Effizient Programmieren I & II

Coding Standard für das Programmierprojekt Generische Flugzeugsimulation

2. Juni 2017

Inhaltsverzeichnis

1 Vorwort

Im Zuge der Vorlesungsreihe Effizient Programmieren I & II soll innerhalb einer Semesterarbeit ein vollständiges Programmierprojekt erstellt werden. Ein Teil dieser Ausarbeitung stellt dieses Dokument, welches Regeln für die Programm Erstellung und Formatierung des Codes bereitstellt. Hierbei liegt der Schwerpunkt am Code Layout und weniger am an der Umsetzung der Programmierung. Ziel ist es einen einheitlich gut leserlichen und aufgebauten Code zu erhalten.

2 Allgemein

2.1 Projektaufbau

Standard 1 Das Projekt soll in der Entwicklungsumgebung in Unterordner aufgeteilt werden.

Standard 2 Die Klassen sollten inhaltlich in gleichen Unterordnern zusammengefasst werden.

Standard 3 Header-Files sollen die Endung.hpp und Source-Files die Endung .cpp erhalten

2.2 Kommentare

Standard 4 Jedes File erhält einen Top-Kommentar mit den wichtigsten Informationen über das jeweilige File

Standard 5 Kommentare sollten richtig und genau sein und auch bleiben

Standard 6 Kommentare und Quellcode sollten klar voneinander getrennt sein.

Standard 7 Beschreibung von Funktionen im Header File sollten ausführlich genug sein, dass der Nutzer rein durch das lesen des Header-Files den Aufbau der Klasse versteht.

Standard 8 Hinter Variablen Deklarationen sollte, wenn es sich um eine physikalische Größe handelt, die SI-Einheit als Kommentar stehen.

3 Code Layout

3.1 Formatierung

Standard 9 Code/Funktionen sollte(n) in Blöcke eingeteilt werden.

Standard 10 Jede(r) Block/Funktion sollte einen eigenen Kommentar erhalten.

Standard 11 Jede Variablendeklaration sollte in einer neuen Zeile starten.

Standard 12 Der Code sollte in der Vertikalen angeglichen werden

Standard 13 Zwischen Kommentar und Funktion sollte eine Zeile Platz gelassen werden.

Standard 14 Wenn möglich sollten Klammern genutzt werden, um Zusammenhänge im Code besser zu erkennen.

Standard 15 Ein File sollte 300 Zeilen exklusive Kommentare nicht überschreiten.

Standard 16 An Funktion übergebene Variablen sollten untereinander geschrieben werden.

3.2 Namensgebung

Standard 17 Namen sollte klar und einzigartig sein.

Standard 18 Namen sollten aufschluss über deren nutzen geben.

3.3 Deklaration von Variablen

Standard 19 Variablen erhalten entsprechend Tabelle ... Präfixe

Standard 20 Die Deklaration von Variablen sollte im Header-File stattfinden.

4 Aufbau von Klassen und Funktionen

4.1 Konstruktor und Destruktor

4.2 Initialisierung

4.3 Methoden

5 Beispiel für eine Klasse