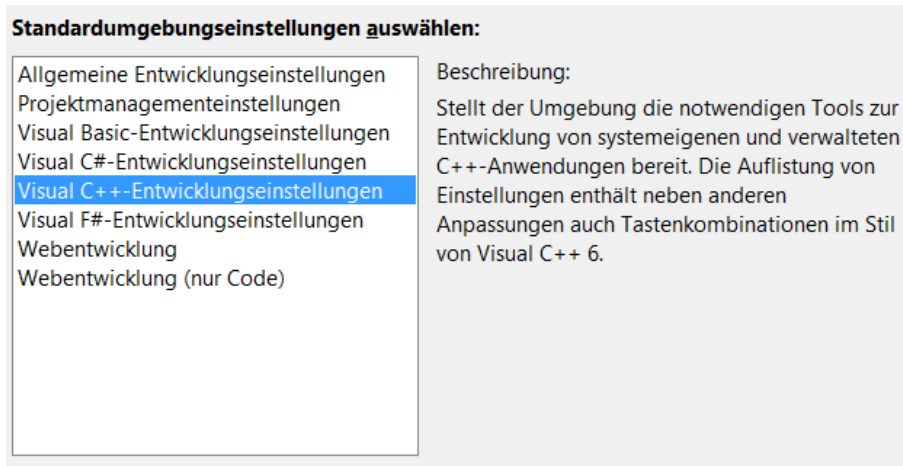


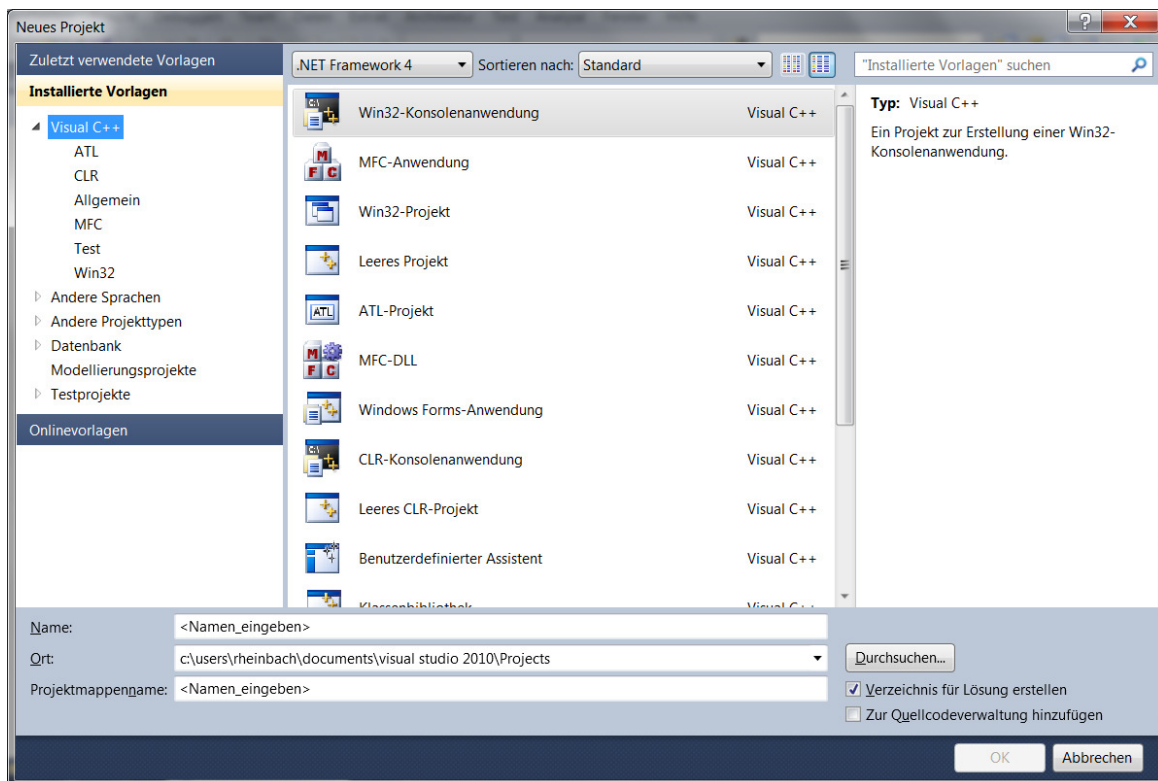
Wenn Sie Visual Studio das erste Mal starten, dann wählen Sie die C++-Spracheinstellungen



Aufgabe 1

Öffnen Sie **Visual Studio 2010** und wählen unter **Datei**→**Neu**→**Projekt** im Dialogfenster erst **“Win32”** und dann rechts **“Win32-Konsolenanwendung”** aus. Nennen Sie Ihr Projekt **“hallowelt”**, klicken sie auf **“OK”**, dann auf **“Weiter”**.

Achtung: Wenn Sie keinen Namen eingegeben haben, bleibt das **“OK”**-Feld grau.

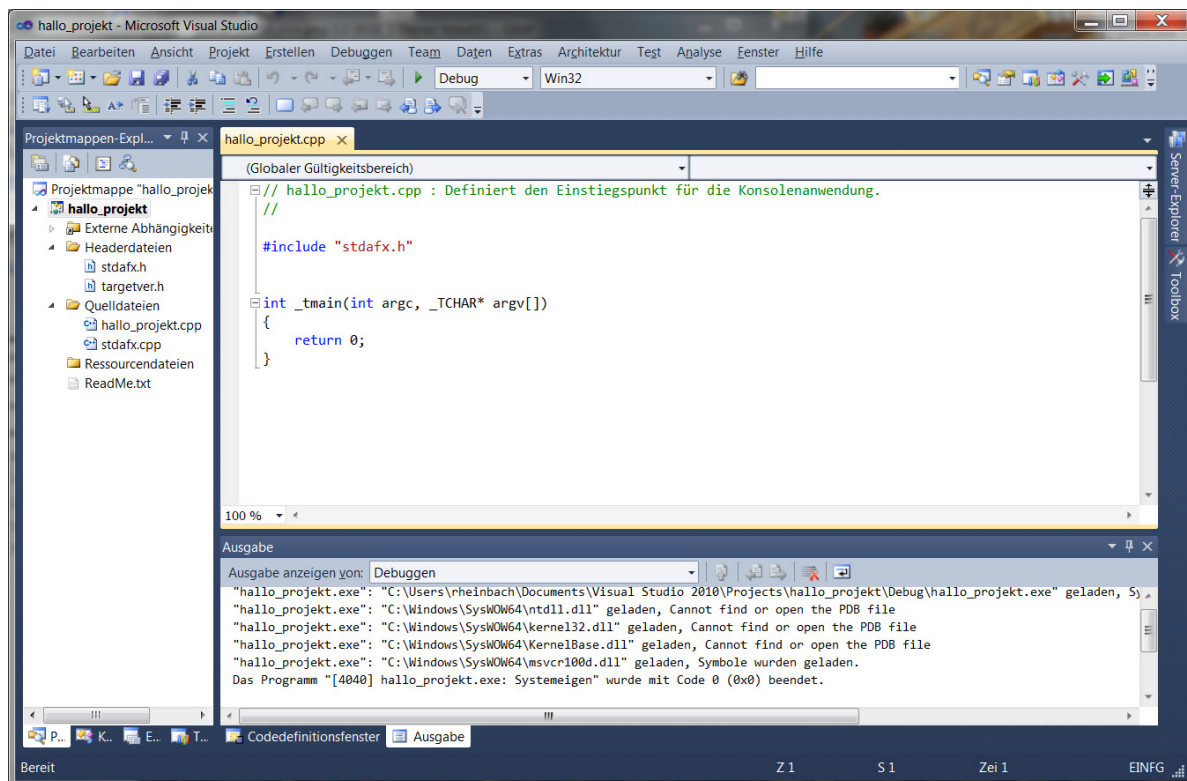


Zusätzliche Optionen:

- Leeres Projekt
- Symbole exportieren
- Vorkompilierter Header

Entfernen Sie im angezeigten Dialogfenster unbedingt das grüne Häkchen bei **Vorkompilierte Header (Precompiled header)** und klicken Sie im Anschluss auf “Fertigstellen”.

Die Entwicklungsumgebung erstellt für Sie automatisch ein leeres Programm.



Lassen Sie sich nicht davon irritieren, dass die Definition des Hauptprogramms `main` nicht konform zum C++-Standard ist. Microsoft passt Standards gerne an eigene Bedürfnisse an. Das automatisch erstellte leere Programm lautet:

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    return 0;
}
```

Zum Glück akzeptiert der Compiler auch standardkonforme Programme. Ändern Sie das Programm ab zu:

```

#include <iostream>
#include <string>


using namespace std;

int main()
{
    string a;
    cout<<"Hallo Welt! Geben Sie etwas ein."<<endl;
    cin>>a;

    return 0;
}

```

Übersetzen Sie das Projekt indem Sie **Erstellen**→**Projektmappe erstellen** auswählen, oder F7 betätigen. Achten Sie im unteren Fenster auf Fehlermeldungen. Beachten Sie, dass Sie stets Hilfe zu einem Schlüsselwort bekommen können, indem Sie den Cursor in das Schlüsselwort Ihres Programms bewegen und dann F1 drücken. Führen Sie das übersetzte Programm aus, indem Sie **Debuggen**→**Debuggen starten** auswählen, oder F5 drücken.

Zum Starten des Programms können auch auf  drücken.

Aufgabe 2

Schreiben Sie ein Programm, dass die folgenden Zeilen auf dem Bildschirm erzeugt:

```

Dies
ist das zweite
Programm.

```

Aufgabe 3

Suchen Sie in “Desktop/Eigene Dateien/Visual Studio 2008/Projects” nach den Dateien Ihres Projektes. In “Debug” finden Sie Ihr Programm, das Sie durch Doppelclick starten können.

Öffnen Sie eine Eingabeaufforderung (“Startmenü/Zubehör”) wechseln Sie mit `cd` in das Verzeichnis “Debug” und starten Sie Ihr Programm durch Eingabe seines Dateinamens.

Aufgabe 4

Schreiben Sie ein Programm, dass die folgenden Zeilen auf dem Bildschirm erzeugt,

Geben Sie ein paar Zeichen ein.

dann die Zeichen in einen `string` einliest und sie wieder ausgibt.

Aufgabe 5

Schreiben Sie ein C++-Programm, das eine ganze Zahl vom Benutzer einliest, sie um 10 erhöht, und sie dann zweimal wieder ausgibt. Schreiben Sie dann ein weiteres Programm, das zwei Zahlen einliest und ihr Produkt wieder ausgibt.

Aufgabe 6

Was gibt das folgende Programm aus? Probieren Sie es aus.

```
#include <iostream>

using namespace std;

int main()
{
    int x=4, y=4;

    cout<<x<<endl;
    cout<<x++<<endl;
    cout<<x<<endl;

    cout<<y<<endl;
    cout<<++y<<endl;
    cout<<y<<endl;

    return 0;
}
```

Aufgabe 7

Mit `#include <cstdlib>` können Sie die Deklaration der Funktion `int rand()` einbinden. Informieren Sie sich mit der Hilfe über die Funktion und schreiben sie das berühmte Zahlenrateprogramm. Eine Sitzung mit diesem Programm sieht typischerweise so aus.

```
Ich habe mir eine Zahl zwischen 0 und 20 ausgedacht. Rate welche!
672
Das ist eine dumme Antwort, die Zahl ist zwischen 0 und 20.
18
Das ist leider falsch. Rate nochmal.
15
Herzlichen Gl\"uckwunsch, das ist richtig.
```

Aufgabe 8

Bestimmen Sie durch ein C++-Programm, welches die größte und kleinste Zahl ist, die ein `int` bzw. ein `unsigned int` fassen kann. Passt in ein `long int` mehr?

Aufgabe 9

Schreiben Sie ein Programm, das eine Zahl einliest und dann ihre Ziffern in umgekehrter Reihenfolge wieder ausgibt. Tipp: Verwenden Sie ganzzahlige Division, um die Ziffern zu bestimmen.

Aufgabe 10

Schreiben Sie ein Programm, das den GGT zweier Zahlen bestimmt, die vorher eingelesen werden. Zur Erinnerung, zum Euklidischen Algorithmus findet man etwa z.B.

```
EUCLID(a,b)
1  solange b nicht gleich 0
2      wenn a > b
3          dann a = a - b
4          sonst b = b - a
5  return a
```

Verfolgen Sie die Berechnungen mit dem Debugger: Durch Klicken links neben eine Programmzeile setzen Sie einen Haltepunkt.

Das Programm wird dann bei Erreichen dieser Zeile angehalten. Sie können nun durch bewegen des Mauszeigers über eine Variable ihren Werte auslesen. Mit "Debuggen/Einzelschritt" oder "Debuggen/Prozedurschritt" bzw. F11 oder F10 können Sie das Programm Zeile für Zeile ausführen.

Aufgabe 11

Der Euklidische Algorithmus kann auch in einer rekursiven Version angegeben werden:

```
EUCLID_RECURSIVE(a,b)
1  wenn b = 0
2      dann return a
3      sonst wenn a > b
4          dann return EUCLID_RECURSIVE(a - b, b)
5          sonst return EUCLID_RECURSIVE(a, b - a)
```

Die rekursive Variante funktioniert durch Selbstaufwurf der Funktion. Verfolgen Sie auch hier mit dem Debugger die Funktionsweise des Programms und beachten Sie die Werte der lokalen Variablen.

Aufgabe 12

Lehrer benötigen oft für Klausuraufgaben für rechtwinklige Dreiecke “schöne” Zahlentripel. Tripel natürlicher Zahlen, die $a^2 + b^2 = c^2$ erfüllen, heißen pythagoräisch. Schreiben Sie ein Programm, das durch systematisches Ausprobieren pythagoräische Zahlentripel findet und sie dann ausgibt.

Aufgabe 13

Das Sieb des Erastosthenes findet Primzahlen, indem es Vielfache von 2, 3 usw. streicht. Die Zahlen, die zurückbleiben sind prim. Schreiben ein Programm, das das Sieb des Eratosthenes verwendet, um Primzahlen zu berechnen und auszugeben.

Sie sind fertig? Fragen Sie nach Aufgabenblatt 2!