



Labor 6

Abgabe: Kalenderwoche 21, Wert: 10/50

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

Die Laboraufgabe ist in Teams von 2-3 Personen zu bearbeiten.

In den folgenden Laboraufgaben wird die Anwendung zu erweitern sein!

Erweitern Sie Lösung zu Labor 5 entsprechenden untenstehende Anforderungen.

Alle Anforderungen aus Labor 5 bleiben gültig, sofern nicht ersetzt!

Funktionale Anforderungen

- F1. Ein Kunde soll einen oder mehrere Scooter reservieren können.
- F2. Ein Kunde soll die Liste seiner reservierten Scooter sehen können.
- F3. Das Programm verlangt zu Beginn die Auswahl eines Speichermodus: flüchtig oder persistent.
Flüchtig bedeutet, dass die Daten nach Programm-Ende verloren gehen. Persistent bedeutet, dass die Daten in einer Datei gespeichert werden.

Weitere Anforderungen

- W1. **Repository-Polymorphismus:** Erstellen sie eine abstrakte Repository-Klasse nach dem Vorbild des CRUD-Repository (aus Seminar 5). Je nach Benutzerauswahl des Speichermodus, wird ein In-Memory oder eine CSV-File-Repository instanziiert.
- W2. Adaptieren sie die bestehende In-Memory-Repository-Klasse als eine Unterklasse von CRUD-Repository. Bei Programmstart werden 10 Autos programmatisch ins Repository hinzugefügt.
- W3. Schreiben sie eine zusätzliche eine CSV-File-Repository-Klasse als eine Unterklasse von CRUD-Repository, die Autodaten in einer Textdatei speichert. Bei Programmstart werden die Einträgen aus derselben Textdatei gelesen, und bei jeder Änderung geschrieben. Die Textdatei soll in CSV Format gespeichert sein. Verwenden Sie die **iostream-Bibliothek**.

BEISPIEL im exakten(!) Format

```
Identifizier,Modell,Inbetriebnahmedatum,Kilometer,Standort,Status  
XAY,Rideway,2022-05-01,291,Strada Constanta 19-21,parked
```

Qualitäts-Anforderungen

- Q1. Generell sollte vermieden werden Objekte von Repository, Controller, Autos, Kunden, usw zu kopieren.
- Q2. Die Trennung der Zuständigkeiten der Schichten (domain, repository, controller, ui) soll gewährleistet sein;
- Q3. Code sollte eine konsistente Benennung aufweisen
 - a. Sprache: Englisch
 - b. Klassennamen sind Substantive, beginnen mit Großbuchstaben
 - c. Methodennamen starten mit Verben, beginnen mit kleinbuchstaben
- Q4. Das Hauptprogramm ist in der Datei **Main.cpp** und enthält die `main()`
- Q5. Die Tests für Repository und Controller sind in einer **test.cpp** und werden im `main()` vor dem Start der UI ausgeführt.

Abgabekriterien

- K1. Die Lösung kann aus mehreren Dateien bestehen. Diese sind als ZIP abzugeben **L5_TeamX.zip** (**X ist die Nummer**) (kein RAR, 7z, o.ä., das Archiv beinhaltet nur Quellcode und keine Verzeichnisse)
- K2. Das Programm muss mit dem GnuCompiler auf folgende Weise kompilierbar sein:
`g++ -std=c++20 -o prog *.cpp`