Anforderungen für die große Hausaufgabe

Grundlegende Anforderungen

Die erste und wichtigste Anforderung ist es, dass der Programmcode kompilierbar sein soll, außerdem muss das Programm die vorher mit dem Praktikumleiter besprochene Spezifikation erfüllen. Beim Vorstellen der Aufgabe darf man von der besprochenen Aufgabe nicht abweichen, die vereinbarte Spezifikation muss vom Programm erfüllt werden.

Die Vorstellung der Aufgabe geschieht persönlich im Rahmen des Software-Labors auf Deutsch. Es ist möglich, dass der Laborleiter einige Fragen über den Aufbau und das Funktionieren des Programmes stellt, oder dass er um die Implementierung kleiner Veränderungen bittet, damit die selbständige Arbeit nachgewiesen werden kann.

Jeder Student soll eine viertel- bzw. halbseitige, mit dem Praktikumleiter vereinbarte Spezifikation abgeben. Diese Spezifikation soll als .pdf-Datei auf die HAWeb-Seite, zur für diesen Zweck aufgegebenen Aufgabe für 0 Punkte hochgeladen werden.

Die Spezifikation muss bis zum Ende der 10. Woche mit dem Laborleiter vereinbart werden. Eine Bedingung für die Erfüllung des Semesters ist die Akzeptierung der Großhausaufgabe bis zu dieser Frist.

Technische Anforderungen

Das Programm soll aus einem einzigen `main.c`-Datei bestehen. Diese Datei soll nur die `main`-Funktion enthalten, aus dieser Funktion sollen alle anderen Funktionen, die sich in einer unterschiedlichen `.c`-Datei befinden und mit Hilfe einer Header-Datei eingeschlossen wurden, aufgerufen werden. Die eventuellen `#define`-Konstanten sollen in einer `define.h`-Datei gespeichert werden.

Der Programmcode soll gut kommentiert, gut strukturiert und übersichtlich sein. Außerdem sollen die Variablenamen selbsterklärend sein. Die Verwendung von globalen Variablen soll vermieden werden.

Das Programm soll die folgenden Elemente enthalten:

- Dateibehandlung (z. B.: eine Topliste speichern, das Spiel speichern usw.)
- Strukturen
- dynamische Speicherbehandlung
- bei der Entwicklung einer Funktionsbibliothek soll auch ein Testprogramm vorhanden sein, um es vorzustellen, wie die Funktionsbibliothek funktioniert.

Dokumentation

Neben dem Programm müssen drei Dokumentation abgegeben werden:

- eine Benutzerdokumentation
- eine Programmiererdokumentation
- eine Testdokumentation

Die **Benutzerdokumentation** soll mindestens 3 Seiten lang sein (12 Punkt, einfacher Zeilenabstand). Hier soll eine kurze Vorstellung des Programmes stehen, was es verwirklicht und was mit dem Programm verwirklicht werden kann. Außerdem soll man es hier beschreiben, wie man das Programm verwenden kann, also die möglichen Kommandozeilenargumente, Menüs und ihre Elemente, Tasten.

Die **Programmiererdokumentation** soll vom Program abhängig mindestens 4 Seiten lang sein (ohne Deckblatt und Inhaltsverzeichnis). Diese Dokumentation kann auch mit Hilfe eines Generierprogramms für Dokumentationen wie z. B. *Doxygen* verwendet werden, in diesem Fall sollen die Beschreibungen im Quellcode stehen.

Über den Inhalt:

- Einige Sätze darüber, was das Programm verwirklicht:
  - *Das Programm liest eine Datei ein, der Benutzer kann es auswählen, wie die Datei modifiziert werden soll, diese Auswahl wird mit einer `switch`-Konstruktion ausgewertet, bei der Beendung des Programmes wird die Datei abgeschlossen.*
- Wofür die einzelnen Funktionen dienen, die Parameter und die Rückgabewerte, was die Funktion verändert (wie z.B. globale Variablen), wie sie die Fehler behandeln.

- *Die Funktion `zeichen_loeschen` übernimmt eine Zeichenkette und ein Zeichen als Parameter, löscht das angegebene Zeichen aus der Zeichenkette, und gibt einen Pointer auf die veränderte Zeichenkette zurück.*

- Wenn es globale Variablen gibt, sollen diese auch beschrieben werden: wofür sie dienen, wann sie einen Anfangswert erhalten, wann und wo sie verändert und verwendet werden.
  - *`spieler` speichert die Anzahl der Spieler, 1 beim Einspielermodus und 2 beim Zweispielermodus, diese Variable wird von der Funktion des Hauptmenüs verändert.*
- Strukturen, wofür sie dienen und wofür ihre Felder dienen.
  - *`struct person` ist eine verkettete Liste, sie speichert den Namen und das Geburtsdatum von Personen.*
- Aufzählungstypen, ihre Verwendung und ihre Werte.
  - *`spieler_zustand` speichert den aktuellen Zustand des spielers, ihre Werte sind STEHEN, GEHEN, LAUFEN, SPRINGEN.*
- Verwendete Algorithmen z. B. bei der Sortierung, Kodierung, und warum eben diese Algorithmen verwendet wurden.
  - *Bei der Sortierung der Daten der Personen habe ich das Verfahren Bubble sort verwendet, denn so ist die Speicherverwendung kleiner.*
- Wenn das Programm Dateien verendet, sollen diese Dateien beschrieben werden, und wenn das Programm ein eigenes Datenformat hat, dann das Format dieser Dateien.
  - *Die Daten einer Person werden in einer Textdatei gespeichert. In jeder Zeile befinden sich die Daten einer Person in der folgenden Reihenfolge: Nummer-Name-Geburtsdatum.*
- Wie das Programm mit dem Benutzer kommuniziert.
  - *Das Programm kann mit Hilfe von Menüs benutzt werden.*
- Fehler während der Ausführung sollen beschrieben werden, und wie das Programm auf sie reagiert.
  - *Wenn die Datei nicht geöffnet werden kann, wird das Programm mit dem Code -1 beendet.*
- Die Beschränkungen des Programmes sollen auch hier stehen:
  - *Das Programm kann nur Dateien öffnen, die kleiner als 4 GB sind.*
- Eventuelle bekannte Fehler im Programm sollen beschrieben werden:
  - *Wenn der Benutzer anstatt einer Zahl einen Buchstaben eingibt, kann das Programm abstürzen.*

In der **Testdokumentation** soll dokumentiert werden, dass das Programm bei verschiedenen Eingaben die entsprechenden Ausgabe erstellt, die Extremfälle richtig behandelt, und dass es einen Schutz gegen fehlerhafte Eingaben hat. Diese Dokumentation soll mindestens eine halbe Seite lang sein.

Die Dokumentationen müssen nicht gedruckt abgegeben werden, aber man muss die Dokumentation während der Vorstellung des Programmes dem Laborleiter zeigen. Der Quellcode muss nach der Akzeptierung auf die HAWeb-Seite hochgeladen werden.

Alles, was in diesem Dokument beschrieben wurde, muss auf Deutsch gemacht werden!