

Übung 2

Ziel der Übung:

- Einrichten und Testen der Entwicklungsumgebung (siehe Dokument "IDE_Einrichten_Testen.pdf")
- Aufbau und allgemeine Syntax eines C++-Programms
- Eingabe und Ausgabe in C++
- Verzweigungen in C++ mit Hilfe der if-Anweisung
- Variablen und Datentypen in C++.
- Schrittweise Ausführung (Debuggen)

Aufgabe 1: Das erste "richtige" Programm

Erstellen Sie ein neues Projekt mit dem Namen "Aufgabe-1". Übernehmen Sie den folgenden Quelltext, welcher das Alter einer Person auf Basis des Geburtsjahres und des aktuellen Jahres berechnet. Ersetzen Sie den Namen und das Geburtsjahr durch ihre eigenen Angaben.

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

int main() {
    // Variablen
    int geburtsjahr = 2000;
    int aktuellesJahr = 2016;
    int alter;

    // Berechnung
    alter = aktuellesJahr - geburtsjahr;

    // Ausgabe
    cout << "Vorname: Max" << endl;
    cout << "Nachname: Mustermann" << endl;
    cout << "Alter: " << alter << endl;

    return 0;
}
```

Teil a) Das Programm enthält ein paar **Fehler**, d.h. beim **Erstellen** werden **Fehlermeldungen** ausgegeben. **Korrigieren** Sie diese Fehler, damit Sie das Programm testen können.

Teil b) In C++ kann man mit "**cin >>**" aus dem **Eingabestrom** von der Konsole einen Wert der Benutzereingabe in eine Variable lesen. Ergänzen Sie das obige Programm so, dass Sie das **Geburtsjahr einlesen** und in die Variable geburtsjahr schreiben.

Aufgabe 2: Flächeninhalt und Umfang eines Kreises berechnen

Der Flächeninhalt A und der Umfang U eines Kreises lassen sich wie folgt berechnen:

$$A = \pi * r^2 = \frac{\pi * d^2}{4}$$
$$U = 2 * \pi * r = \pi * d$$

In C++ kann eine Berechnung mit Hilfe der Grundoperationen + (Addition), - (Subtraktion), * (Multiplikation) und / (Division) durchgeführt werden.

Teil a) Erstellen Sie ein neues Projekt mit dem Namen "Aufgabe-2" und übernehmen Sie den folgenden Quelltext. Vervollständigen Sie das angegebene Programm durch die **Berechnung und Ausgabe des Flächeninhalts und des Umfangs** eines Kreises mit dem gegebenen Durchmesser.

Hinweis: Um den Wert für PI zu erhalten, können Sie die PI-Konstante `M_PI` aus der `cmath`-Bibliothek verwenden.

```
#define _USE_MATH_DEFINES
#include <cmath>
#include <iostream>
using namespace std;

int main() {
    // Variablen
    float durchmesser = 1;
    float flaecheninhalt;
    float umfang;

    // Hier bitte die Berechnungen einfüegen
    ...

    // Hier bitte die Ausgabe vervollständigen
    cout << "Der Flaecheninhalt betraegt: " << ... << endl;
    cout << "Der Umfang ist: " << ... << endl;

    return 0;
}
```

Teil b) Ergänzen Sie das obige Programm so, dass Sie den Wert für den **Durchmesser einlesen** und in die Variable `durchmesser` schreiben.

Teil c) Um den Flächeninhalt bzw. den Umfang eines Kreises berechnen zu können, sollte geprüft werden ob der Durchmesser größer als 0 ist. Um eine Bedingung zu prüfen, können Sie in C++ die **if-Anweisung** verwenden. Ergänzen Sie das obige Programm so, dass Sie den eingelesenen Wert für den **Durchmesser prüfen**. Führen Sie die Berechnungen nur durch, wenn die Bedingung erfüllt ist, ansonsten geben Sie eine Fehlermeldung aus.

Aufgabe 3: Variablen, Datentypen und deren Werte

Gegeben sei folgendes C++-Programm:

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

int main() {
    int i, j, k;
    float x, y, z;

    i= 1;
    j= i + i * 9;
    k= j / 4;

    // Variablenbelegung hier?

    x= 2.5;
    y= x + i;
    z= j / 4;

    // Variablenbelegung hier?

    return(0);
}
```

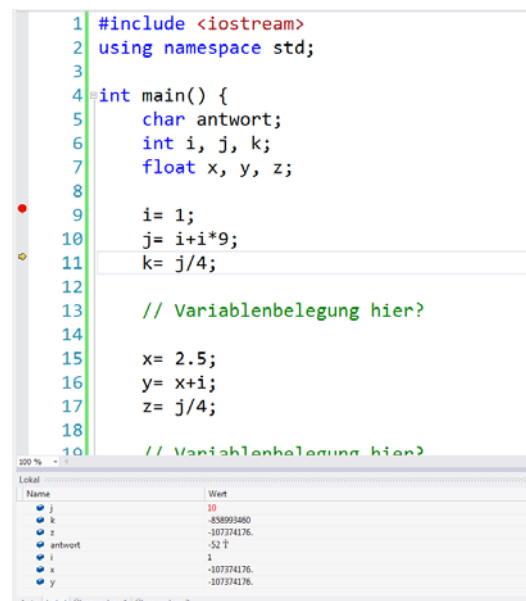
Teil a) Überlegen Sie sich bitte, welche Werte die Variablen *i*, *j*, *k*, *x*, *y* und *z* an den gekennzeichneten Stellen haben?

Teil b) Erweitern Sie das Programm so, dass Sie die **Belegung der Variablen** am Ende des Programms **ausgegeben**. Können Sie sich die Variablenbelegungen erklären?

Teil c) Erweitern Sie das Programm so, dass Sie die **Werte für i und x einlesen**.

Teil d) Die Entwicklungsumgebung stellt die Möglichkeit bereit, ein Programm **schrittweise auszuführen** und dabei in jedem Schritt anzusehen, welche Werte aktuell in den Variablen gespeichert sind. Dieser Prozess heißt **"Debuggen"**.

- Für das Debuggen können **sogenannte Breakpoints** gesetzt werden. Dazu gehen Sie in eine Programmzeile am Anfang der Datei und setzen mit der Taste **<F9>** einen Breakpoint. Diese wird durch einen roten Punkt gekennzeichnet.
- Starten Sie jetzt das Programm mit der Taste **<F5>** (Debugging starten).
- Das Programm wird bis zum Breakpoint ausgeführt und dann angehalten.
- Sie können ab diesem Breakpoint das Programm **schrittweise** ablaufen lassen, mit der Taste **<F10>** kommen Sie jeweils zum nächsten Befehl.
- Sehen Sie sich bitte in dem Fenster **"Lokal"** (unten links) an, welche Variablen existieren und welchen Wert diese haben.



Hausaufgaben Serie 2

Hausaufgaben als txt oder cpp Datei mit Namen, Studiengang und Matrikelnummer unter „**Aufgaben**“ auf StudIP hochladen.
Abgabe bis **02.11.2021** (ÜG-1), **03.11.2021** (ÜG-4), **04.11.2021** (ÜG-2) und **05.11.2021** (ÜG-3).

Aufgabe 1: Syntaxfehler korrigieren

Erzeugen Sie ein neues Projekt, wobei folgender Quelltext exakt zu übernehmen ist!

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

int main() {
    int zahl 3;
    cout << "Die Zahl hat den ";
    cout << "Wert " zahl;
    return 0
}
```

Beim Übersetzen ("Build" Prozess) erhalten Sie **Fehlermeldungen**. Welche sind das?

Korrigieren Sie das Programm und geben es in der korrigierten Version ab.

(3 Punkte)

Aufgabe 2: Artikelbestellung prüfen

Erstellen Sie ein C++-Programm mit dem unten angegebenen Quelltext. Das Programm soll ein gegebenes Budget bezüglich der Bestellung eines Artikels prüfen. Das Budget, die Anzahl der Artikel und deren Stückpreis sollten von der Konsole eingelesen werden. Überlegen Sie auch, welche Variablen Sie zusätzlich zu den Variablen budget, artikel_anzahl und artikel_preis benötigen. Ersetzen Sie die Punkte im Programm durch Ihre eigenen Angaben. Geben Sie Ihre Quelldatei mit dem **vollständigen Programm** ab.

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

int main() {
    int budget;
    int artikel_anzahl;
    int artikel_preis;
    // Budget, Anzahl der Artikel und Stückpreis einlesen
    ...
    // Berechnung des Gesamtpreises
    ...
    // Ausgabe
    cout << "Sie haben "<< ... << " Artikel zu je "<< ... << " Euro bestellt." << endl;
    cout << "Der Gesamtpreis der Bestellung beträgt "<< ... << " Euro. " << endl;
    // Berechnung des Restbudgets
    ...
    // Budget prüfen (Restbudget grösser gleich 0) und Ausgabe
    ...
    cout << "Sie haben ein Restbudget von " << ... << " Euro. " << endl;
    // oder
    cout << "Sie ueberschreiten Ihr Budget um "<< ... << " Euro." << endl;

    return 0;
}
```

Beispiel: Die Eingaben (in blau dargestellt) und Ausgaben sollen wie folgt aussehen.

Budget eingeben: 100

Anzahl der Artikel eingeben: 10

Stueckpreis eingeben: 6

Sie haben 10 Artikel zu je 6 Euro bestellt.

Der Gesamtpreis der Bestellung betraegt 60 Euro.

Sie haben ein Restbudget von 40 Euro.

(8 Punkte)

Aufgabe 3: Währung umrechnen

Schreiben Sie ein C++-Programm, das einen Betrag von schwedischen Kronen (SEK) in Euro umrechnet und ausgibt. Zu diesem Zweck sind der Betrag und der Umrechnungskurs von SEK in Euro einzulesen.

Beispiel: Die Eingaben (in blau dargestellt) und Ausgaben sollen wie folgt aussehen.

Betrag in SEK eingeben: 12

Umrechnungskurs SEK in Euro eingeben: 9.35

Betrag in Euro: 112.2

(5 Punkte)