CAE-Anwendungsprogrammierung in einer höheren Programmiersprache

Bonusaufgabe

Paderborn, WS 2022/2023

Hintergrund:

Ziel dieser Aufgabe ist es, ein Programm zu schreiben, das die Wartungsplanung für verschiedene Maschinen vereinfacht.
Mit dem C++-Programm soll der Benutzer in der Lage sein, Wartungspläne zu erstellen und zu verwalten. Für eine Maschine gibt es beliebig viele Wartungspläne (mindestens einen), Mitarbeiter mit Qualifikationen, die den Plänen zugewiesen werden und Firmen, in welchen die Maschinen stehen mit Ansprechpartnern.

Durchführung:

Da Sie die Aufgabe in Gruppen bearbeiten: Verwalten Sie den Code in einem gemeinsamen Git Repository[1]. Anleitungen finden Sie dazu im Internet oder fragen Sie in der Vorlesung oder in Panda.

Abgabe:

Laden Sie den

- C++ Code mit allen zur Kompilierung benötigten Dateien inkl. Makefile,
- Screenshots der Menüführung sowie das UML-Diagramm,
- Einen Ordner mit txt-Dateien der jeweiligen Maschinen, Pläne etc., sowie
- Eine txt-Datei mit dem Link zum Git-Repository (entweder öffentlich zugänglich oder Andreas Schultz andreas.schultz@uni-paderborn.de einladen)

als zip-Archiv (benannt nach dem Gruppennamen) in PANDA bis zum 31. Januar 2023 12:00 Uhr hoch.

Vorstellung:

Stellen Sie Ihr Vorgehen und Ihr Ergebnis anschließend in einer kurzen Präsentation/Demo vor. Der Termin ist am Mittwoch, 1. Februar 2023, ab 14 Uhr im Raum P1.2.04.

Aufgaben:

1. Erstellen Sie ein UML-Diagramm der zu programmierenden Software anhand der folgenden Anforderungen bzw. Aufgaben. Überlegen Sie sich sinnvolle Aufteilungen der Aufgaben sowohl in Klassen als auch auf die Personen der Gruppe. Überlegen Sie sich schon hier, welche Attribute und Methoden eine Klasse braucht und wie die Schnittstellen zu anderen Klassen aufgebaut sind.

[1] 1 GitLab der Uni Paderborn, GitHub, GitLab, Bitbucket oder vergleichbar

2. Das Programm (als Konsolenanwendung) soll dem Nutzer folgende Möglichkeiten bieten:
    a. Eine Maschine anlegen. Jede Maschine hat eine ID, einen Namen sowie zugehörige Firma. Maschinen unterteilen sich in stationäre Maschinen (mit einem Standort) und beweglichen Maschinen (mit einem Gewicht).
    b. Einen Wartungsplan für eine Maschine anlegen. Ein Wartungsplan soll eine ID, einen Namen, die betroffene Maschine, das Intervall, die notwendige Qualifikation (Auszubildender, Geselle, Meister) den zuständigen und genug qualifizierten Mitarbeiter sowie die Firma der Maschine haben.
    c. Intervalle anlegen. Intervalle sollen, einmal angelegt, für mehrere Maschinen genutzt werden können. Z.B. alle 4 Wochen oder 1x jährlich
    d. Mitarbeiter sollen eine ID, ihren Namen und ihre Qualifikation speichern können.
    e. Firmen sollen eine ID, ihren Namen, ihre eigenen Mitarbeiter sowie die Maschinen speichern können.
    f. Die Daten sollen gespeichert und bearbeitet werden können. Die in den Teilen a) – d) genannten Eigenschaften sollen in logisch sinnvollen .txt Dateien gespeichert werden und im Programm durch den Nutzer bearbeitet werden können. Ein einfaches Überschreiben mit neuen Werten in den .txt Dateien reicht. Beim Programmstart sollen eventuell vorhandene Daten aus Dateien eingelesen werden.
    g. Daten sollen gelöscht werden können. Beim Löschen soll auf eventuell verwendete Referenzen geachtet werden und diese sinnvoll angepasst werden. (z.B. wird ein Mitarbeiter gelöscht, soll er aus allen Wartungsplänen entfernt werden).
    h. Eine tabellarische und übersichtlich formatierte Übersicht aller Maschinen, Wartungspläne, Intervalle, Mitarbeiter, Firmen ausgeben, welche alle jeweiligen Daten enthalten. Maschinen, Firmen und Mitarbeiter sollen dabei alphabetisch nach ihrem Namen bzw. Nachnamen sortiert ausgegeben werden. Die Intervalle sollen nach ihrer Dauer sortiert ausgegeben werden.
3. Umsetzung
    a. Nutzen Sie im Programm mindestens einen Container aus der Standard Template Library von C++.
    b. Verwenden Sie bei Funktionen/Methoden mindestens je einmal einen call-by-value- und einen call-by-reference-Aufruf
    c. Nutzen Sie die objekt-orientierte Programmierung inkl. Vererbung wo es möglich und sinnvoll ist.
    d. Trennen Sie ihren Code für Klassen in eine Header- und eine Code-Datei auf.