FH-OÖ Hagenberg/HSD
SDP3, WS 2025
Übung 1

Name: Aufwand in h:

Mat.Nr: Punkte:

Übungsgruppe: korrigiert:

Beispiel1: Fuhrpark (24 Punkte)

Ein Fuhrpark soll verschiedene Fahrzeuge verwalten: PKWs, LKWs und Motorräder. Entwerfen Sie dazu ein geeignetes Klassendiagramm (Klassenhierarchie) und ordnen Sie folgende Eigenschaften den einzelnen Klassen zu: Automarke, Kennzeichen und die Kraftstoffart (Benzin, Diesel, elektrisch oder Gas). Weiters muss jedes Fahrzeug ein Fahrtenbuch führen. Ein Eintrag im Fahrtenbuch speichert das Datum und die Anzahl der gefahrenen Kilometer an diesem Tag.

Geben Sie Set- und Get-Methoden nur dann an, wenn sie sinnvoll sind!

Die Fahrzeuge stellen zur Ausgabe eine `Print`-Methode zur Verfügung!

Ein Fuhrpark soll folgende Aufgaben erledigen können:

1. Hinzufügen von neuen Fahrzeugen.
2. Entfernen von bestehenden Fahrzeugen.
3. Suchen eines Fahrzeuges nach seinem Kennzeichen.
4. Ausgeben aller Fahrzeuge samt ihrer Eigenschaften und dem Fahrtenbuch auf dem Ausgabestrom und in einer Datei.
5. Verwenden Sie im Fuhrpark zur Verwaltung aller Fahrzeuge einen entsprechenden Container!
6. Der Fuhrpark muss kopierbar und zweisbar sein!

Die Ausgabe soll folgendermaßen aussehen:

```
Fahrzeugart: Motorrad
Marke:       Honda CBR
Kennzeichen: FR-45AU
```

```
04.04.2018:  52 km
05.06.2018:   5 km

Fahrzeugart: PKW
Marke:       Opel Astra
Kennzeichen: LL-345UI
04.07.2018:  51 km
05.07.2018:  45 km

Fahrzeugart: LKW
Marke:       Scania 1100
Kennzeichen: PE-34MU
04.08.2018:  512 km
05.08.2018:   45 km
07.08.2018:  678 km
14.08.2018:  321 km
```

Die Fahrzeugart wird nicht als Attribut gespeichert, sondern bei der Ausgabe direkt ausgegeben! Für den Fuhrpark ist der Ausgabeoperator zu überschreiben.

Für jedes Fahrzeug soll die Summe der gefahrenen Kilometer ermittelt werden können und der Fuhrpark soll die Summe der gefahrenen Kilometer aller seiner Fahrzeuge liefern. Verwenden Sie dazu entsprechende Algorithmen.

Die folgenden Punkte gelten auch für alle nachfolgenden Übungen:

1. Werfen Sie wo nötig Exceptions und geben Sie Fehlermeldungen aus!
2. Implementieren Sie einen ausführlichen Testtreiber und geben sie entsprechende Meldungen für die Testprotokollierung aus.
3. Verfassen Sie weiters eine Systemdokumentation mit folgendem Inhalt:

    - Verteilung der Aufgaben auf die Teammitglieder.
    - Anforderungsdefinition mit eventuell zusätzlich getroffenen Annahmen. Treffen Sie für alle unzureichenden Angaben sinnvolle Annahmen und begründen Sie diese.
    - Systementwurf in Form eines Klassendiagrammes mit allen Klassen und deren Beziehungen, inklusive der wichtigsten Attribute und Methoden. Geben Sie zusätzlich in den entsprechenden Header-Dateien den Verfasser an! Das Klassendiagramm muss nicht vollständig dem implementierten Sourcecode entsprechen! Geben Sie weiters Ihre Designentscheidungen an und begründen sie diese!
    - Testausgaben: die Ausgaben sollen aussagekräftig sein, damit aus der Ausgabe erkennbar ist, was getestet wurde.

- Vollständig dokumentierter Sourcecode (Korrektur der Tutoren). Verwenden Sie Doxygen-Kommentare.

4. Die einzelnen Klassen (Komponenten) werden direkt im Quellcode dokumentiert und mit Hilfe von Doxygen eine HTML-Doku generiert.

5. Führen Sie zusammen mit Ihrer Teamkollegin bzw. mit Ihrem Teamkollegen vor der Realisierung eine Aufwandsschätzung in (Ph) durch und notieren Sie die geschätzte Zeitdauer am Deckblatt.

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!