FH OBERÖSTERREICH HSD FH-HAGENBERG

Systemdokumentation
Projekt Fuhrpark

Version 1.0

S. Offenberger, S. Vogelhuber

Hagenberg, 11. Oktober 2025

Inhaltsverzeichnis

1 Organisatorisches

1.1 Team

- Simon Offenberger, Matr.-Nr.: S2410306027, E-Mail: Simon.Offenberger@fh-hagenberg.at
- Susi Sorglos, Matr.-Nr.: yyyy, E-Mail: Susi.Sorglos@fh-hagenberg.at

1.2 Aufteilung der Verantwortlichkeitsbereiche

- Simon Offenberger
  - Design Klassendiagramm
  - Implementierung und Test der Klassen:
    - Object,
    - RecordEntry,
    - DriveRecord,
    - Vehicle,
  - Implementierung des Testtreibers
  - Dokumentation
- Simon Vogelhuber
  - Design Klassendiagramm
  - Implementierung und Komponententest der Klassen:
    - Garage
    - Car,

    - Bike und
    - Truck
  - Implementierung des Testtreibers
  - Dokumentation

1.3 Aufwand

- Simon Offenberger: geschätzt 10 Ph / tatsächlich x Ph
- Simon Vogelhuber: geschätzt x Ph / tatsächlich x Ph

2 Anforderungsdefinition (Systemspezifikation)

In diesem System werden die Tiere eines Zoo's abgebildet und dort gespeichert. Die Tiere speichern das Gewicht in Kilogramm als Ganzzahl und werden mit einer forlaufenden Nummer identifiziert. Sie besitzen eine gemeinsame Schnittstelle die folgende Funktionalität liefert:

- Gib einen Laut (entsprechende Ausgabe auf std::cout).
- Liefere einen String mit den gespeicherten Attributen.
- Erstelle einen Klon von sich selbst.

Der Zoo speichert alle Tiere und besitzt die Tier-Objekte nach dem Hinzufügen. Via Zugriffsmethoden kann auf die Tiere zugegriffen werden und eine String-Methode liefert eine Repräsentation aller Tiere im Zoo mit allen Attributen als eine abgeschlossene Zeichenkette. Zusätzlich kann der Zoo inklusive all seiner Tiere kopiert und zugewiesen werden.

3 Systementwurf

3.1 Klassendiagramm

Hier wird das Klassendiagramm eingefügt. Sollte dieses nicht auf eine A4-Seite passen, so kann es in eine eingene pdf-Datei ausgelagert werden. Verweisen Sie an dieser Stelle auf diese Datei.

3.2 Designentscheidungen

Im Klassendiagramm wurde der Polymorphismus angewendet, um unterschiedliche Tierarten mit der gemeinsamen Schnittstelle 'Animal' anzusprechen. Die Klasse 'Zoo' speichert einen Container mit der abstrakte Basisklasse 'Animal' als Elementtyp und kann somit alle bestehenden und auch neuen Tierarten verwalten, die sich von der gemeinsamen Basisklasse 'Animal' ableiten.

Designentscheidungen sind von entscheidender Bedeutung für die Qualität und den Erfolg einer Softwareanwendung. Sie beeinflussen nicht nur die technische Umsetzung, sondern auch die Fähigkeit der Anwendung, zukünftigen Anforderungen gerecht zu werden und Änderungen effizient zu bewältigen.
Sie beantworten meist folgende Fragen:

- Warum wurde die Klassenhierarchie so gewählt?
- Wurden Design Pattern verwendet und warum?
- Wurde Abstraktion und der Polymorphismus angewendet?
- Wie kann die Klassenstruktur einfach erweitert werden?
-

4 Dokumentation der Komponenten (Klassen)

Die Dokumentation der einzelnen Klassen und Komponenten erfolgt direkt im Quellcode mit Doxygen-Kommentaren. Erzeugen Sie danach eine HTML-Doku und verweisen Sie auf die Start-HTML-Datei.

Die HTML-Startdatei befindet sich im Verzeichnis ./../doxy/html/index.html

5 Testprotokollierung

```
Visual Leak Detector read settings from: C:\Program Files (x86)\Visu
Visual Leak Detector Version 2.5.1 installed.
Testing Invalid Animal Creation...
[PASS] Exception caught: Weight must be a positive number
[FAIL] Expected exception for invalid cat weight not thrown.
Testing Null Animal Addition to Zoo...
[PASS] Exception caught: null_pointer param in Zoo::Add(...)
Testing Animal Weights...
[PASS] Animal weight test passed.
[PASS] Animal weight test passed.
[PASS] Animal weight test passed.
Testing Animal IDs...
[PASS] Animal ID test passed.
[PASS] Animal ID test passed.
[PASS] Animal ID test passed.
Testing Animal Sounds...
[PASS] Animal sound test passed.
[PASS] Animal sound test passed.
[PASS] Animal sound test passed.
Testing Animal ToString...
[PASS] Animal ToString test passed.
[PASS] Animal ToString test passed.
[PASS] Animal ToString test passed.
Testing Animal Cloning...
[PASS] Animal cloning test passed.
[PASS] Animal cloning test passed.
Testing Zoo Contents...
[PASS] Zoo ToString test passed.
No memory leaks detected.
Visual Leak Detector is now exiting.
```

6 Quellcode

Die Klassen werden entsprechend der Klassenhierarchie von oben nach unten angegeben. Zuerst die Header-Datei, gefolgt von der Implementierung. Der Testtreiber (oder Client-Klassen) werden am Ende angegeben.