

Gao Ruilin
Chen Jing
Liu Chang

Aufgabe 1 Information Hiding :

a)

b)

final-Modifikators kann auch in der Klasse Service- Registry möglich und sinnvoll sein. Der Zweck der Information Hiding besteht darin, die Sicherheit zu erhöhen und die Programmierung zu vereinfachen.

Die final-Modifikators kann Klassen, Attribute und Methoden beschreiben.

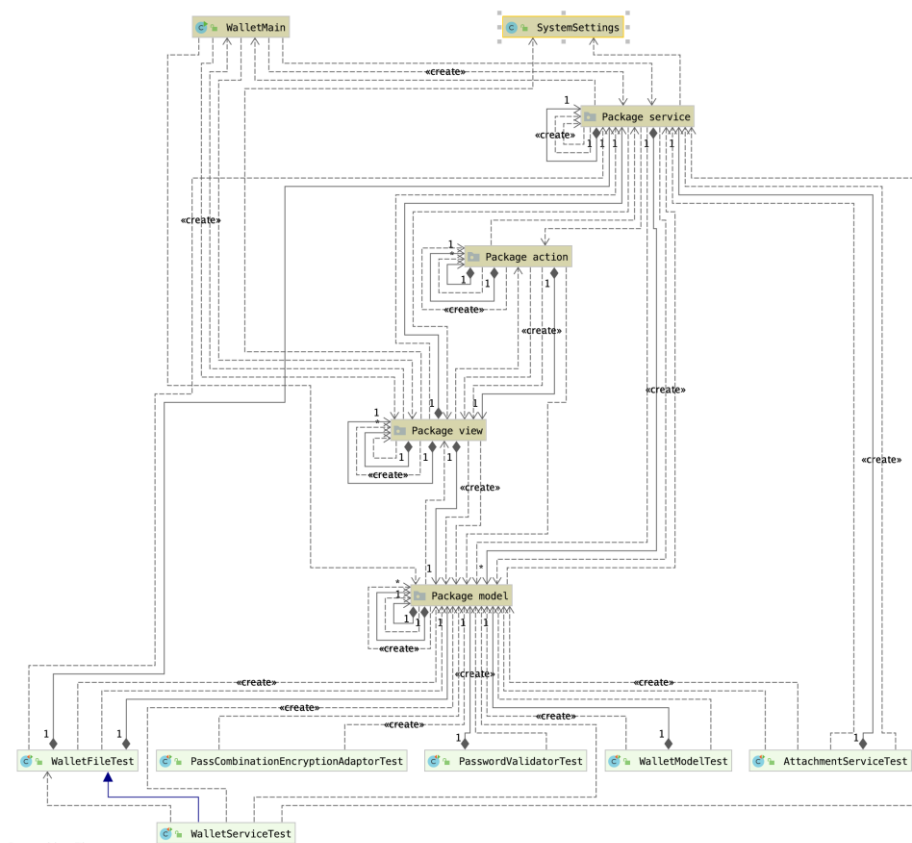
Wenn final eine Klasse ändert, wird die Klasse zur endgültigen Klasse und kann nicht vererbt werden.

Wenn die endgültige Änderungsmethode verwendet wird, wird diese Methode zur endgültigen Methode und kann nicht von Unterklassen überschrieben werden. Diese Methode kann jedoch weiterhin vererbt werden.

Wenn das endgültige Änderungsattribut (Variable) konstant ist, bleibt sein fester Wert unverändert und der Wert kann nicht geändert werden

Aufgabe 2 Architektur:

a)



1. Schicht: Präsentation -> view - Beinhaltet das UI

2. Schicht: Anwendungslogik -> service & action - Die packages verarbeiten die Daten aus der Datenerhaltung (Schicht 3)

und leiten die Ergebnisse an die Präsentationsschicht (view) weiter.

3. Schicht: Datenerhaltung -> model - In den Klassen des packages model werden Daten gespeichert und geladen.

Es handelt sich nicht um eine strikte Schichtenarchitektur. Begründung:

Die Klasse ExportItemsAction im package action aus der Schicht

Anwendungslogik kann auf die Klasse DialogUltis

aus der Schicht Präsentation zugreifen. DialogUltis liegt aber in einer höheren Schicht als ExportItemsAction.

Der Zugriff auf eine höhere Schicht darf aber bei einer strikten

Schichtenarchitektur nicht vorkommen.

b) Durch Singleton-Muster ist uns erlaubt, die Anzahl unserer erzeugten Objekte zu kontrollieren.

Unsere Objekte verwalten wir nun durch ein Klassenattribut namens unique, auf das wir Zugriff über eine öffentliche Klassenoperation gewähren. Diese nennen wir instance().

Diese Funktion überprüft, wenn sie aufgerufen wird, ob bereits ein Objekt des Klassentyps vorhanden ist und ruft, wenn dies nicht der Fall ist, unseren Konstruktor auf. Dies ist nun möglich, da wir uns innerhalb derselben Klasse befinden.

Aufgabe 3 Persistierung in evault:

a)

Form der Datenpersistierung :

Serialisierung von internen Objektstrukturen
JSON – einfaches, lesbares Format

genau persistierte Daten :

Postbank
URL = "https://www.postbank.de/"
accountType = "Onlinekonto"

b) try-with-resources

Unabhängig davon, ob die Anweisung normal oder abnormal endet, muss sie nach der Verwendung geschlossen werden

c) JSON-Struktur

d)

zwei Probleme