6 UTF8 und Stringverarbeitung

6.1 Übungsaufgabe

6.1.1 Aufgabe 1

Verwenden Sie die Musterlösung aus Übung 3-2 und machen Sie das Programm unicode-fähig. Arbeiten Sie mit `wcin`, `wcout`, `wstrings`, `iswpunct`, `locale`, etc.

6.1.2 Aufgabe2

Verwenden Sie den folgenden Text: <http://www.zeno.org/Literatur/M/Dostoevskij,+F%C3%ABdor+Michajlovi%C4%8D/Romane/Der+Idiot>

Verwenden Sie Ihre Lösung aus Aufgabe 1, um jede Zeile der Datei `der_idiot_100.txt` zu lesen, auszugeben und in der nächsten Zeile die Positionen aller Punktationszeichen dieser Zeile anzuzeigen.

z.B.

```
Morgen, tobte der Mann, ist endgültig klar, ob Bayern weiterkommt!
```

6.1.3 Aufgabe 3

Schreiben Sie ein C++ Programm, welches statistische Angaben des Textfiles `der_idiot_100.txt` (UTF8) ermittelt. Geben Sie die Anzahl der Zeilen, Wörter und Buchstaben aus.

6.2 Wiederholung und zusätzliche Informationen

6.2.1 Isolatint Dateien in Eclipse importieren

- Neue File erstellen und entsprechend benennen z.B. "der_idiot_100.txt"
- Text Copy und Pasten, aber dabei beachten dass der korrekte Text (mit sichtbaren Umlauten) kopiert wird
- Codierung der File korrekt einstellen über Rechtsklick - Properties - TextfileEncoding - Für Isolatint Datei auf ISO-8859-1 stellen

- Wenn die Umlaute in der Eclipse Datei wieder zerstört sind, ist die Datei richtig auf Isolatın kodiert

6.2.2 UTF8 und C++

Um Unicode-Zeichen mit C++ verarbeiten zu können, wurden neue Datentypen eingeführt. `wstring` ist der äquivalente Unicode-Datentyp für `strings`, `wchar_t` ist der äquivalente Unicode-Datentyp für `wchar_ts`. Aus den normalen Strings/Characters wird durch das Voranstellen von „L“ ein `Widestring/ Widecharacter`.

Ebenso gibt es für die meisten Funktionen äquivalente `Widestring/Widecharacter`-Funktionen, z.B. `cin` wird zu `wcin`.

Da es nicht sinnvoll ist, normale Funktionen und `Widestring/Widecharacter`-Funktionen zu mischen, kann es in Programmen zu Problemen kommen, bei denen Dateinamen vom User eingegeben werden können. Dateinamen werden dem `wifstream` als `Cstring` übergeben. `Wstrings` können aber nur über einen kleinen Umweg in `Cstrings` konvertiert werden:

```
//Dateiname wird als wstring eingelesen
wstring filename;
wcout << "Bitte geben Sie den Dateinamen ein: " << endl;
getline(wcin, filename);

//Dateiname wird von wstring in string konvertiert
string filenameString(filename.begin(), filename.end());

//Datei kann nun normal mit der c_str() Methode für Strings geöffnet werden
wifstream datei(filenameString.c_str());
datei.imbue(locale("de_DE.utf-8"));
```

6.3 Lösungsvorschlag

6.3.1 Aufgabe 1

```
/*Autor: Nicola Greth
 * Uebung 5, Aufgabe 1
 * Programm: Zeigt Punctuationszeichen an - UNICODE-Version
 */
#include <iostream>
#include <fstream>

int main() {
    //Kodierung wird auf utf8 gesetzt
    std::locale utf8locale("de_DE.UTF-8");
    setlocale(LC_ALL, "de_DE.UTF-8");
    std::wstring zeile;
```

```

//Textzeile einlesen
std::wcout << L"Geben Sie eine Textzeile ein>>";
getline(std::wcin, zeile);

//Gibt die unbearbeitete Zeile aus
std::wcout << zeile << std::endl;
int count_punct = 0;

for (int i = 0; i < zeile.size(); i++) {
    //Wenn es ein Piktuationszeichen ist soll die Position markiert werden
    if (iswpunct(zeile.at(i))) {
        std::wcout << "^";
        count_punct++;
    }

    //Wenn es kein Piktuationszeichen ist soll es eine Stelle weiter rücken
    else {
        std::wcout << " ";
    }
}

std::wcout << std::endl;

//Wenn es gar kein Piktuationszeichen gab
if (!count_punct) {
    std::wcout << L"Kein Piktuationszeichen in dieser Zeile gefunden!"
    << std::endl;
}
return 0;
}

```

6.3.2 Aufgabe 2

```

/*Autor: Nicola Greth
 * Uebung 5, Aufgabe 2
 * Programm: Zeigt Piktuationszeichen in Datei an - UNICODE-Version
 */
#include <iostream>
#include <fstream>

int main() {
    //Kodierung wird auf utf8 gesetzt
    std::locale utf8locale("de_DE.UTF-8");
    setlocale(LC_ALL, "de_DE.UTF-8");
    std::wstring zeile;

    //Datei wird geöffnet und korrekt kodiert
    std::string filename = "der_idiot_100.txt";
    std::wifstream input;
    input.open(filename.c_str());
    input.imbue(utf8locale);
}

```

```
while (getline(input, zeile)) {
    //Gibt die unbearbeitete Zeile aus
    std::wcout << zeile << std::endl;

    //Loesung mit Bool statt int
    bool count_punct = false;

    for (int i = 0; i < zeile.size(); i++) {

        //Wenn es ein Punctuationszeichen ist soll die Position markiert werden
        if (iswpunct(zeile.at(i))) {
            std::wcout << "^";
            count_punct = true;
        }

        //Wenn es kein Punctuationszeichen ist soll es eine Stelle weiter rücken
        else {
            std::wcout << " ";
        }
    }

    std::wcout << std::endl;

    //Wenn es gar kein Punctuationszeichen gab
    if (!count_punct) {
        std::wcout << L"Kein Punctuationszeichen in dieser Zeile gefunden!" <<
            std::endl;
    }
}
return 0;
}
```

6.3.3 Aufgabe 3

```
/*Autor: Nicola Greth
 * Uebung 5, Aufgabe 3
 * Programm: Statistik von Dateien (Unicode Version)
 */
#include <iostream>
#include <locale>
#include <fstream>
#include <sstream>

int main() {
    setlocale(LC_ALL, "de_DE.UTF-8");
    std::wstring line;
    std::wifstream datei("der_idiot_100.txt");
    datei.imbue(std::locale("de_DE.utf-8"));

    //Zaehlvariablen werden deklariert und initialisiert
    int countLines = 0;
```

```
int countWords = 0;
int countLetters = 0;

while (getline(datei, line)) {

    //zaehlt Zeilen
    countLines++;

    //zaehlt Buchstaben
    for (int i = 0; i < line.size(); i++) {
        if (!std::isspace(line.at(i))) {
            if (!std::ispunct(line.at(i))) {
                countLetters++;
            }
        }
    }

    //zaehlt Woerter (Version mit stringstream)
    std::wstring word;
    std::wstringstream wordsOfLine;
    wordsOfLine << line;

    while (wordsOfLine >> word) {
        countWords++;
    }
}

datei.close();

//Ergebnisse werden ausgegeben
std::wcout << L"Im Text gibt es " << std::endl;
std::wcout << countLines << L" Zeilen, " << std::endl;
std::wcout << countWords << L" Wörter und " << std::endl;
std::wcout << countLetters << L" Buchstaben." << std::endl;
return 0;
}
```