

C/C++ Übungsblatt 6 (Block 2)

Prof. Dr. Klaus Obermayer und Mitarbeiter

Präprozessor und von C nach C++

Erinnerung: Bitte denken Sie an die Bearbeitung des ersten Block-Tests.

Verfügbar ab:	28.11.2022
Abgabe bis:	04.12.2022

Aufgabe 1: Funktionen als Makros

3 Punkte

Vervollständigen Sie im folgenden Quellcode drei Funktionen, die mit Hilfe von Makros realisiert werden sollen. Die Funktionen sollen dabei folgendes tun:

- `KM_TO_MILES(km)` rechnet die **double**-Variable `km` von Kilometer in Meilen um.
- `LINE(c)` gibt auf der Konsole 10 mal den **char** `c` aus und fügt am Ende einen Zeilenumbruch hinzu.
- `MOD2(zahl)` Berechnet das Ergebnis von der Ganzzahl `zahl` modulo 2.

Erstellen Sie für die Lösung keine anderen Hilfsfunktionen. Es sind für die Makroimplementierung keine Funktionsaufrufe, außer der Funktion `printf` der Bibliothek `<stdio.h>` zugelassen. Fügen Sie nur an den markierten Stellen ihren Code hinzu. Lösungen, die die Anforderungen verletzen, werden mit 0 Punkten gewertet.

Nutzen Sie die folgende Vorgabe (die auch auf ISIS zur Verfügung steht):

```
1 #include <stdio.h>
2
3 #define KM_TO_MILES(km) /*HIER CODE EINFUEGEN*/
4
5
6 #define LINE(c) /*HIER CODE EINFUEGEN*/
7
8
9 #define MOD2(zahl) /*HIER CODE EINFUEGEN*/
10
11
12 int main()
13 {
14     printf("Der Erdumfang betraegt rund %.2f Meilen.\n", KM_TO_MILES(40074));
15     LINE('-');
16
17     int zahl = 255;
18     printf("%d mod 2 ist %d.\n", zahl, MOD2(zahl));
19     printf("%d mod 2 ist %d.\n", zahl+1, MOD2(zahl+1));
20     printf("%d mod 2 ist %d.\n", zahl+2, MOD2(zahl+2));
21     printf("%d mod 2 ist %d.\n", zahl+3, MOD2(zahl+3));
22     LINE('-');
23
24 }
```

Aufgabe 2: Addieren in C++**2 Punkte**

Gegeben ist das folgende C-Programm, welches auch ein gültiges C++-Programm ist.

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("--- Programm zum Addieren zweier Zahlen ---\n");
6
7     double sum1;
8     double sum2;
9     printf("Bitte gib den ersten Faktor ein: ");
10    scanf("%lf", &sum1);
11    printf("Bitte gib den zweiten Faktor ein: ");
12    scanf("%lf", &sum2);
13    printf("Das Produkt ist %g\n", sum1 + sum2);
14
15    return 0;
16 }
```

Erstellen Sie ein für C++ typisches Programm mit der gleichen Funktionalität wie das oben gegebene C-Programm. Ersetzen Sie hierzu die Bibliothek `stdio` durch die C++-Bibliothek `iostream`. Verwenden Sie den Namensraum `std` in der `main`-Funktion implizit, indem Sie die `using namespace`-Anweisung verwenden.

Aufgabe 3: Referenzen**2 Punkte**

Gegeben ist das folgende C++-Programm.

```
1 #include <iostream>
2
3 int main()
4 {
5     int x;
6     int y;
7     int &r = x;
8     int *p = &r;
9     int *q = &x;
10
11     x = 30;
12     y = 17;
13
14     x *= 4;
15     y = r;
16     *p = 10;
17
18     return 0;
19 }
```

Füllen Sie die folgende Tabelle mit den Werten der Variablen nach dem Ausführen der entsprechenden Zeile.

Hinweis: In den Spalten `*p` und `*q` sind die Inhalte der, durch die Zeiger `p` und `q`, referenzierten Adressen gefragt (Dereferenzierung).

Zeile	x	y	r	*p	*q
12	30	17	30	30	30
14					
15					
16					

Aufgabe 4: Speicherverwaltung und Namespace

3 Punkte

Gegeben ist ein unvollständiges C++-Programm in der Datei `messung.cpp`. Das Programm soll den Benutzer nach einer Anzahl von Temperaturmessungen an verschiedenen Orten fragen. Danach müssen per Hand die Temperaturen und die Ortsnamen eingegeben werden. Abschließend soll

- die Durchschnittstemperatur

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- und die Standardabweichung

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

ausgegeben werden.

Ergänzen Sie die `main`-Funktion so, dass diese in geeigneter Weise die anderen Funktionen verwendet. Weiterhin soll sie in geeigneter Weise Heapspeicher reservieren und wieder freigeben. Verwenden Sie hierfür die Operatoren `new` und `delete`.

Ergänzen Sie die anderen Funktionen so, dass die geforderte Funktionalität entsteht.

Hinweis: Die Bibliothek `cmath` stellt die Funktion `double sqrt (double x)` zum Berechnen von Quadratwurzeln bereit.

Vorgabe (auch auf ISIS zu finden):

Listing 1: `messung.cpp`

```

1 #include <iostream>
2 #include <iomanip> // fuer std::setw(int)
3 #include <string>
4 #include <cmath>
5
6 /*
7  * Print-Funktionen fuer float messwerte[] und std::string orte[]
8  */
9 namespace printer {
10     void printArray(float *messwerte, std::string *orte, int anzahl);
11     void printDurchschnitt(float *messwerte, int anzahl);
12     void printAbweichung(float *messwerte, int anzahl);
13
14     void printArray(float *messwerte, std::string *orte, int anzahl)
15     {
16         std::cout << "Alle Messwerte:" << std::endl;
17         for(int i = 0; i < anzahl; ++i) {
18             std::cout << std::setw(3) << i+1 << " ) " << messwerte[i] << " Grad
19 Celsius in " << orte[i] << std::endl;
20         }
21     }

```

```
21
22 void printDurchschnitt(float *messwerte, int anzahl)
23 {
24     // TODO: Durchschnittstemperatur auf der Konsole ausgeben
25
26
27 }
28
29 void printAbweichung(float *messwerte, int anzahl)
30 {
31     // TODO: Abweichung auf der Konsole ausgeben
32
33
34 }
35
36 }
37
38 using namespace std;
39
40 /*
41  * Programm zum Speichern von Temperaturmessungen (Wert und Ort)
42  * Fragt den Nutzer nach Anzahl der Daten und dann nach den Daten selbst.
43  *
44  * Zusatzinformationen fuer neugierige Studenten:
45  * - std::setw(int n)
46  *     Fuer Breitenangabe in Streams
47  *     (http://en.cppreference.com/w/cpp/io/manip/setw)
48  */
49 int main(void)
50 {
51     cout << "Programm zum Speichern von Temperaturmesswerten und zugehoerigen
52     Orten." << endl;
53
54     int anzahl;
55     float *messwerte;
56     std::string *orte;
57
58     // Anzahl einlesen
59     do {
60         cout << "Bitte geben Sie die Anzahl (zwischen 1 und 999) der Messwerte
61         ein: ";
62         cin >> anzahl;
63         cin.clear(); // setzt den Zustand vom cin-stream zurueck
64         cin.ignore(1000, '\n'); // leer den cin-stream
65     } while ( !(1 <= anzahl && anzahl <= 999) );
66
67     // TODO: Heapspeicher fuer Messwerte allozieren.
68
69
70     // Messwerte setzen
71     cout << "Bitte geben Sie nun abwechselnd die Messwerte (als Kommazahl, z.B.
72     13.7) und Orte (z.B. Berlin) ein." << endl;
73     for (int i = 0; i < anzahl; ++i) {
74         cout << setw(3) << i+1 << ") Messwert: ";
75         cin >> messwerte[i];
```

```
75 |  
76 |     cout << setw(3) << i+1 << " )      Ort: ";  
77 |     cin >> ws; // alle white spaces entfernen  
78 |     getline(cin, orte[i]);  
79 | }  
80 |  
81 | // TODO: Konsolenausgaben durchfuehren  
82 |  
83 |  
84 |  
85 | // TODO: Reservierten Speicher freigeben.  
86 |  
87 |  
88 |  
89 | return 0;  
90 | }
```