

Funktionen

Deklaration	Erweiterung des Gültigkeitsbereiches der Funktion
<p>Eine Funktion kann im Programm nur in Zeilen verwendet werden, welche nach der ersten Deklaration der Funktion kommen.</p> <p>Die Definition der Funktion ist immer gleichzeitig auch eine Deklaration.</p> <p>Bei Deklarationen kann man <code>const</code> bei den Argumenten auch weglassen (sollte dies wenn möglich aber nicht machen).</p>	
<pre>void B (const int i); // separate declaration void A (const int i) { if (i <= 0) return; // stop calls std::cout << "A"; B(i/2); // use of B although its definition happens below } void B (const int i) { if (i <= 0) return; // stop calls std::cout << "B"; A(i/2); }</pre>	

Standardbibliothek

<code>std::pow</code>	Potenzieren
Erfordert: <code>#include <cmath></code>	
<pre>double a = std::pow(2.5, 2); // Computes 2.5 ^ 2 == 6.25</pre>	

Programmier-Befehle - Woche 6

<code>std::sqrt</code>	Quadratwurzel
Erfordert: <code>#include <cmath></code>	
IEEE 754 garantiert, dass der (mathematisch) exakte Wert auf die nächste repräsentierbare Zahl gerundet wird.	
<pre>double a = std::sqrt(14.0625); // Result: 3.75</pre>	

<code>std::abs</code>	Absolutbetrag
Erfordert: <code>#include <cmath></code>	
<pre>double a = std::abs(-3.5); // Result: 3.5</pre>	

<code>std::min</code>	Minimum zweier Argumente
Erfordert: <code>#include <algorithm></code>	
Sonst gibt es noch: <code>std::max</code> Maximum zweier Argumente	
Wichtig ist, dass beide Argumente vom selben Typ sind. Sonst geht es nicht .	
<pre>double z; std::cin >> z; std::cout << std::min(z, 1.0); // min of z and 1 std::cout << std::min(z, 1); // Error: z is double, 1 is int</pre>	

Generell

namespace	Katalogisierung von Befehlen
<p>Mit <code>namespaces</code> kann man Funktionen, Typen, etc. katalogisieren (z.B. bezüglich Projekten). Beispielsweise werden viele der "offiziellen" Funktionen dieser Vorlesung im <code>namespace ifmp</code> zusammengefasst. So können Sie diese Funktionen einfach von Ihren eigenen Funktionen unterscheiden.</p>	
<pre>namespace ifmp { // namespace called ifmp // POST: "Hi" was written to the terminal void output_func () { // this function is in namespace ifmp std::cout << "Hi"; } } int main () { ifmp::output_func(); // use output_func from namespace ifmp ifmp::integer a = 2147483647; // also in namespace ifmp return 0; }</pre>	