Streams, Kodierung und Funktionen

C++ Übung am 12. Mai 2016

Wiederholung: Streams

```
#include <iostream> // Input-Output-Stream

  std::cout   // Standardausgabe
  std::cin    // Standardeingabe
  std::cerr   // Standardfehlerausgabe
```

```
#include <fstream>  // File-Stream

  std::ifstream   // Input-File-Stream
  std::ofstream   // Output-File-Stream
```

```
#include <sstream>  // Stringstreams

  std::stringstream
```

Internationalisierunng

```
#include <iostream> // Input-Output-Stream

  std::wcout    // Standardausgabe
  std::wcin     // Standardeingabe
  std::wcerr    // Standardfehlerausgabe
```

```
#include <fstream>  // File-Stream

  std::wifstream   // Input-File-Stream
  std::wofstream   // Output-File-Stream
```

```
#include <sstream>  // Stringstreams

  std::wstringstream
```

Stringstream

```
#include <iostream>
#include <sstream>  // Stringstreams

int main () {
   std::wcout << L"Bitte geben Sie eine Zeile ein!" << std::endl;
   std::wstring myLine;
   getline(std::wcin, myLine);

   std::wstringstream myStream;
   myStream << myLine;

   std::wstring myWord;
   while (myStream >> myWord) {
      std::wcout << myWord << std::endl;
   }
   return 0;
}
```

Wiederholung: Öffnen

```
#include <iostream> // Input-Output-Stream
#include <fstream>  // File-Stream

// Zum Schreiben:

ofstream raus(„ausgabe.txt“);  // oder:

ofstream raus;
raus.open(„ausgabe.txt“);      // oder:

ofstream raus;
string filename = „ausgabe.txt“;
raus.open(filename.c_str());
```

Wiederholung: Kodierung

```
setlocale(LC_ALL, “”);
   // Locale des Systems verwenden
std::locale utf8locale(„de_DE.utf-8“);
   // Eigene Locale erstellen

fstream datei;
datei.imbue(utf8locale);
   // Einfärben einer Datei auf
      bestimmte Locale

wstring name = L“Max Hadersbeck“;
```

Stringfunktionen

`Aufruf allgemein: str.funktion();`

Funktion	Aufrufbeispiel
size()/ length()	`int länge = str.length();`
at()	`char last = str.at(länge - 1);`
swap()	`str1.swap(str2);`
insert()	`str.insert(pos, str2);`
erase()	`str.erase(pos, len);`
replace()	`str.replace(pos, len, str2);`
substr()	`string str2 = str1.substr(pos, len);`
c_str()	`datei.open(filename.c_str());`
find()	`int pos = str1.find(str2, abpos);`
compare()	`int i = str1.compare(str2);`

wichtige Funktionen

Rückgabewert	Funktion
bool	ispunct(char)
bool	iswpunct(wchar_t)
bool	iswalpha(wchar_t)
bool	iswalnum(wchar_t)
bool	iswdigit(wchar_t)
bool	iswspace(wchar_t)

char	tolower(char)
wchar_t	towlower(wchar_t)
char	toupper(char)
wchar_t	towupper(wchar_t)

www.cplusplus.com

Bei Fragen erstmal nachlesen.
Zum Beispiel auf:

http://www.cplusplus.com/reference/string/string/

Funktionen

Deklaration:

Funktionsaufruf:

Funktions-
implementation:

```
#include <iostream>

int addiere(int zahl1, int zahl2);
void druckeSummeAus(int summe);

int main() {
    int zahl1 = 3;
    int zahl2 = 5;
    int summe = 0;
    summe = addiere(zahl1, zahl2);
    druckeSummeAus(summe);
    return 0;
}

int addiere(int zahl1, int zahl2) {
    return zahl1 + zahl2;
}

void druckeSummeAus(int summe) {
    std::cout << "Summe: " << summe << std::endl;
}
```

Call by value/ reference

```
#include <iostream>

void doubleByValue(int zahl);
void doubleByReference(int &zahl);

int main() {
    int meineZahl = 5;
    doubleByValue(meineZahl);
    std::cout <<"meine Zahl: " << meineZahl << std::endl;
    doubleByReference(meineZahl);
    std::cout <<"meine Zahl: " << meineZahl << std::endl;
}

void doubleByValue(int zahl) {  // Wert übergeben
    zahl = zahl * 2 ;
    std::cout<<"Verdoppelte Zahl: " << zahl << std::endl;
}

void doubleByReference(int &zahl) { // Zeiger auf Variable übergeben
    zahl = zahl * 2 ;
    std::cout<<"Verdoppelte Zahl: " << zahl << std::endl;
}
```

Call by value/ reference

```
Ausgabe:

Verdoppelte Zahl: 10
meine Zahl: 5
Verdoppelte Zahl: 10
meine Zahl: 10
```

Hinweise Hausaufgaben

Hinweise Hausaufgaben

Wo muss std:: hin?

- vor allem vor: cin, cout, endl, string, locale

Braucht man endl?

- für \n und flush

Hinweise Hausaufgaben

```
ofstream log;
ifstream password;

string logfilename ("log.txt");
string passwordfilename("mypasswrd.txt");
```

```
log.open(logfilename.c_str());
password.open(passwordfilename.c_str());


// geht kürzer!
```

Hinweise Hausaufgaben

```
ofstream log("log.txt");
ifstream password("mypasswrd.txt");

~~string logfilename ("log.txt");~~
~~string passwordfilename("mypasswrd.txt");~~

~~log.open(logfilename.c_str());~~
~~password.open(passwordfilename.c_str());~~
```

Hinweise Hausaufgaben

```
do {
   getline(cin,test);
   if (test==zeile){
         cout << "Gut gemacht!" << endl;
         break;
   }
if (i>4){
   cout << "Pech gehabt" << endl;
   break;
}
cout << "Versuchs nochmal. Hast noch " << 5-i << "
   Versuche!" << endl;
i++;
} while (1);
```

// diese While-Schleife ist sinnlos, da sie nie abbricht -> Bedingung muss zu Abbruch führen!