Dirk Praetorius,
Thomas Führer

Sommersemester 2014
20.05.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 10

Aufgabe 10.1. Schreiben Sie eine Funktion `unique`, welche einen Vektor sortiert, die doppelten Einträge entfernt, und den gekürzten Vektor zurückliefert. Verwenden Sie hierzu den C++-Standardcontainer `vector`. Erklären Sie die Unterschiede zu Ihrer ersten Implementierung. Speichern Sie den Source-Code unter `unique.cpp` in das Verzeichnis `serie10`.

Aufgabe 10.2. Implementieren Sie ein einfaches *Tic Tac Toe* Spiel. Falls Sie Ihnen nicht bekannt sind, können Sie die Regeln unter `http://de.wikipedia.org/wiki/Tic_Tac_Toe` nachlesen. Das Spiel soll mindestens die folgenden Kriterien erfüllen:

1. Es sollen zwei Spieler gegeneinander antreten können die jeweils abwechselnd am Zug sind.
2. Es soll automatisch erkannt werden, wann ein Spieler gewonnen hat.
3. Nach jedem Zug soll das aktuelle Spielfeld grafisch in der Konsole ausgegeben werden.
4. Kann sich am Ende keiner der Spieler durchsetzen, so soll `'unentschieden'` ausgegeben werden.

Verwenden Sie den Container `vector` für das Spielfeld, also etwa `vector<vector<char> >`. Jeder Spieler soll ein eigenes Zeichen verwenden, z.B.: `'x'` für Spieler 1 und `'o'` für Spieler 2. Das Spielfeld soll dabei ein $n \times n$ Feld mit $n \geq 3$ sein.

Aufgabe 10.3. Der Verschlüsselungsalgorithmus ROT13 ersetzt jeden Buchstaben der Eingabe zyklisch durch den im Alphabet um 13 Buchstaben weiter hinten (oder vorne) stehenden Buchstaben. Die Buchstaben A, B, C, ... werden auf N, O, P, ... abgebildet, die Buchstaben N,O,P, ... auf A, B, C, ... Setzen Sie diesen Algorithmus in der Funktion `rot13` um. Ihre Funktion braucht nur für Großbuchstaben zu funktionieren.
Schreiben Sie nun eine weitere Funktion `rot_n`, der Sie zusätzlich übergeben können, um wieviele Stellen die Eingabe rotiert werden soll. Überlegen Sie sich, wie man mit `rot_n` verschlüsselte Texte entschlüsseln kann. Schreiben Sie dazu eine Funktion `unrot_n`.
Hinweis: Ein Zeichen ist vom Typ `char` und wird intern durch eine Zahl dargestellt, `'A'` beispielsweise durch 65. Ziehen Sie von einem Buchstaben `'A'`, ..., `'Z'` den Buchstaben `'A'` ab, so ist das Ergebnis intern eine Zahl zwischen 0 und 25. Führen Sie dort die notwendigen Rechenoperationen durch und addieren Sie anschließend wieder den Buchstaben `'A'` hinzu! Weitere Informationen zu ASCII finden Sie auf Wikipedia. Speichern Sie den Source-Code unter `rot.{hpp,cpp}` in das Verzeichnis `serie10`.

Aufgabe 10.4. 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 $m, n \in \mathbb{N}$ eine Nullmatrix der Größe $m \times n$ erzeugt, sowie einen Standardkonstruktor, der eine leere Matrix der Größe 0×0 generiert. Natürlich darf auch ein entsprechender Destruktor nicht fehlen. Speichern Sie den Source-Code unter `matrix.{hpp/cpp}` in das Verzeichnis `serie10`. *Hinweis:* Verwenden Sie `new` und `delete[]` um Speicher zu allokieren bzw. freizugeben. Achten Sie auch auf die "Dreierregel".

Aufgabe 10.5. Schreiben Sie ein `Makefile` für die aktuelle Übungsserie. Dieses soll zumindest folgende Dinge umfassen:

- Erzeugen von Programmen aller von Ihnen gelösten Aufgaben.
- Das Generieren einer dynamischen Bibliothek und deren Verwendung in einem Programm.

Aufgabe 10.6. Was tut der folgende C++ Code? Welche Funktionalität haben die Funktionen `func1` und `func2`? Wie sieht die Bildschirmausgabe aus?

```
#include <iostream>
#include <vector>

using namespace std;

void func2(vector<double> &dp, int mp);
void func1(vector<double> &dp) {
   int mp = dp.size();
   func2(dp,mp);

   for (int i = mp-1; i>=1; i--){
      for(int j = 0; j<i; j++){
         if (dp[j] > dp[j+1])
            swap(dp[j], dp[j+1]); //Vertausche dp[j] und dp[j+1]
      }
      func2(dp,mp);
   }
}

void func2(vector<double> &dp, int mp){
   for (int k = 0; k<= mp-1; k++){
      cout << dp[k] << endl;
   }
}

int main(){
   vector<double> a(4,0);
   a[0] = 14;
   a[1] = 12;
   a[2] = 7;
   a[3] = 4;
   func1(a);
}
```

Aufgabe 10.7. 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 10.8. Erklären Sie den Unterschied zwischen Referenzen und Pointern mit eigenen Worten. Schreiben Sie exemplarisch einen Code, welcher die Inhalte zweier Variablen vertauscht, einmal mit Pointern und einmal mit Referenzen. Welche Vorteile bringt die Verwendung von Referenzen gegenüber Pointern mit sich? Welche Nachteile?