

Diskussion am: 11.12.2024

Aufgabe 1: Grundlagen

- a) Warum sind Design Patterns wichtig?
- b) Nennen Sie 3 GoF-Design-Patterns und ordnen Sie jeweils die Zweck-Klassifikation zu.
- c) Beschreiben Sie die Funktion eines Interfaces und geben Sie an, in wie weit sich Interfaces von Klassen-Vererbung unterscheiden.

Aufgabe 2: C4-Diagramm

Für diese Aufgabe ist das folgende Szenario gegeben:

Sie sollen eine Software-Anwendung für ein Leipzig-weite Mehrweg-To-Go-Behälter für Getränke und Speisen entwickeln. Statt Pfand zu bezahlen, ist jeder Behälter mit einem QR-Code ausgestattet, welcher bei der Ausleihe von Usern mit der Anwendung gescannt und damit mit ihrem Konto verknüpft wird. Hierfür registrieren sich User vorab.

User können teilnehmende Geschäfte in einer übersichtlichen Kartenansicht suchen. Der Scan-Vorgang muss schnell funktionieren. Überschreiten User die Rückgabefrist, wird automatisch eine Gebühr von 1€ pro Woche über einen externen Zahlungsdienstleister eingezogen. Geschäfte können Behälter über eine eigene Ansicht in der Anwendung bestellen.

User-Daten sind sicher zu speichern, damit keine Nutzungsmuster erbeutet werden können. Die Stadt Leipzig finanziert das Projekt und wünscht Zugang zu aggregierten Statistiken, etwa zur Anzahl der ausgegebenen Behälter.

- a) Erstellen Sie Level 1 (Context) und Level 2 (Container) eines C4-Diagramms für die Funktionalitäten, die im obenstehenden Szenario beschrieben werden.
- b) Erstellen Sie Level 3 (Components) eines C4-Diagramms für den Container, der unter anderem die Kundenkonten verwaltet.

Aufgabe 3: Composite-Pattern

Für diese Aufgabe ist das folgende Szenario gegeben:

Sie entwickeln ein Betriebssystem für Netzwerk-Datenspeicher (Network-Attached Storage, NAS). Sowohl für einzelne Dateien als auch für ganze Ordner sollen Aktionen, wie etwa das Berechnen der Größe, oder das Ändern von Zugriffsrechten ausgeführt werden können. Zusätzlich sollen sich verschiedene externe Datenquellen als spezielle Ordner in der Dateistruktur einbinden lassen.

- Recherchieren Sie das Composite-Pattern. Nennen Sie analog zur Vorlesung die Zweck-Klassifikation und die Motivation (zu lösende Herausforderung).
- Für das gegebene Szenario soll für den ersten Prototypen die Größe von Dateien, Ordnern, und Nextcloud-Integrations-Ordnern berechnet werden können. Geben Sie das UML-Diagramm für das Composite-Pattern mit diesen drei Klassen an.
- Diskutieren Sie Gemeinsamkeiten und Unterschiede des Composite-Patterns mit
 - dem Visitor-Pattern und
 - dem Decorator-Pattern.

Aufgabe 4: Entwurfsmuster und Konfigurierbarkeit

Gegeben ist folgendes Szenario und Diagramm:

Für die Menüplanungs-Software einer Kantine sollen einzelne Hauptgerichte modelliert werden. Jedes Hauptgericht kann normales Gericht, oder aber vegetarisch, oder kalorienreduziert sein, wobei mehrere Kategorien gleichzeitig zutreffen können; es kann also beispielsweise ein kalorienreduziertes vegetarisches Gericht geben. Der initiale Klassenentwurf (UML-Diagramm unten) berücksichtigt diese Eigenschaften.

a) Die Kantine möchte in Zukunft auch glutenfreie Hauptgerichte anbieten. Welche Aspekte des Klassenentwurfs erschweren diese Erweiterung?

b) Mit welchem Entwurfsmuster lässt sich die Modularität des Entwurfes verbessern? Korrigieren Sie den UML-Klassenentwurf entsprechend.

c) Implementieren Sie Ihre neue Lösung für mindestens eine Eigenschaft eines Hauptgerichts in Java.

