbedingungen unterscheiden?
2. Wie kann man Nachbedingungen von Invarianten unterscheiden?
3. Wie kann man Invarianten von History-Constraints unterscheiden?

Zeit: 3 Minuten

Gute Faktorisierung

zusammengehörige Eigenschaften zu übersichtlichen Einheiten zusammengefasst

Eigenschaften und Aspekte, die nichts miteinander zu tun haben, unabhängig voneinander änderbar, obwohl Programmänderungen kaum voraussehbar

Klassen-Zusammenhalt und **Objekt-Kopplung** helfen bei der Abschätzung

Klassen-Zusammenhalt

Klassen-Zusammenhalt = Grad des Zusammenhangs zwischen Klasseninhalten

hoher Klassen-Zusammenhalt = Hinweis auf gute Faktorisierung

hoher Klassen-Zusammenhalt dadurch erkennbar:

- Variablen und Methoden passen gut zusammen

- Namen und Kommentare sind treffend gewählt

- Änderungen verringern Klassenzusammenhalt merklich

Objekt-Kopplung

Objekt-Kopplung = Stärke der Abhängigkeiten der Objekte voneinander

schwache Objekt-Kopplung = Hinweis auf lokale Änderbarkeit

schwache Objekt-Kopplung dadurch erkennbar:

- wenige nach außen sichtbare Methoden und Variablen

- wenige Nachrichten an andere Objekte

- geringe Anzahl an Parametern pro Nachricht

Ersetzbarkeit und Verhalten

Methoden in Untertypen müssen sich so verhalten,
wie von Methoden in Obertypen erwartet

Kommentare in Untertypen spezifizieren selbes Verhalten wie in Obertypen,
aber Kommentare in Untertypen können das Verhalten genauer spezifizieren

Ersetzbarkeit entkoppelt Programmteile:

