

Projektarbeit "Verkettete Liste sortieren"

1. Kurzbeschreibung

In einer Einzel- oder Gruppenarbeit (max. zwei Lernende) erstellen Sie ein Programm, mit welchem eine einfach verkettete Liste erstellt und danach sortiert werden soll.

Das Programm (ein Windows Konsolenprogramm) muss in C geschrieben werden, mit Hilfe von Visual Studio 2017 unter Verwendung der ANSI C Library (von Visual Studio bzw. Visual C++ 2017). Es sind keine C++ Befehle oder andere Bibliotheken erlaubt.

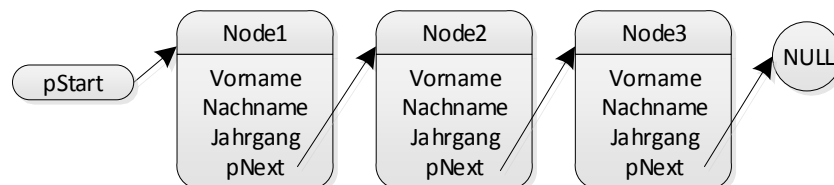
Zum Projekt gehört auch eine Grafik und eine Tabelle, mit welcher die Zeiten der Sortierung in Abhängigkeit (Funktion) der Anzahl Elemente dargestellt werden. Machen Sie mindestens 5 Messungen mit jeweils verschiedenen Anzahl von Elementen, die sich um einen Faktor 1000 unterscheiden sollten. Abschliessend erstellen Sie auch eine kurze Analyse der erzielten Resultate.

2. Datenstruktur und Aufbau der verketteten Liste

Die Knoten der Liste verwenden die folgende Datenstruktur:

```
typedef struct Person {  
    char Vorname[40];  
    char Nachname[40];  
    int Jahrgang;  
    struct Person* pNext;  
} struPerson;
```

Eine einfach verkettete List hat dann den folgenden Aufbau:



3. Datenerstellung, Variablen

Die Daten pro verwendetes Element sollen per Zufallszahlen-Generator, siehe C-Funktion rand(), nach den folgenden Regeln erzeugt werden:

- Jahrgang: eine Zahl von 1900 bis und mit 2018
- Vorname und Nachname: jeweils nur ein Grossbuchstabe 'A' bis und mit 'Z' (ohne Umlaute)
Beachten Sie, dass es sich bei den Variablen «Vorname» und «Nachname» um Strings handelt.

Es dürfen **keine globalen Variablen** verwendet werden.

4. Funktionalität

Das Programm besteht aus einem kleinen, einfachen, textbasierten Menü. Mit diesem Menü werden die folgenden Funktionalitäten aufgerufen:

- Liste erstellen.
Die Anzahl der gewünschten Elemente wird eingegeben.
- Liste löschen und entfernen
- Ein gewünschtes Element löschen.
Das gewünschte Element wird durch die Eingabe des Vor- und Nachnamens bestimmt. Wenn es mehrere Elemente mit dem gleichen Vor- und Nachnamen gibt, werden alle die gleich sind, unabhängig vom Jahrgang, gelöscht.
- Liste sortieren.
- List ausgeben.
Die Anzahl Elemente, welche ausgegeben werden sollen, wird eingegeben. Ist die Eingabe 0 werden alle Elemente ausgegeben.
- Programm beenden.

5. Main-Funktion

Das ganze Programm besteht aus einer minimalen main-Funktion mit einem textbasierten einfachen Menü, welches für jede Aufgabe eine entsprechende Funktion aufruft, wie z. B. CreateListe, SortListe usw.

Da keine globalen Variablen verwendet werden dürfen, müssen diese im «main» deklariert werden und bei Bedarf an die Funktionen übergeben werden.

6. Sortierung

Sie dürfen einen Sortieralgorithmus frei auswählen. Einzige Bedingung ist, dass er funktioniert.

Die Sortierung erfolgt fix nach dem Nachnamen und dann nach dem Vornamen. Bei der Sortierung wird die Reihenfolge der Elemente verändert und nicht deren Inhalte umkopiert.

7. Arbeitsaufteilung

Teilen Sie die Arbeiten anhand der Programmfunktionalität auf. Schreiben Sie bei jedem Kommentar zu einer Funktion hin, wer der beiden Gruppenmitglieder diese geschrieben hat. Das gleiche gilt für die Analyse mit den Messresultaten.

8. Erwartetes Resultat

Dieses Projekt muss folgende Elemente beinhalten:

- Programm mit der oben beschriebenen Funktionalität, codiert nach den oben beschriebenen Regeln
- Übersichtlicher, verständlicher Code mit blockweiser Kommentierung, d.h. es muss nicht jede Zeile kommentiert sein!
- Pro Funktion einen Kommentar mit dem Namen des Programmierers
- Eine Tabelle und Grafik mit den gemessenen Resultaten
- Eine schriftliche Kurzanalyse der erzielten Resultate (gemessenen Zeiten)

Auf jeder Seite der Kurzanalyse müssen folgende Angaben stehen:

- die Nachnamen und Vornamen der Gruppenmitglieder bzw. der Nachname und Vornamen bei einer Einzelarbeit
- Klasse
- Nummer der aktuellen Seite und die totale Anzahl Seiten

Eine entsprechende Word-Vorlage kann bei Bedarf verlangt werden.

9. Bonus

Sie können eine bessere Bewertung bzw. zusätzliche Punkte erzielen, indem Sie Ihr Programm wie folgt **freiwillig** um eine oder mehrere der folgenden Möglichkeiten erweitern:

- Um eine Auswahl nach welchen Variablen sortiert werden soll.
- Um einen weiteren Sortieralgorithmus für die Sortierung der Liste und somit um eine zusätzliche Kurve in der Grafik bzw. weitere Resultate in der Tabelle und eine angepasste Analyse.
- Durch die Verwendung von doppelt verketteten Listen.

Weitere freiwillige Erweiterungen des Projektes müssen mit der Lehrperson abgesprochen sein, ansonsten werden Punkte abgezogen.

10. Abgabe

Umfang: Eine gezippte Datei mit dem Namen der beiden Gruppenmitglieder in der folgenden Form:

Sort_Nachname1_Vorname1_Nachname2_Vorname2.zip

welche die folgenden Dateien enthält:

- Code in einer einzigen cpp-Datei (ListenSortierung.cpp)
- Ausführbares funktionierendes Programm (ListenSortierung.exe)
- Resultate (Tabelle und Grafik) inkl. Analyse in einer pdf-Datei (Resultat.pdf)

Art: die gezippte Datei auf der Klassenseite (Sharepoint) in den Ordner «M411 Projektarbeit» unterhalb der «Ablagebox» kopieren.

ACHTUNG: Mailen der zip-Datei funktioniert nicht!

Da diese eine exe-Dateien enthält, entfernt der Virenschutz aus Sicherheitsgründen die angefügte zip-Datei aus dem Mail.

Termin: bis spätestens Freitag, 11 Januar 2019 um 23:59 Uhr.

Datum und Zeit der Speicherung gilt!

Nicht termingerechte Abgaben werden mit einer Note 1 bewertet!