1. Übung

1. Aufgabe

Abgabe: 16.05.2024, 10 Uhr im Moodle als zip-Datei

Diese Übung enthält ein paar einfache Eingewöhnungsaufgaben für C/C++. Verwenden Sie können die C++-Elemente benutzen, die Sie bereits kennengelernt haben (z.B. Ein- und Ausgabe). Außerdem enthält die Aufgabe eine sehr einfache Modularisierung. Die Übung soll ohne Objektorientierung entwickelt werden.

Lernziele

- Wiederholung der Arbeit mit Variablen
- Wiederholung von Kontrollstrukturen
- Wiederholung von Funktionen
- Wiederholung dynamische Speicherallokierung
- Modularisierung und Arbeiten mit Makefiles

Modularisierung

Das Programm soll aus den folgenden Modulen bestehen:

- Random
- Mathe
- Arrays
- Dialog

Entwickeln Sie zu allen Modulen cpp- und h-Dateien mit den unten beschriebenen Funktionen.

Modul Mathe

Das Modul enthält die folgenden Funktionen:

- long berechneEndlichesProdukt(int start, int ende)
- long berechneTeilersumme(int zahl)

Endliches Produkt Das endliche Produkt ist eine mathematische Funktion, bei der natürliche Zahlen von einem Startwert bis zu einem Endwert multipliziert werden. Ist einer der beiden Werte kleiner oder gleich 0, ist das Ergebnis 0. In diesem Fall muss die Berechnung überhaupt nicht durchgeführt werden. Ist der Endwert kleiner als der Startwert, sollen die beiden Werte getauscht werden.

Beispiele:
s = 3, e = 4, Ergebnis: 12
s = -4, e = 4, Ergebnis: 0
s = 6, e = 4, Ergebnis = 4 * 5 * 6 = 120

Teilersumme Die Methode berechneTeilersumme(long zahl) berechnet die Teilersumme einer natürlichen Zahl. Wenn die Übergabe keine natürliche Zahl ist, soll die Funktion -1 zurückgeben.

Beispiel: teilersumme(6) = 1 + 2 + 3 + 6 = 12

Random

Dieses Modul enthält Funktionen zum Erzeugen von Zufallswerten mit den folgenden Funktionen:

- void initGenerator()
- int erzeugeInt(int maxWert)

Sie können den Zufallszahlengenerator implementieren, wie Sie möchten, aber im einfachsten Fall gehen Sie folgendermaßen vor. Binden Sie in diesem Modul die folgenden Standardbibliotheken ein:

- cstdlib
- ctime

initGenerator() initialisiert den Zufallszahlengenerator mit der folgenden Anweisung
std::srand(std::time(0)); Diese Funktion darf in Ihrem Programm nur ein einziges Mal aufgerufen werden.
Die Funktion erzeugeInt() erzeugt einen zufälligen int-Wert und gibt diesen zurück. Wenn ein maxWert > 0 eingegeben wird, werden Zufallszahlen $>= 0$ und $<=$ maxZahl erzeugt. Ist maxWert $<= 0$, werden Zufallszahlen ohne Begrenzung erzeugt. Eine Zufallszahl mit Begrenzung erzeugen Sie mittels zahl = std::rand() % maxZahl; Für eine Zufallszahl ohne Begrenzung nach oben lassen Sie den Modulo-Operator weg.

Modul Arrays

Dieses Modul bietet Funktionen zur Bearbeitung von Arrays an:

- int* fuelleArray(int laenge, int maxZahl)
- void gibAusArray(int* array, int anzahlWerte)
- double berechneDurchschnitt(int* zahlen, int anzahlWerte)
- int bestimmeMaximum(int * zahlen, int anzahlWerte)
- int bestimmeMinimum(int * zahlen, int anzahlWerte)

fuelleArray Reserviert dynamisch Speicher für ein Array der Länge laenge und gibt einen Pointer auf dieses Array zurück. Der Parameter maxZahl gibt eine Obergrenze für die Zufallszahlen an. Ist maxWert $<= 0$, werden Zufallszahlen ohne Begrenzung erzeugt. Dazu ruft fuelleArray() die entsprechenden Funktionen aus dem Modul Random auf.

gibArrayAus In dieser Funktion reicht die Ausgabe des Arrayinhalts mit cout.
Die übrigen Funktionen sollten selbsterklärend sein.

Modul Dialog

Schreiben Sie einen einfachen Dialog, mit dem der Benutzer das Programm steuern kann. Der Dialog soll in etwa so aussehen:

```
1: Array mit Zufallswerten anlegen;
2: Array anzeigen;
3: Durchschnitt berechnen;
4: Minimum bestimmen;
5: Maximum bestimmen;
6: Teilersumme berechnen;
7: Endliches Produkt berechnen;
0: beenden -> 1
```

```
Wie viele Werte soll das Array enthalten?10

Geben Sie eine Obergrenze fuer die Zufallswerte an (0 = keine Obergrenze):100
Array wurde erzeugt

1: Array mit Zufallswerten anlegen;
2: Array anzeigen;
3: Durchschnitt berechnen;
4: Minimum bestimmen;
5: Maximum bestimmen;
6: Teilersumme berechnen;
7: Endliches Produkt berechnen;
0: beenden -> 3
Durchschnitt: 49.7

1: Array mit Zufallswerten anlegen;
2: Array anzeigen;
3: Durchschnitt berechnen;
4: Minimum bestimmen;
5: Maximum bestimmen;
6: Teilersumme berechnen;
7: Endliches Produkt berechnen;
0: beenden -> 4
Minimum: 0

1: Array mit Zufallswerten anlegen;
2: Array anzeigen;
3: Durchschnitt berechnen;
4: Minimum bestimmen;
5: Maximum bestimmen;
6: Teilersumme berechnen;
7: Endliches Produkt berechnen;
0: beenden -> 2
41
67
34
0
69
24
78
58
62
64

1: Array mit Zufallswerten anlegen;
2: Array anzeigen;
3: Durchschnitt berechnen;
4: Minimum bestimmen;
5: Maximum bestimmen;
6: Teilersumme berechnen;
7: Endliches Produkt berechnen;
0: beenden ->
```

Makefile

Erstellen Sie ein Makefile, das die Module miteinander verbindet und ein lauffähiges Programm erzeugt.