UNIVERSITÄT
LEIPZIG

Softwaretechnik 2024/25 – Übung 04

Prof. Dr. Norbert Siegmund
Stefan Jahns, M.Sc.

Aufgabe 1: Grundlagen

a) Was versteht man unter dem Begriff „Verantwortlichkeit“?

- Das Wissen, welches ein Objekt verwaltet und anbietet
- Die Aktionen, die es ausführen kann
- Spezifiziert was ein Objekt tut (nicht wie)

Aufgabe 1: Grundlagen

b) Was sind Kollaborationen und wie stehen sie in Beziehung zu Verantwortlichkeiten?

- Anfragen an Dienste, die benötigt werden, um Verantwortlichkeiten zu erfüllen
- Kollaborationen enthüllen Kontroll- und Informationsflüsse
- Können fehlende Verantwortlichkeiten offenbaren
- Analysen von Kommunikationsmustern können fehlerhaft zugewiesene Verantwortlichkeiten identifizieren

Aufgabe 1: Grundlagen

c) Was sind Richtlinien zum Entwerfen guter Klassen-Hierachien?

Modelliere eine “kind-of” Hierarchie:

- Unterklassen sollten alle geerbten Verantwortlichkeiten unterstützen, und eher noch mehr

Schiebe gemeinsame Verantwortlichkeiten so hoch wie möglich:

- Klassen, die gemeinsame Verantwortlichkeiten teilen sollten von einer gemeinsamen abstrakten Superklasse erben; führe fehlende Superklassen ein

Aufgabe 1: Grundlagen

c) Was sind Richtlinien zum Entwerfen guter Klassen-Hierachien?

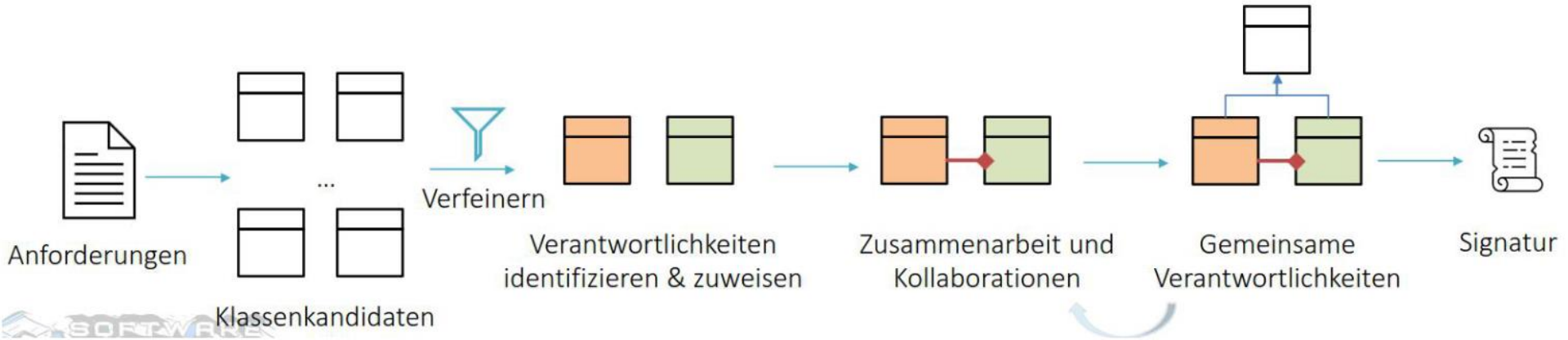
Stelle sicher, dass abstrakte Klassen nicht von konkreten Klassen erben:

- Eliminiere dies durch die Einführung weiterer gemeinsamer abstrakter Superklassen: abstrakte Klassen sollten Verantwortlichkeiten in einem implementierungsunabhängigen Weg unterstützen

Eliminiere Klassen, die keine neue Funktionalität hinzufügen:

- Klassen sollten entweder neue Verantwortlichkeiten oder eine bestimmte Implementierung von vererbten Verantwortlichkeiten hinzufügen

Aufgabe 2: Responsibility-Driven Design



Aufgabe 2: Responsibility-Driven Design

a) Geben Sie für das obige Szenario mindestens fünf geeignete Klassen an. Ordnen Sie jeder gewählten Klasse ein generelles Prinzip der Klassen-auswahl aus den „Prinzipien der Klassenauswahl“ der Vorlesung zu.

Klassen	Prinzip
Geschäft	Physikalisches Objekt
User/Konto	Physikalisches Objekt
Behälter	Physik. Objekte / Kategorien von Klassen
QR-Code	Konzeptuelle Entität / Interface zum System
Statistik	Konzeptuelle Entität
App/ Kartenansicht	Interface zum System
E-Mail (nicht im Text)	Interface zum System
…	Weitere Klassen möglich

Aufgabe 2: Responsibility-Driven Design

b) Welche Verantwortlichkeiten haben ihre Klassen?

Klasse	Funktion	Daten
Geschäft	Datenstruktur	Name, Position
User/Konto	Behälter verknüpfen	Name, Behälter, (E-Mail)
Behälter	Status geben/setzen	ID, Größe, Status
QR-Code	Berechnet Pixel	Pixel ↔ ID
Statistik	Aggregiert	Ausleih-Anzahl

Aufgabe 2: Responsibility-Driven Design

c) Bestimmen Sie Kollaborationen und Beziehungen (bzw. Vererbungen) ihrer Klassen.

Klasse	Kollaborationen
Geschäft	-
User/Konto	Behälter verknüpfen Scannt QR-Code
Behälter	-
QR-Code	Erhält Behälter / ID zur Berechnung
Statistik	Protokolliert Behälter