

8. Abend

Ziel :

- Wir verbessern unser Verständnis für Funktionen und erkennen Situationen, in denen wir Funktionen einsetzen sollten

Funktionen mit default-Argumenten

Die meisten Funktionen haben ein oder mehrere Parameter. Hier ein sehr nützliches Beispiel einer Funktion, die mehrmals Hallo ausgeben kann.

```
#include <iostream>

using namespace std;

void SagHallo(int wievielmals)
{
    for(int i = 0; i < wievielmals; ++i)
    {
        cout << "Hallo" << endl;
    }
}

int main()
{
    SagHallo(5);

    return 0;
}
```

In diesem Beispiel nutzt die main-Funktion die "SagHallo"-Funktion um fünf mal Hallo auszugeben. Meistens reicht es nur einmal Hallo zu sagen. Also würden wir die Funktion mit dem Wert 1 als Parameter aufrufen. Noch besser wäre es wir könnten die Funktion ohne Parameter aufrufen um einmal Hallo auszugeben und sonst einen Parameter um mehrmals Hallo zu sagen.

In C++ ist es möglich default-Parameter zu definieren. Das heisst dass man einem Parameter einen gewünschten Wert geben kann und diesen beim Aufruf nicht mehr angeben muss, aber kann falls man diesen Vorgabewert nicht brauchen will !

Das ganze wird deutlich wenn wir uns ein Beispiel ansehen:

```
#include <iostream>

using namespace std;

// im default-Fall nur einmal "Hallo"
// ausgeben
void SagHallo(int wievielmal = 1)
{
    for(int i = 0; i < wievielmal; ++i)
    {
        cout << "Hallo" << endl;
    }
}

int main()
{
    // Wir rufen SagHallo ohne
    // Parameter auf, also wird
    // der default-Parameter
    // verwendet
    SagHallo();

    cout << "Wieviel Hallo ? ";
    int anzahlHallo;
    cin >> anzahlHallo;

    SagHallo(anzahlHallo);

    return 0;
}
```

Es können auch mehrere Parameter default-Werte haben. Dabei ist zu beachten, dass die Parameter mit default-Werten die letzten in der Parameterliste sind. Anders ausgedrückt können nach einem Parameter mit default-Werten nur andere Parameter mit default-Werten folgen.

```
#include <iostream>
#include <string>

using namespace std;

void SagWas(string was = "Hallo", int wievielmals = 1)
{
    for(int i = 0; i < wievielmals; ++i)
    {
        cout << was << endl;
    }
}

int main()
{
    SagWas();

    SagWas("Test", 2);

    SagWas("Saluti");

    return 0;
}
```

Überladen einer Funktion

In C++ ist es möglich mehrere Funktionen mit dem gleichen Namen zu haben, solange sie sich in den Parameterlisten unterscheiden. Man nennt dies Überschreiben einer Funktion. Die Anzahl und der Datentyp der Parameter wird vom Compiler also zusätzlich zum Namen verwendet um Funktionen zu unterscheiden. Die Anzahl und der Datentyp der Parameter einer Funktion bezeichnet man als Signatur der Funktion.

Beim folgenden Beispiel überschreiben wir die Funktion max für zwei double Werte sowie für zwei int Werte.

```
int max(int a, int b)
{
    if(a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}

double max(double a, double b)
{
    if(a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}

int main()
{
    double maxd = max(12.5, 45.7);

    int maxn = max(2, 5);

    return 0;
}
```

Übung Notendurchschnitt

Die klassische Problemstellung eines jeden Studenten :-) Ein Programm zur Berechnung des Notendurchschnitts...

Das Programm soll folgendermassen funktionieren :

Der Benutzer wird begrüsst und kann danach eine Note eingeben. Nach dieser Noteingabe wird die Anzahl Noten angezeigt und der Durchschnitt. Danach wird der Benutzer gefragt, ob er eine weitere Note eingeben will. Falls ja, kann er eine weitere Note eingeben, worauf das Programm wieder die Anzahl Noten und den Durchschnitt anzeigt.

Versuche die Aufgabe aufzuteilen, indem du beispielsweise eine Funktion schreibst, die den Durchschnitt berechnet. Übergebe dieser Funktion jeweils die Summe der Noten und die Anzahl Noten. Die Funktion gibt den Durchschnitt als Rückgabewert zurück.

Eine weitere Funktion könnte den Benutzer jeweils Fragen ob er eine weitere Note eingeben will oder nicht. Sie könnte als Rückgabewert einen bool haben.

Auch das Einlesen der Note könnte eine Funktion sein, die zugleich die Eingabe des Benutzers überprüft und die gültige Note als Rückgabewert hat.

Selbstverständlich kann die Begrüssung auch in einer Funktion versteckt sein.

Du wirst im Programm die Summe der Noten als Variable brauchen. Was für ein Datentyp ist für eine Schulnote passend ? Auch die Anzahl der Noten musst du in einer Variable merken, die du bei jeder neuen Note erhöhen musst. Was ist der Datentyp hierfür ?

Wenn du willst kannst du die Ausgabe schön formatiert machen. Du kannst dafür die Console.h und Console.cpp verwenden, die ich am letzten Abend vorgestellt habe.