Klausur zur Vorlesung „Höhere Programmierung“ im SS 2011

Andreas Neumann M.A.

Datum : 25.07.2011

Vorname:	
Nachname:	
Matrikelnummer:	
Studiengang:	

Die Klausur besteht aus **6 Aufgaben**. Die Punktzahl ist bei jeder Aufgabe angegeben. Die Bearbeitungsdauer beträgt **60 Minuten**. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben.
Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein.

Bewertung

Aufgabe	Erreichte Punktzahl	Mögliche Punktzahl
Aufgabe 1 – IO	11	11
Aufgabe 2 – Frequenzliste	16	16
Aufgabe 3 – Objektorientierte Programmierung	20	20
Aufgabe 4 – Collections, STL	21	21
Aufgabe 5 – Boost	18	18
Aufgabe 6 – Syntaxfehler finden	14	14

Gesamtpunktzahl	100	100	
Note			

Einwilligungserklärung

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses unter Verwendung meiner Matrikelnummer im Internet zu.

_____ (Name, Datum)

Aufgabe 1 – IO

Was geben folgende Programmfragmente auf dem Terminal aus? Sie können davon ausgehen, dass das Terminal UTF-8-fähig ist und alle locales korrekt gesetzt wurden.

Teilaufgabe 1.1 - Ausgabe mit ISO-8859-X

Programmfragment	Erwartete Ausgabe auf dem Terminal
`char wort[] = {'A','B','C'};` `cout << wort[2] << endl;`	C
`double zahl = 42.7;` `cout << zahl << "," << (int) zahl << endl;`	42.7 , 42

`string einWort = "Eis";`
`string nochEinWort = "verkaeufer";`
`cout << einWort << nochEinWort << endl;`
`cout << (einWort + nochEinWort).at(3) << endl;`
	Eisverkaeufer
	v

`/* Großbuchsaben im ASCII-Bereich beginnen bei 65 und Enden bei 90 */`
`string name = "HAL";`
`int i;`
`char c;`
`for ( i = 0; i < 3 ; i++ ) {`
`  c = name.at(i);`
`  cout << ++c;`
`} ;`
`cout << endl;`
	IBM

Teilaufgabe 1.2 - Ausgabe mit Unicode

Programmfragment	Erwartete Ausgabe auf dem Terminal
`wstring ws = L"ˈjuːnɪkoʊd";` `wcout << ws[4] << endl;`	n

`wstring xElementN( L"x∈ℕ" );`
`wcout << xElementN.length() << endl;`
`wcout << xElementN.at(2) << endl;`
	3
	ℕ

`/* Unicode-Alpha  'A'  ->  913`
`  ASCII  'A'  ->  65`
`*/`
`wchar_t alphaOrA = L'A';`
`wcout << alphaOrA << endl;`
`wcout << (int) alphaOrA << endl;`
	A
	913 oder 65, ist anhand des dargestellten Zeichens nicht entschiedbar.

Punkte (11 / 11)

Aufgabe 2 – Frequenzliste

Teilaufgabe 2.1 – Erstellen einer Frequenzliste

Schreiben sie ein komplett lauffähiges C++ - Programm, welches die Frequenzliste eines in utf-8-kodierten Textes namens **einText.txt** erzeugt.

```
#include <fstream>
#include <map>
#include <string>
using namespace std;
int main() {
	locale utf8("de_DE.UTF-8");

	wifstream file("einText.txt");
	file.imbue(utf8);
	map<wstring,long> frq;
	wstring token;

	while ( file >> token ) {
		map<wstring,long>::iterator it;

		it = frq.find(token);
		if( it != frq.end() ) {
			it -> second++;
		}
		else {
			frq.insert(map<wstring,long>::value_type(token,1))
		}
	}
}
```

Teilaufgabe 2.2 – Ausgabe einer Frequenzliste

Erweitern sie das Programm dahingehend, dass nach Erzeugung der Frequenzliste diese durchlaufen und auf einem utf-8-fähigen Terminal ausgegeben wird.

```
#include <iostream>
setlocale(LC_ALL,"");
map<wstring,long>::iterator it;
for (it = frq.begin() ; it != frq.end(); it++) {
	wcout << it->first << L"\t" << it -> second << endl;
}
```

Punkte (16 / 16)

Aufgabe 3 – Objektorientierte Programmierung

Schreiben sie eine Klasse **Film** mit den privaten Attributen **wstring titel** , **double spieldauer** und **bool gesehen**. **titel** und **spieldauer** sollen als Attribute dem Konstruktor der Klasse übergeben werden.
Eine öffentliche Methode **anschauen** soll den Wert von **gesehen** von **false** auf **true** setzen.
Denken sie an die „include guards“.

Teilaufgabe 3.1 Header – Film.h

```
#ifndef FILM_H
#define FILM_H

#include <string>

class Film {
public:
    Film( std::wstring, double );
    void anschauen();
private:
    bool gesehen;
    double spieldauer;
    std::wstring titel;
};
#endif /* FILM_H */
```

Teilaufgabe 3.2 Implementierungsdatei – Film.cpp

```
#include "Film.h"

using namespace std;

Film::Film( wstring titel, double spieldauer) {
    this->titel = titel;
    this->spieldauer = spieldauer;
    this->gesehen = false;
}

void Film::anschauen() {
    this->gesehen=true;
}
```

Teilaufgabe 3.3 Main-Methode – main.cpp

Erzeugen sie zwei Instanzen des Objekts Film. Rufen sie die Methode **anschauen** bei der ersten Instanz des Objekts auf.

```
#include "Film.h"

int main(){
	Film dasWandelndeSchloss(L"ハウルの動く城",90);
	Film magnolienAusStahl(L"Steel Magnolias",114);

	dasWandelndeSchloss.anschauen();
}
```

3.4 Kompilieren des Programms

Geben sie den Befehl zum kompilieren des Programms aus Aufgabe 3 auf der Konsole an.

```
$ g++ main.cpp Film.cpp
```

Punkte (20 / 20)

Aufgabe 4 – Collections, STL

Teilaufgabe 4.1

Schreiben sie ein komplett lauffähiges Programm welches die Zahlen von 1 - 100 000 in ein Array und in einen Vektor einliest.

```
#define MAXSIZE 100000
#include <vector>

using namespace std;
int main() {
	long zahlenArray[MAXSIZE];
	vector<long> zahlenVector;

	long i=1; // 1 Counter long
	while (i <= MAXSIZE) { g
		zahlenArray[i - 1] = i;
		zahlenVector.push_back(i);
		i++;
	}
}
```

Teilaufgabe 4.2

Durchlaufen sie den Vektor mit einem Iterator und geben alle Zahlen aus, die durch 42 ohne Rest teilbar sind.

```
#include<iostream>

vector<long>::iterator it;
for (it = zahlenVector.begin(); it != zahlenVector.end(); it++) {
	if (*it % 42 == 0) {
		wcout << * it << endl;
	}
}
```

Teilaufgabe 4.3

Durchlaufen sie das Array rückwärts und bilden sie die Summe aller Zahlen im Array. Geben sie bei jedem tausendsten Element ein Zwischenergebnis aus.

```
long sum = 0;
long pos = 99999;

while (pos >= 0) {
	sum+= zahlenArray[pos];
	pos--;
	if (pos % 1000 == 0) { // 1 %1000
		wcout << L"Zwischenergebnis:" << sum << endl;
	}
}
wcout << "Summe:" << sum << endl;
```

Punkte (21 / 21)

Aufgabe 5 – Boost

Füllen sie die Leerstellen mit den korrekten Codefragmenten um ein lauffähiges Programm zu erzeugen.

Teilaufgabe 5.1

Gegeben ist folgendes Programm, welches einen Text an Leer- und Satzzeichen aufsplittet und in einem späteren Schritt die einzelnen Tokens auf dem Terminal ausgibt.

```
#include <string>
#include <vector>
#include <iostream>
[#include <boost/foreach.hpp>] //1
[#include <boost/algorithm/string.hpp>] //1
using namespace std;
[using namespace boost]; //1

int main() {
  setlocale(LC_ALL,"");
  [wstring] yoda = L"Yes. A Jedi's strength flows from the Force. But beware the dark
side. Anger, fear, aggression. The dark side of the Force are they. Easily they flow,
quick to join you in a fight.!"; //1
  vector<[wstring]> words;
  split( [words], yoda, [is_any_of(L" .,;:!?")], token_compress_on ); // 1 1

  BOOST_FOREACH( [wstring word], words) { //1
                wcout <<[ word ]<< endl; //1
  }
} // 9 Punkte
```

Teilaufgabe 5.2

Gegeben ist folgendes Programm, welches anhand von regulären Ausdrücken bestimmt, ob ein Namen einem bestimmten Kriterium entspricht und dazu einen passenden Satz ausgibt.

```
#include <iostream>
#include <string>
#include <boost/foreach.hpp>
#include [<boost/regex.hpp>] // 1
[using namespace std;] //1

int main() {
  string [namen[]] = {"Luke","Leia","Han","Chewbacca"}; // 1
  boost::regex vierBuchstaben(["\\w{4}"]); //1
  boost::regex beginntMitC(["^C.*"]); // 1

  BOOST_FOREACH( [string name], namen) { //1

      if ( boost::regex_match( name, [vierBuchstaben] ) ) { // 1
           cout << name <<"! The force is strong in this one. " << endl;
      }
      if ( [boost::regex_match]( name ,beginntMitC ) ) { // 1
                      [cout] << name <<  " is a Wookiee!" << endl; //1
                }
      }
}
```

Punkte (18 / 18)

Aufgabe 6 – Syntaxfehler finden

Unterstreichen und korrigieren sie die vorhandenen Syntaxfehler.

Teilaufgabe 6.1 (8 Syntaxfehler)

```
#include<iostream>
include<list> // 1#
#include<algorithm>
using std namespace; // 1 std namespace vertauscht

void putOut(string);
list[string] buildList(); // 1 <>

int main() {
    list<string> myList = buildList()  //1 ;

    for_each( myList.begin() ; myList.end(), putOut() );  //1 ;1 Funktion ohne
KLammern
}

void putOut(string ding) {
    cout >> ding >> endl; //1 falsch rum
}

list<string> buildList() {
    list<string> myList = list<wstring>(); //1 wstring
    myList.push_back("1");
    myList.push_back("2");
    myList.push_back("drei");
    myList.push_front("null");

    return myList;
}
```

Teilaufgabe 6.2 (6 Syntaxfehler)

```
#include <string>
#include <iostream>
#include "boost/foreach.hpp> // 1
using namespace std;

int main[] {       //1 []
    setlocale(LC_ALL, "'); //1 '
    wstring unicode( "[ˈjuːnɪkoʊd]" ); //1 L

    BOOST_FOREACH( wchar_t ch  unicode ){ //1 ,
        wcout << ch  << endl;
      //1 schließende Klammer
}
```

Punkte (14 / 14)

Zusatzblatt

Nutzen Sie dieses Blatt, wenn ihnen der Platz ausgeht.