

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 9

Aufgabe 9.1. Schreiben Sie ein Programm, welches ein Wort einliest und überprüft ob es sich bei dem eingegeben Wort um ein Palindrom handelt. Ein Palindrom ist ein Wort, welches von vorne und hinten gelesen gleich lautet, z.B.: Anna, Otto, Reliefpfeiler. Verwenden Sie in dem Beispiel den Datentyp string. Speichern Sie den Source-Code unter `palindrom.cpp` in das Verzeichnis `serie09`.

Hinweis: Am Besten lagern Sie die Kernfunktionalität in eine Funktion `bool isPalindrome(string wort)` aus.

Aufgabe 9.2. Was gibt folgender Code am Bildschirm aus und warum?

```
#include <iostream>
#include <string>
using namespace std;

class T1 {
    string t1;
public:
    T1(string val) { cout << "Ich bin Konstruktor von " << val << endl; t1=val; }
    T1() { cout << "Ich bin Konstruktor von default" << endl; t1="default"; }
    ~T1() { cout << "Ich bin Destruktor von " << t1 << endl; }
};

int main() {
    T1 bert("bert");
    T1 bob;
    T1 def("bob");
    return 0;
}
```

Aufgabe 9.3. Implementieren Sie die Zugriffsfunktionen der Klasse Bruch, welche durch

```
class Bruch {
    long zaehler;
    unsigned long nenner;
public:
    Bruch();
    Bruch(long zaehler, unsigned long nenner);
    setZahler(long z);
    setNenner(unsigned long n);
    double getWert();
};
```

gegeben ist. Die Methode `getWert` soll dabei den Dezimalwert des Bruchs zurückgeben. Achten Sie bei den `set`-Funktionen auf eventuelle Sicherheitsabfragen. Implementieren Sie auch den Konstruktor `Bruch()`, welcher den Zähler auf 0 und den Nenner auf 1 setzen soll. Speichern Sie den Source-Code unter `bruch.cpp` in das Verzeichnis `serie09`.

Aufgabe 9.4. Schreiben Sie eine Funktion `unique`, die einen Vektor $x \in \mathbb{R}^n$ aufsteigend sortiert, doppelte Einträge streicht und den Vektor in gekürzter Form zurückgibt. Die Funktion soll also beispielsweise den Vektor $x = (4, 3, 5, 1, 4, 3, 4) \in \mathbb{R}^7$ durch den Vektor $x = (1, 3, 4, 5) \in \mathbb{R}^4$ überschreiben. Verwenden Sie hierfür die `Vector`-Bibliothek aus der Vorlesung. Speichern Sie den Source-Code unter `unique.c` in das Verzeichnis `serie09`.

Aufgabe 9.5. Die Standardeingabe `cin` in C++ liest eine Eingabe, wie Sie wissen, nur bis zum ersten Trennzeichen. Schreiben Sie eine Funktion `myFullName`, die Ihren Vor- und Nachnamen über die Tastatur einliest und jeweils in einem string abspeichert. Diese beiden Strings sollen nun in einem einzelnen String gespeichert werden, der schließlich ausgegeben wird. Kennen Sie eine Möglichkeit um längere Tastatureingaben einzulesen? Speichern Sie den Source-Code unter `MyFullName.cpp` in das Verzeichnis `serie09`.

Aufgabe 9.6. Schreiben Sie eine Klasse `Matrix` zur Speicherung von Matrizen der Größe $m \times n$. Die Einträge sollen hierbei als langer `double*`-Vektor der Länge mn gespeichert werden. Schreiben Sie Methoden, welche die Möglichkeit bieten Einträge zu erstellen, bzw. auszulesen. Achten Sie hierbei auf eventuelle Sicherheitsabfragen. Schreiben Sie einen Konstruktor, der bei übergebenen Werten $a, b \in \mathbb{N}$ eine Nullmatrix der Größe $a \times b$ erzeugt, sowie einen Standardkonstruktor, der eine leere Matrix der Größe 0×0 generiert. Natürlich darf auch ein entsprechender Destruktor nicht fehlen. Orientieren Sie sich hierbei an Folie 230 der Vorlesung. Speichern Sie den Source-Code unter `matrix.cpp` in das Verzeichnis `serie09`.

Aufgabe 9.7. Schreiben Sie eine Funktion `matmult` welche das Produkt zweier Matrizen $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times q}$ bestimmt. Die Funktion soll zwei Matrizen übernehmen und das Produkt zurückgeben. Überprüfen Sie auch, ob die Dimensionen der Matrizen zusammenpassen! Verwenden Sie die Klasse aus Aufgabe 9.6. Speichern Sie den Source-Code unter `matmult.cpp` in das Verzeichnis `serie09`.

Aufgabe 9.8. Schreiben Sie ein `Makefile` für die aktuelle Übungsserie. Dieses soll zumindest folgende Dinge umfassen:

- Erzeugen von Programmen aller von Ihnen gelösten Aufgaben.
- Ein Beispiel, aus dem Sie zwei verschiedene Programme generieren ohne(!) die Kernaufgabe in zwei verschiedene Dateien zu kopieren. (Sie könnten beispielsweise ein Programm generieren, das ausgibt, ob das Wort „Lagernebenregal“ ein Palindrom ist und ein zweites Programm, das ein Wort von der Tastatur einliest und selbiges überprüft. Die Funktion `isPalindrome` soll aber nur in einer einzigen Datei enthalten sein.)

Welche Vorteile hat es keine Kopien Ihrer Funktionen zu erstellen?

- Das Generieren einer dynamischen Bibliothek und deren Verwendung in einem Programm. Was passiert, wenn Sie Ihre dynamische Bibliothek `myLib.so` nach dem Kompilieren umbenennen?