Exercise Week 05

GianAndrea Müller
mailto:muellegi@student.ethz

March 28, 2018

Exercise Week 05

GianAndrea Müller
mailto:muellegi@student.ethz

March 28, 2018

Time Schedule

- 20' Self assessment 2
- 15' Korrektur
- 10' Standard Library
- 15' Referenzen mit Übung

Time Schedule

- 20' Self assessment 2
- 15' Korrektur
- 10' Standard Library
- 15' Referenzen mit Übung

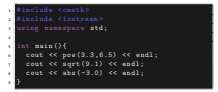
Learning Objectives

- Kenntnis von Bibliotheken
- Verständnis von Referenzen

Learning Objectives

- Kenntnis von Bibliotheken
- Verständnis von Referenzen

Standardbibliotheken



Standardbibliotheken

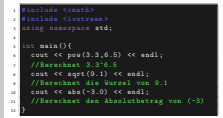
- Bibliothek: cmath für eine Auswahl mathematischer Funktionen.

```cpp
#include <cmath>
#include <iostream>
using namespace std;

int main(){
  cout << pow(3.3,6.5) << endl;
  cout << sqrt(9.1) << endl;
  cout << abs(-3.0) << endl;
}
```

- Lösung

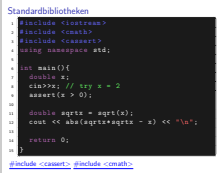
Standardbibliotheken



Standardbibliotheken

```cpp
#include <cmath>
#include <iostream>
using namespace std;

int main(){
  cout << pow(3.3,6.5) << endl;
  //Berechnet 3.3^6.5
  cout << sqrt(9.1) << endl;
  //Berechnet die Wurzel von 9.1
  cout << abs(-3.0) << endl;
  //Berechnet den Absolutbetrag von (-3)
}
```



- cassert um Bedingungen zu überprüfen.
- Wenn nicht erfüllt wird mit Error abgebrochen.
- Fehler von sqrt() überprüfen.

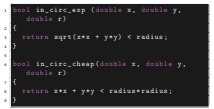
Standardbibliotheken

```
1  #include <iostream>
2  #include <cmath>
3  #include <cassert>
4  using namespace std;
5
6  int main(){
7    double x;
8    cin>>x; // try x = 2
9    assert(x > 0);
10
11   double sqrtx = sqrt(x);
12   cout << abs(sqrtx*sqrtx - x) << "\n";
13
14   return 0;
15 }
```

#include <cassert> #include <cmath>

- Weiteres Beispiel mit funktionen.
- Sqrt kann vermieden werden -¿ teure operation.
- Wird später wichtig effizient zu programmieren.

Standardbibliotheken



Standardbibliotheken

```
1  bool in_circ_exp (double x, double y,
       double r)
2  {
3    return sqrt(x*x + y*y) < radius;
4  }
5
6  bool in_circ_cheap(double x, double y,
       double r)
7  {
8    return x*x + y*y < radius*radius;
9  }
```

- Algorithm für eine Auswahl an kleinen Alrogrithmen wie max und min.

Standardbibliotheken

Standardbibliotheken

```
1 #include <iostream>
2 #include <algorithm>
3
4 int main(){
5   cout << min(3.5,4.1) << "\n";
6   cout << max(3.4,9.1) << "\n";
7   return 0;
8 }
```

#include <algorithm>

Referenzen

Referenzen

- Trotzdem wird n nicht vergrössert.
- Das liegt daran, dass 3 als rvalue an die Funktion übergeben wird.
- Dies lässt sich mit einem Zeichen beheben:

```
1 void increment (int m) {
2 m++;
3 }
4 int main () {
5 int n = 3;
6 increment (n);
7 return 0;
8 }
```

Referenzen

Referenzen

- Zweiter Name für die gleiche Variable.
- Eine Referenz wird bei ihrer Deklaration initialisiert!
- Eine Referenz "zeigt" immer auf die gleiche Variable.

```
1 int i = 1;
2 int& j = i;
3 i++; // i = 2
4 j++; // i = 3
```

Referenzen

Referenzen

- Wirkung: m wird neu als Referenz übergeben.
- Beim Funktionsaufruf wird für n der neue Name m als Referenz erzeugt.
- Parameter müssen nicht verändert werden, sie werden automatisch referenziert.
- Es ist aber nicht mehr möglich direkt eine Zahl als Parameter zu übergeben.

```
1 void increment (int& m) { //only line
    changed
2 m++;
3 }
4 int main () {
5 int n = 3;
6 increment (n);
7 return 0;
8 }
```

Funktionstypen
Call by value
bool even (unsigned int a){
while (a>=1) a-=2;
return a != 1;
}
Call by reference
void half (int & b){
b /= 2;
}

Funktionstypen

- **Call by value**

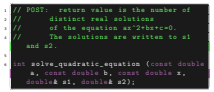
```
1 bool even (unsigned int a){
2   while (a>=1) a-=2;
3   return a != 1;
4 }
```

- **Call by reference**

```
1 void half (int & b){
2   b /= 2;
3 }
```

- Kombination von Beidem
- Ermöglich das "zurückgeben" von mehr datenpunkten!

Funktionstypen



Funktionstypen

```
1 // POST:  return value is the number of
2 //       distinct real solutions
3 //       of the equation ax^2+bx+c=0.
4 //       The solutions are written to s1
    and s2.
5
6 int solve_quadratic_equation (const double
     a, const double b, const double x,
    double& s1, double& s2);
```

Letzte Seite
assi-link

Letzte Seite

assi-link