Babeș-Bolyai Universität, Cluj Napoca
Fakultät für Mathematik und Informatik
Objekt-Orientierte Programmierung, 2021

Labor 4

Diskussion: 12./13./14. April, **Abgabe:** 18. April, **Wert:** 6/50

Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!

Friedrich ist der Manager eines Obstgeschäfts, und er braucht eine Anwendung, um sein Obstlager zu verwalten. Für jedes Produkt werden folgende Attribute gespeichert: Name, Herkunft, Haltbarkeitsdatum, Menge und Preis. Friedrich hat mit Hilfe von einem Analysten eine Liste von Anforderungen erstellt.

Funktionale Anforderungen

F1. Er möchte Produkte hinzufügen, löschen oder bearbeiten können. Jedes Produkt wird durch seinen Namen, seine Herkunft das Haltbarkeitsdatum eindeutig identifiziert. Falls ein vorhandenes Produkt hinzugefügt wird, wird nur seine Menge aktualisiert.
F2. Er möchte alle verfügbaren Produkte auflisten, die eine bestimmte Zeichenkette (String) enthalten (falls die String leer ist, werden alle verfügbaren Produkte zurückgeben), sortiert nach Name.
F3. Er möchte nur die Produkte sehen, die knapp sind (Menge weniger als X Punkte, wobei der Wert von X von Friedrich angegeben wird).
F4. Er möchte alle Produkte nach Haltbarkeitsdatum aufsteigend sortiert sehen.

Weitere Anforderungen

W1. Die Interaktion des Benutzers mit dem Programm passiert via Konsole. Dem Benutzer werden dort das Menu und Ergebnissen angezeigt, und dort auch die Befehle des Benutzers eingelesen (User Interface, UI).
W2. Das Programm soll bei Start Beispieldaten von 10 oder mehr Produkten beinhalten. Die Lagerdaten müssen **nicht** persistent gespeichert werden.

Qualitäts-Anforderungen

Q1. Die Anwendung soll dem untenstehende Architekturbeispiel folgen, und entsprechend **modularisiert** werden (mit namespaces für domain, repository, controller, ui);
Q2. Wo Zeiger notwendig sind, dort sind **Smart-Pointer** zu verwenden.
Q3. Code sollte **selbsterklärend** sein: Klassen, Methoden und Variablen sollen **zweckgemäße Namen** haben

Abgabekriterien

K1. Die Lösung kann aus mehreren Dateien bestehen. Diese sind als ZIP abzugeben **`L3_Nachname_Vorname.zip`** (kein RAR, 7z, o.ä., das Archiv beinhaltet nur Quellcode und keine Verzeichnisse)
K2. Die Datei **`Main.cpp`** enthält `main()`
K3. Das Programm muss mit dem GnuCompiler auf folgende Weise kompilierbar sein:
```
g++ -std=c++17 -o prog *.cpp
```

Empfehlungen

R1. Implementierung der Module in der Reihenfolge ihrer Verwendung, also aus der Perspektive des Benutzersicht: domain, ui, controller, repository.
R2. Um sicherzustellen dass repository und controller funktionieren, ohne Verwendung der Benutzeroberfläche, sind Unit-Tests nützlich;

Babeș-Bolyai Universität, Cluj Napoca
Fakultät für Mathematik und Informatik
Objekt-Orientierte Programmierung, 2021

Labor 4

Diskussion: 12./13./14. April, **Abgabe:** 18. April, **Wert:** 6/50

Teilnahme am Labor wird nur für diejenigen registriert, die etwas vorzeigen!

Architektur Beispiel: