

3 Erste Programme

3.1 Übungsaufgabe

Verwenden Sie für jede Aufgabe die C++ IDE von Eclipse (pro Aufgabe ein Projekt).

3.1.1 Aufgabe 1

Schreiben Sie ein C++ Programm, das einen String einliest und testet, ob der erste und der letzte Buchstabe eines Strings identisch ist.

3.1.2 Aufgabe 2

Schreiben Sie ein C++ Programm, das eine Zeile einliest und testet, ob der erste und letzte Buchstabe einer Zeile identisch ist.

3.1.3 Aufgabe 3

Schreiben Sie ein Programm `translate`. Dieses Programm soll eine Textzeile einlesen und alle Großbuchstaben in Kleinbuchstaben und umgekehrt konvertieren und ausgeben.

3.1.4 Aufgabe 4

Was passiert, wenn Sie in das Programm aus Aufgabe 3 einen Text mit Umlauten eingeben?

3.1.5 Aufgabe 5

Schreiben Sie das Programm `read.cxx` in Beispiel 3.14 aus dem Vorlesungsskript ab und testen Sie verschiedene Eingaben:

- Eingabe einer Zahl
- Eingabe von mehreren Zahlen
- Eingabe eines Wortes (ohne Umlaute)
- Eingabe von zwei Worten (ohne Umlaute)
- Eingabe von Umlauten
- Eingabe einer ganzen Zeile (mit Umlauten)

3.2 Wiederholung und zusätzliche Informationen

3.2.1 Einlesen und Ausgeben von Strings

Zum Einlesen und Ausgeben von Daten gibt es in C++ 3 Streams. Diese lesen alle Zeichen bis zum `<return>`.

- Einlesen von Daten: `cin` mit dem Operator `>`
- Ausgeben von Daten: `cout` mit dem Operator `<`
- Fehlermeldungen: `cerr` mit dem Operator `<`

Ganze Zeilen können außerdem mit der Methode `getline()` eingelesen werden. `getline()` liest alle Zeichen bis zum `newline` Zeichen. Dieses bleibt auf dem Eingabestrom.

3.2.2 Namespaces

Um zu vermeiden, dass Mehrdeutigkeiten bezüglich der Benennung von Variablen, Funktionen, etc. aufkommen, gibt es verschiedene Namensbereiche, in denen diese eindeutig benannt wurden. Die Standardfunktionen sind im Default-Namespace `std` deklariert.

Möchte man also die Funktionen oder Klassen des Namespace `std` verwenden, so kann man entweder am Anfang des Programmes `using namespace std;` einstellen, oder `std::` vor die einzelnen benutzten Funktionen oder Klassen stellen.

```
using namespace std;
string name;
cout << "Hallo";

oder

std::string name;
std::cout << "Hallo";
```

3.2.3 Hilfreiche Shortcuts in Eclipse

- Autocomplete: CTRL + Space
- Mehrere Zeilen auskommentieren: CTRL + /
- Format Code: CTRL + SHIFT + F
- aktuelle Zeile löschen: CTRL + D

3.2.4 Fehlermeldungen in Eclipse

Fehlermeldungen in Eclipse sind eine große Hilfestellung auf dem Weg zum funktionierenden Programm. Meistens kann man das Problem gut aus der Fehlermeldung entnehmen. Häufig ist es auch sinnvoll nach Fehlermeldungen im Internet zu suchen, da es den Fehler immer schon einmal gab!

- Gelber Käfer: meldet Fehler, die aber keine Exception werfen
- Roter Käfer: meldet Fehler, die das Programm zum Absturz bringen oder das Kompilieren verhindern
- Rotes Kreuz: meldet fatale Fehler
- Gelbes Fragezeichen: gibt Vorschläge für syntaktische Fehler

Typische Fehlermeldungen:

- `Unresolved inclusion: <iostream> - fatal error: Datei oder Verzeichnis nicht gefunden:`
IOStream wurde nicht erkannt, weil ein Leerzeichen zu viel in `<iostream>` steht
- `Missing ;: fehlendes Semikolon`
- `Symbol 'endl' could not be resolved:` der Namespace für `endl` wurde nicht mit `std::endl` oder `using namespace std;` angegeben
- `Invalid Arguments ...:` Die übergebenen Argumente passen nicht, es werden andere Datentypen erwartet
- `Multiple markers at this line - Missing; - Unbounded string encountered:` Es fehlt ein Semikolon oder der String wurde nicht geschlossen
- `No return, in function returning non-void:` return Statement fehlt in Funktion

3.3 Lösungsvorschlag

3.3.1 Aufgabe 1

```
/*Autor: Nicola Greth
 * Uebung 2, Aufgabe 1
 * Programm: testet ob erster und letzter Buchstabe eines Strings identisch sind
 */
#include <iostream>
#include <string>

int main() {
    std::string meinString;
    std::cout << "Bitte geben Sie eine Zeichenkette ein:>>> " << std::endl;
```

```
std::cin >> meinString;

if (meinString[0] == meinString[meinString.length()-1]) {
    std::cout << "Der erste und der letzte Buchstabe Ihrer Zeichenkette
    sind identisch." << std::endl;
}

else {
    std::cout << "Der erste und der letzte Buchstabe Ihrer Zeichenkette
    sind nicht identisch." << std::endl;
}

return 0;
}
```

3.3.2 Aufgabe 2

```
/*Autor: Nicola Greth
 * Uebung 2, Aufgabe 2
 * Programm: testet ob erster und letzter Buchstabe einer Zeile identisch sind
 */
#include <iostream>
#include <string>

int main() {
    std::string meineZeile;
    std::cout << "Bitte geben Sie eine Zeile ein:>>> " << std::endl;
    getline(std::cin, meineZeile); //getline liest alle Zeichen bis zum
    NewlineZeichen

    if (meineZeile[0] == meineZeile[meineZeile.length()-1]) {
        std::cout << "Der erste und der letzte Buchstabe Ihrer Zeile sind
        identisch." << std::endl;
    }

    else {
        std::cout << "Der erste und der letzte Buchstabe Ihrer Zeile sind
        nicht identisch." << std::endl;
    }

    return 0;
}
```

3.3.3 Aufgabe 3

```
/*Autor: Nicola Greth
 * Uebung 2, Aufgabe 3
 * Programm: liest eine Zeile und konvertiert Groß- und Kleinbuchstaben
 */
#include <iostream>
#include <string>

int main() {
```

```

std::string meineZeile;
std::cout << "Bitte geben Sie eine Zeile ein:>>> " << std::endl;
getline(std::cin, meineZeile);

for (int i = 0; i < meineZeile.length(); i++) {
    if (isupper(meineZeile[i])) { //wenn es ein Großbuchstabe ist
        meineZeile[i] = tolower(meineZeile[i]); //mach einen
        Kleinbuchstaben daraus
    }

    else { //wenn nicht
        meineZeile[i] = toupper(meineZeile[i]); //mach einen
        Großbuchstaben daraus
    }
}

std::cout << "Das ist Ihre neue Zeile:>>>" << meineZeile << std::endl;
return 0;
}

```

3.3.4 Aufgabe 4

Die Umlaute werden nicht konvertiert.

3.3.5 Aufgabe 5

```

/*Autor: Nicola Greth
 * Uebung 2, Aufgabe 5
 * Programm: read.cpp
 */
#include <iostream>
#include <string>
using namespace std;

int main () {
    char zeichen;

    while (cin.get( zeichen ))
        cout .put ( zeichen );
    return 0;
}

```

Tests der verschiedenen Eingaben:

- Eingabe einer Zahl: Programm gibt die eingetippte Zahl wieder aus
- Eingabe von mehreren Zahlen: Programm gibt die eingetippten Zahlen wieder aus
- Eingabe eines Wortes (ohne Umlaute): Programm gibt das Wort wieder aus
- Eingabe von zwei Worten (ohne Umlaute): Programm gibt die Wörter wieder aus

-
- Eingabe von Umlauten: Umlaute können nicht dargestellt werden
 - Eingabe einer ganzen Zeile (mit Umlauten): Umlaute können nicht dargestellt werden