

↓
Kapitel Visual C++ als PDF exportieren

VISUAL C++

Visual C++ ist ein Compiler des Unternehmens Microsoft zur Entwicklung von Software in der Programmiersprache C++ unter dem Betriebssystem Windows und anderen Betriebssystemen von Microsoft. Aktuelle Versionen von Visual C++ verfügen über den erweiterten C++-Befehlssatz C++/CLI, der unter anderem die Nutzung der .NET-Programmierung vereinfachen soll.

- [Installation des RAD-Studios 10.2](#)
- [Erstellen eines ausführbaren EXE-Projekts](#)
- [RAD-Studio Tutorial \(deutsch\)](#)
- [DocWiki RAD-Studio 10.2](#)

Übungsbeispiele

- [01 Einführungsbeispiel](#)
- [02 Editor](#)
- [03 Forms & Listboxen](#)
- [04 Bibliothek](#)
- [05 Quadratische Gleichungen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus



Last update: **2018/03/20 14:11**

Einführung - Beispiel

Angabe

Dieses Programm gibt einen ersten kurzen Einblick in die visuelle Programmierung mit dem EMBARCADERO RAD STUDIO.

Lösung - Unit.cpp

```
//-----  
-  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
-  
  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
-  
  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-  
  
void __fastcall TForm1::EditNachnameClick(TObject *Sender)  
{  
    ShowMessage("Sie haben das Textfeld angeklickt");  
}  
//-----  
-  
  
void __fastcall TForm1::EditNachnameKeyPress(TObject *Sender,  
System::WideChar &Key)  
{  
    EditCopy->Text=EditNachname->Text;  
}  
//-----  
-
```

Download

[01Einfuehrung.zip](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus:01ueb



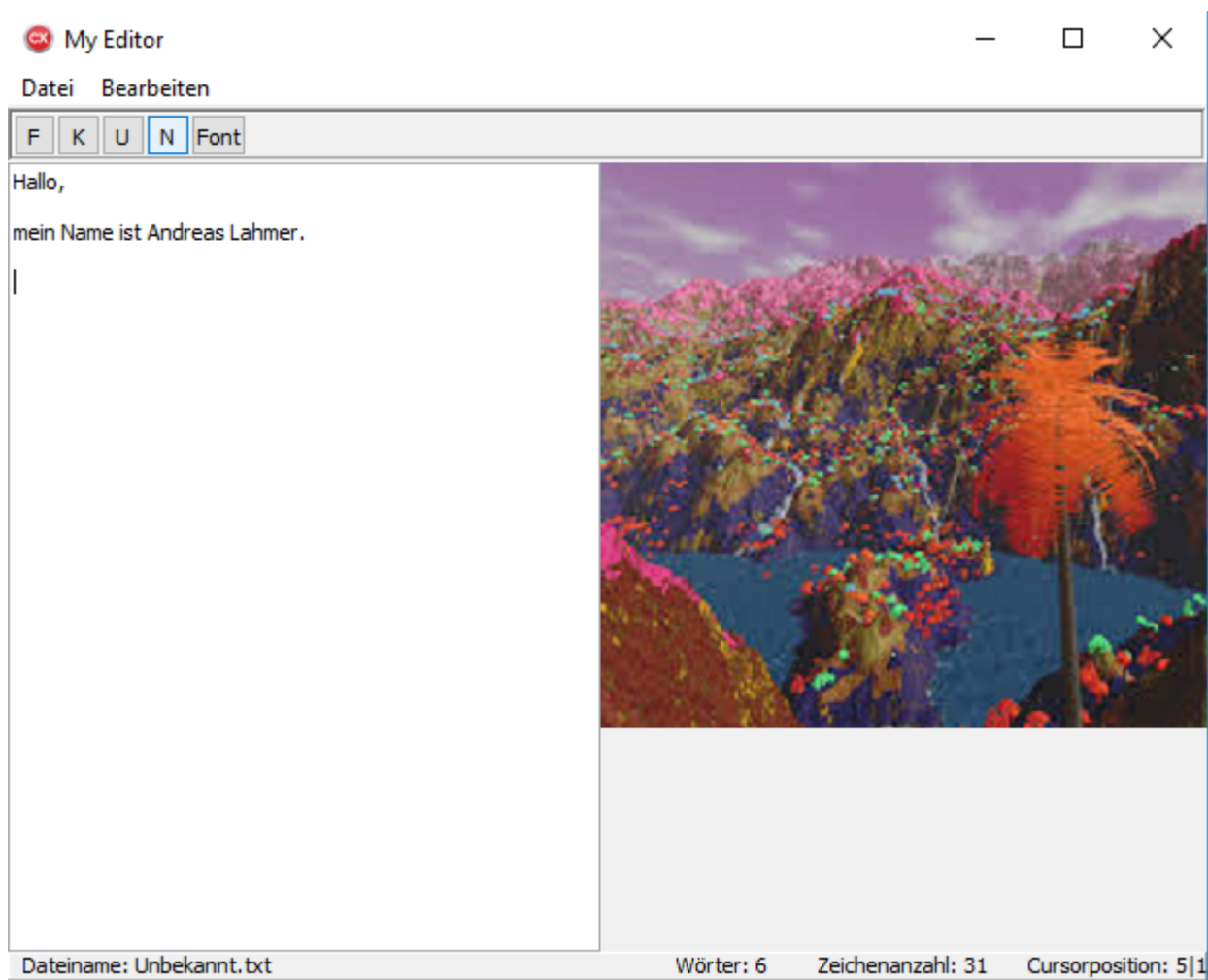
Last update: **2018/03/20 13:55**

Editor - Beispiel

Angabe

Dieses Visual C++ Programm realisiert einen Texteditor mit verschiedenen Einstellungen und Funktionen (Schriftart, Schriftformatierung, Suchen, Ersetzen, Drucken, Anzahl Zeichen, Anzahl Wörter, Cursorposition).

Screenshot



Lösung - Unit.cpp

```
//-----  
-  
  
#include <vcl.h>  
#pragma hdrstop
```

```
#include "EditorUnit1.h"
//-----
-
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
-
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
-

void __fastcall TForm1::Ende1Click(TObject *Sender)
{
    Form1->Close();
}
//-----
-

void __fastcall TForm1::Neu1Click(TObject *Sender)
{
    RichEdit1->Clear();
}
//-----
-

void __fastcall TForm1::Speichern1Click(TObject *Sender)
{
    if (SaveDialog1->Execute()) {
        RichEdit1->Lines->SaveToFile(SaveDialog1->FileName);
        Label1->Caption="Dateiname: "+SaveDialog1->FileName;
    }
}
//-----
-

void __fastcall TForm1::ffnen1Click(TObject *Sender)
{
    if (OpenDialog1->Execute()) {
        RichEdit1->Lines->LoadFromFile(OpenDialog1->FileName);
        Label1->Caption="Dateiname: "+OpenDialog1->FileName;
    }
}
//-----
-

void Zeichenanzahl()
{
    // Form1->Label2->Caption="Zeichenanzahl:
    "+IntToStr(Form1->RichEdit1->Text.Length());
```

```
int ZA=0;
for(int i=0; i<Form1->RichEdit1->Lines->Count;i++)           //Für jede
Zeile
{
    String str = Form1->RichEdit1->Lines->Strings[i].c_str();
    for(int j=0; j<str.Length();j++)                         //Für
jedes Zeichen in einer Zeile
    {
        if(Form1->RichEdit1->Lines->Strings[i].c_str()[j]!=' ')
            ZA=ZA+1;
    }
}
Form1->Label2->Caption="Zeichenanzahl: " + IntToStr(ZA);
}

void Wortanzahl()
{
    int words=0, lastchar=0;
    int laenge=0;

    //Trim() schneidet am Ende und am Anfang die Leerzeichen und Umbrüche
weg
    for(int i=0; i<Form1->RichEdit1->Text.Trim().Length();i++)
    {
        if(Form1->RichEdit1->Text.Trim().c_str()[i]==13 ||
Form1->RichEdit1->Text.Trim().c_str()[i]==32)
        {
            if(lastchar==13 || lastchar==10 || lastchar==32)
            {
                //Kein weiteres Wort, da bevor bereits ein Leerzeichen bzw.
Zeilenumbruch
            }
            else
            {
                words++;
            }
        }
        lastchar=Form1->RichEdit1->Text.Trim().c_str()[i];
    }
    words++;
    Form1->Label4->Caption="Wörter: " + IntToStr(words);
}

void __fastcall TForm1::RichEdit1Change(TObject *Sender)
{
    Zeichenanzahl();
    Wortanzahl();
}

//-----
-
```

```

void CursorPosition() {
    Form1->Label3->Caption="Cursorposition:
"+VarToStr(1+Form1->RichEdit1->CaretPos.y)+"|"+VarToStr(1+Form1->RichEdit1->
CaretPos.x);
}

void __fastcall TForm1::RichEdit1KeyUp(TObject *Sender, WORD &Key,
TShiftState Shift)
{
    CursorPosition();
}
//-----
-

void __fastcall TForm1::Ersetzen1Click(TObject *Sender)
{
    ReplaceDialog1->Position=Point(RichEdit1->Left + RichEdit1->Width,
RichEdit1->Top);
    ReplaceDialog1->Execute();
}

void __fastcall TForm1::SearchBox1Change(TObject *Sender)
{
    int FoundAt=0;
    TSearchTypes mySearchTypes = TSearchTypes();
    mySearchTypes << stWholeWord;
    FoundAt=RichEdit1->FindTextW(SearchBox1->Text, 0,
RichEdit1->Text.Length(),mySearchTypes);
    if(FoundAt!=-1)
    {
        RichEdit1->SetFocus();
        RichEdit1->SelStart=FoundAt;
        RichEdit1->SelLength=SearchBox1->Text.Length();
    }
}
//-----
-

void __fastcall TForm1::Suchen1Click(TObject *Sender)
{
    FindDialog1->Execute();
}
//-----
-

void __fastcall TForm1::FindDialog1Find(TObject *Sender)
{
    int FoundAt=0;
    TSearchTypes mySearchTypes = TSearchTypes();
    if(FindDialog1->Options.Contains(frMatchCase))

```

```
{
    mySearchTypes << stMatchCase;
}
if(FindDialog1->Options.Contains(frWholeWord))
{
    mySearchTypes << stWholeWord;
}
FoundAt=RichEdit1->FindTextW(FindDialog1->FindTextW, 0,
RichEdit1->Text.Length(),mySearchTypes);
if(FoundAt!=-1)
{
    RichEdit1->SetFocus();
    RichEdit1->SelStart=FoundAt;
    RichEdit1->SelLength=FindDialog1->FindTextW.Length();
}
}
//-----
-

void __fastcall TForm1::ReplaceDialog1Find(TObject *Sender)
{
    int FoundAt=0;
    TSearchTypes mySearchTypes = TSearchTypes();
    if(ReplaceDialog1->Options.Contains(frMatchCase))
    {
        mySearchTypes << stMatchCase;
    }
    if(ReplaceDialog1->Options.Contains(frWholeWord))
    {
        mySearchTypes << stWholeWord;
    }
    FoundAt=RichEdit1->FindTextW(ReplaceDialog1->FindTextW, 0,
RichEdit1->Text.Length(),mySearchTypes);
    if(FoundAt!=-1)
    {
        RichEdit1->SetFocus();
        RichEdit1->SelStart=FoundAt;
        RichEdit1->SelLength=ReplaceDialog1->FindTextW.Length();
    }
}
//-----
-

void __fastcall TForm1::ReplaceDialog1Replace(TObject *Sender)
{
    int FoundAt=0;
    TReplaceFlags flags;
    TSearchTypes mySearchTypes = TSearchTypes();
    if(ReplaceDialog1->Options.Contains(frMatchCase))
    {
```



```

        mySearchTypes << stMatchCase;
    }
    else
    {
        flags << rfIgnoreCase;
    }
    if(ReplaceDialog1->Options.Contains(frWholeWord))
    {
        mySearchTypes << stWholeWord;
    }
    if(ReplaceDialog1->Options.Contains(frReplaceAll))
    {
        flags << rfReplaceAll;
RichEdit1->Text=StringReplace(RichEdit1->Text,ReplaceDialog1->FindTextW,Repl
aceDialog1->ReplaceTextW,flags);
    }
    else
    {
        FoundAt=RichEdit1->FindTextW(ReplaceDialog1->FindTextW, 0,
RichEdit1->Text.Length(),mySearchTypes);
        if(FoundAt!=-1)
        {
            RichEdit1->SetFocus();
            RichEdit1->SelStart=FoundAt;
            RichEdit1->SelLength=ReplaceDialog1->FindTextW.Length();
            RichEdit1->SelText=ReplaceDialog1->ReplaceTextW;
        }
    }
}
//-----
-

void __fastcall TForm1::SpeedButton1Click(TObject *Sender)
{
    if(    RichEdit1->SelAttributes->Style.Contains(fsBold))
    {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style >>
fsBold;
        RichEdit1->SelText=RichEdit1->SelText;
    }
    else {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style <<
fsBold;
        RichEdit1->SelText=RichEdit1->SelText;
    }
}
//-----
-

void __fastcall TForm1::SpeedButton2Click(TObject *Sender)
{

```

```
    if(    RichEdit1->SelAttributes->Style.Contains(fsItalic))
    {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style >>
fsItalic;
        RichEdit1->SelText=RichEdit1->SelText;
    }
    else
    {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style <<
fsItalic;
        RichEdit1->SelText=RichEdit1->SelText;
    }
}
```

```
//-----
-
```

```
void __fastcall TForm1::SpeedButton3Click(TObject *Sender)
{
    if(    RichEdit1->SelAttributes->Style.Contains(fsUnderline))
    {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style >>
fsUnderline;
        RichEdit1->SelText=RichEdit1->SelText;
    }
    else
    {
        RichEdit1->SelAttributes->Style=RichEdit1->SelAttributes->Style <<
fsUnderline;
        RichEdit1->SelText=RichEdit1->SelText;
    }
}
```

```
//-----
-
```

```
void __fastcall TForm1::SpeedButton4Click(TObject *Sender)
{
    RichEdit1->SelAttributes->Style = TFontStyles();
    RichEdit1->SelText=RichEdit1->SelText;
}
```

```
//-----
-
```

```
void __fastcall TForm1::SpeedButton5Click(TObject *Sender)
{
    FontDialog1->Execute();
    RichEdit1->Font=FontDialog1->Font;
    RichEdit1->SelText=RichEdit1->SelText;
}
```

```
//-----
-
```

```
void RichEditToCanvas(TRichEdit* RichEdit, TCanvas *Canvas)
{
    int y=500;
    int ZA=10;
    int x=500;

    AnsiString zeile;

    for(int i=0; i<Form1->RichEdit1->Lines->Count;i++)
    {
        zeile=Form1->RichEdit1->Lines->Strings[i];
        Canvas->TextOutW(x,y,zeile);
        y=y+ZA+Canvas->TextHeight(zeile);
    }
}

void __fastcall TForm1::Drucken1Click(TObject *Sender)
{
    if(PrintDialog1->Execute())
    {
        Printer()->BeginDoc();
        RichEditToCanvas(RichEdit1, Printer()->Canvas);
        Printer()->EndDoc();
    }
}

//-----
-

void __fastcall TForm1::Bildladen1Click(TObject *Sender)
{
    OpenPictureDialog1->Execute();
    Form1->Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
}

//-----
-
```

Download

[02Editor.zip](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus:02ueb

Last update: **2018/03/20 13:48**

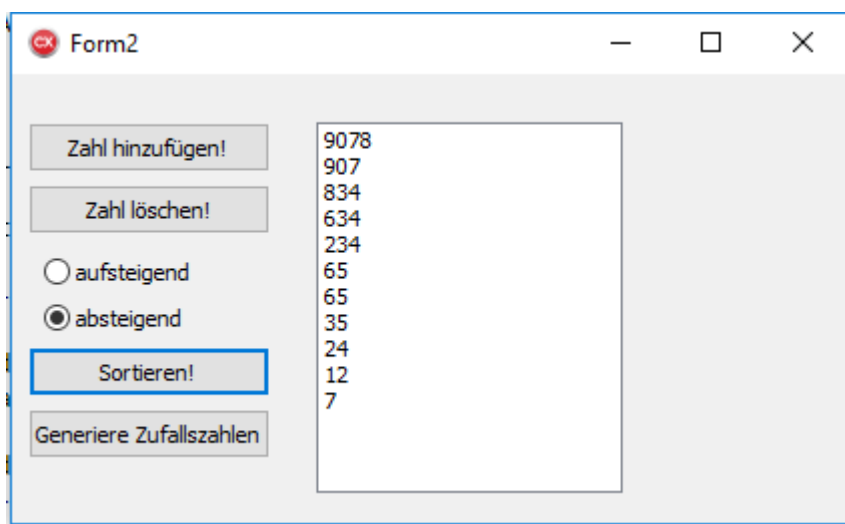


Zahlen sortieren - Beispiel

Angabe

In diesem Visual C++ Programm lernen Sie das Befüllen von einer ListBox und das Löschen einzelner Elemente. Des Weiteren lernen Sie, wie sie ein zweites Formular erstellen und dieses aufrufen bzw. Daten zwischen den beiden Formularen austauschen. Auch das Sortieren der Zahlen in einer Listbox bzw. das automatische Generieren von Zufallszahlen lernen Sie kennen.

Screenshot



Lösung

Unit2.cpp

```
//-----  
-  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit2.h"  
#include "Unit3.h"  
#include "Unit4.h"  
//-----  
-  
  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm2 *Form2;
```

```
//-----  
-  
__fastcall TForm2::TForm2(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-  
void __fastcall TForm2::Button1Click(TObject *Sender)  
{  
    Form3->Show();  
}  
//-----  
-  
void __fastcall TForm2::Button2Click(TObject *Sender)  
{  
    if(Form2->RadioButton1->Checked)  
    {  
        for(int i=0;i<Form2->ListBox1->Items->Count;i++)  
        {  
            //ShowMessage(Form2->ListBox1->Items->Strings[i]);  
            for(int j=0;j<Form2->ListBox1->Items->Count;j++)  
            {  
if(Form2->ListBox1->Items->Strings[j].ToInt()>Form2->ListBox1->Items->String  
s[i].ToInt())  
                {  
                    String hilf=Form2->ListBox1->Items->Strings[j];  
Form2->ListBox1->Items->Strings[j]=Form2->ListBox1->Items->Strings[i];  
                    Form2->ListBox1->Items->Strings[i]=hilf;  
                }  
            }  
        }  
    }  
    if(Form2->RadioButton2->Checked)  
    {  
        for(int i=0;i<Form2->ListBox1->Items->Count;i++)  
        {  
            //ShowMessage(Form2->ListBox1->Items->Strings[i]);  
            for(int j=0;j<Form2->ListBox1->Items->Count;j++)  
            {  
if(Form2->ListBox1->Items->Strings[j].ToInt()<Form2->ListBox1->Items->String  
s[i].ToInt())  
                {  
                    String hilf=Form2->ListBox1->Items->Strings[j];  
Form2->ListBox1->Items->Strings[j]=Form2->ListBox1->Items->Strings[i];  
                    Form2->ListBox1->Items->Strings[i]=hilf;  
                }  
            }  
        }  
    }  
}
```

```
}
//-----
-
void __fastcall TForm2::Button3Click(TObject *Sender)
{
    Form2->ListBox1->DeleteSelected();
}
//-----
-
void __fastcall TForm2::Button4Click(TObject *Sender)
{
    Form4->Show();
}
//-----
-
void __fastcall TForm2::ListBox1KeyPress(TObject *Sender, System::WideChar
&Key)
{
    if(int(Key)==8)
        Form2->ListBox1->DeleteSelected();
}
//-----
-
```

Unit3.cpp

```
//-----
-

#include <vcl.h>
#pragma hdrstop

#include "Unit2.h"
#include "Unit3.h"
//-----
-
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
//-----
-
__fastcall TForm3::TForm3(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
-
void __fastcall TForm3::Button1Click(TObject *Sender)
{
}
```

```
Form2->ListBox1->Items->Add(Form3->Edit1->Text);
Form3->Edit1->SetFocus();
}
//-----
-
```

Unit4.cpp

```
//-----
-

#include <vcl.h>
#include <cstdlib>
#include <time.h>

#pragma hdrstop

#include "Unit4.h"
#include "Unit2.h"
//-----
-

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
//-----
-

__fastcall TForm4::TForm4(TComponent* Owner)
: TForm(Owner)
{
}
//-----
-

void __fastcall TForm4::Button1Click(TObject *Sender)
{
    for(int i=0;i<Form4->Edit1->Text.ToInt();i++)
    {
        Form4->Memo1->Lines->Add(rand()%150);
    }
}
//-----
-

void __fastcall TForm4::Button2Click(TObject *Sender)
{
    Form4->Close();
    for(int i=0;i<Form4->Memo1->Lines->Count;i++)
    {
        Form2->ListBox1->Items->Add(Form4->Memo1->Lines->Strings[i]);
    }
}
```


//-----
-

Download

[03Forms.zip](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus:03ueb



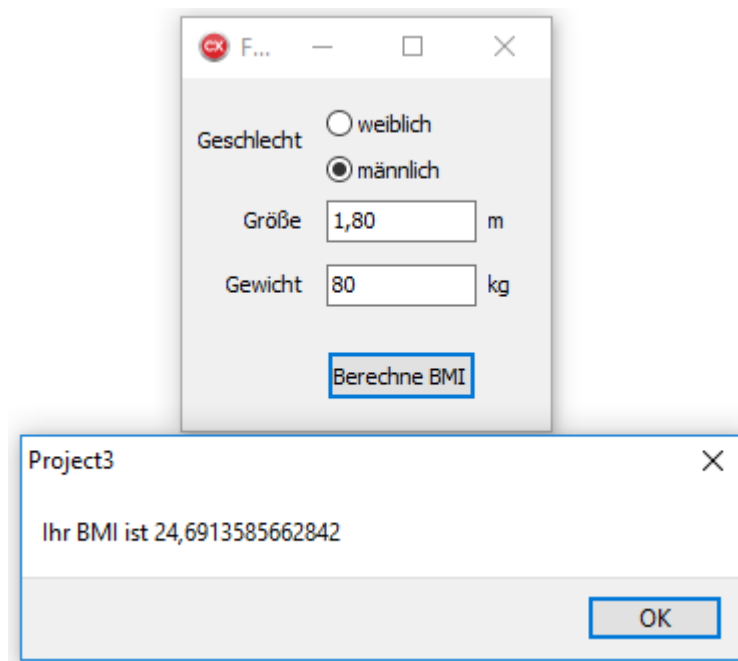
Last update: **2018/03/20 13:56**

Erstellen einer eigenen Bibliothek- Beispiel

Angabe

In diesem Visual C++ Programm lernen Sie das Erstellen und Verwenden eigener Funktionen innerhalb von Bibliotheken - sprich Header (.h) und Quellcode-Dateien (.cpp).

Screenshot



Lösung

Unit.cpp

```
//-----  
-  
  
#include <vcl.h>  
  
#pragma hdrstop  
  
#include "MyTools.h"  
#include "Unit1.h"  
//-----  
-  
  
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
-
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
-
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    float bmi=0.0, gewicht=0.0, groesse=0.0;
    //BMI=Gewicht/(Größe^2)

    if(Edit1->Text.Length()>0 && Edit2->Text.Length()>0)
    {
        groesse=StrToFloat(Edit1->Text);
        gewicht=StrToFloat(Edit2->Text);

        if(RadioButton1->Checked)           //weiblich
        {
            if(groesse!=0)
            {
                bmi=gewicht/(groesse*groesse);
                ShowMessage("Ihr BMI ist " + FloatToStr(bmi));
                if(bmi>=40)
                    ShowMessage("Sie haben starke Adipositas!");
                else if(bmi>=31)
                    ShowMessage("Sie haben Adipositas!");
                else if(bmi>=25)
                    ShowMessage("Sie haben Übergewicht!");
                else if(bmi>=19)
                    ShowMessage("Sie haben Normalgewicht!");
                else if(bmi<19)
                    ShowMessage("Sie haben Untergewicht!");
            }
            else
            {
                ShowMessage("Geben Sie eine gültige Groesse ein!");
            }
        }
        else if (RadioButton2->Checked)     //männlich
        {
            if(groesse!=0)
            {
                bmi=gewicht/(groesse*groesse);
                ShowMessage("Ihr BMI ist " + FloatToStr(bmi));
                if(bmi>=40)
                    ShowMessage("Sie haben starke Adipositas!");
            }
        }
    }
}
```

```
        else if(bmi>=31)
            ShowMessage("Sie haben Adipositas!");
        else if(bmi>=26)
            ShowMessage("Sie haben Übergewicht!");
        else if(bmi>=20)
            ShowMessage("Sie haben Normalgewicht!");
        else if(bmi<20)
            ShowMessage("Sie haben Untergewicht!");
    }
    else
    {
        ShowMessage("Geben Sie eine gültige Groesse ein!");
    }
}
else
{
    ShowMessage("Es wurde kein Radiobutton gedrückt!");
}
}
}

//-----
-

void __fastcall TForm1::Edit1KeyPress(TObject *Sender, System::WideChar
&Key)
{
    checkFloatInput(Edit1, Key);
}
//-----
-
```

MyTools.h

```
void checkFloatInput(System::WideChar &Key, TEdit *Eingabefeld);
```

MyTools.cpp

```
#include <vcl.h>

void checkFloatInput(System::WideChar &Key, TEdit *Eingabefeld)
{
    //int(Key)>=AsciiWertvon 0
    if((Key>='0' && Key<='9') || (int)Key==8 || Key==',')
    {
        //Eingabe ist ok
        if(Key==',')

```

```
{
    if(Eingabefeld->Text.Pos(Key)>0)
    {
        Key=0;
    }
}
else
{
    Key=0;
}
}
```

Download

[04Bibliothek.zip](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus:04ueb



Last update: **2018/03/20 14:03**

Quadratische Gleichungen

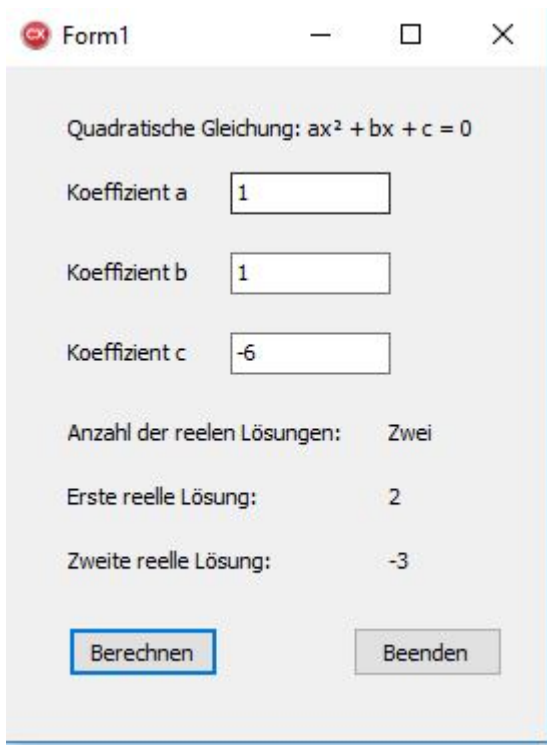
Angabe

Erstelle ein Projekt, mit welchem quadratische Gleichungen gelöst werden können. (siehe Beispieldatei.) Die Koeffizienten a , b und c der quadratischen Gleichung $ax^2+bx+c=0$ sollen in Edit-Feldern eingegeben werden können. Durch Klick auf den Button „Berechnen“ werden je nach Diskriminante ($D=b^2-4ac$) keine Lösungen, eine oder zwei Lösungen ausgegeben.

Hinweis: Verwende `#include <math.h>` Damit kannst du die Wurzel `sqrt(b*b-4*a*c)` verwenden. Achte darauf, dass jeder Fall funktioniert: z.B. $a=1, b=1, c=-6 \rightarrow$ Zwei Lösungen $x_1= 2,00$ und $x_2=-3,00$
 $a=1, b=1, c=1 \rightarrow$ Keine Lösungen $a=1, b=-4, c=4 \rightarrow$ Eine Lösung $x_1=2,00$

Stelle mittels eigens eingebundener Bibliothek sicher, dass nur Zahlen (inkl. Komma) eingegeben werden dürfen, dass die Berechnung nur dann erfolgt, wenn die Edit-Felder nicht leer sind und dass a nicht 0 sein darf!

Screenshot



The screenshot shows a C++ application window titled "Form1". Inside the window, there is a text label "Quadratische Gleichung: $ax^2 + bx + c = 0$ ". Below this, there are three input fields for coefficients: "Koeffizient a" with value "1", "Koeffizient b" with value "1", and "Koeffizient c" with value "-6". Below the input fields, there are three lines of output: "Anzahl der reellen Lösungen: Zwei", "Erste reelle Lösung: 2", and "Zweite reelle Lösung: -3". At the bottom, there are two buttons: "Berechnen" (highlighted with a blue border) and "Beenden".

Lösung

Unit.cpp

```
//-----  
-  
  
#include <vcl.h>  
#pragma hdrstop  
#include <math.h>  
  
#include "Unit1.h"  
//-----  
-  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
-  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    float a=0.0, b=0.0, c=0.0;  
  
    a=StrToFloat(Edit1->Text);  
    b=StrToFloat(Edit2->Text);  
    c=StrToFloat(Edit3->Text);  
  
    if(b*b-4*a*c>0)  
    {  
        Label8->Caption="Zwei";  
        Label9->Caption=(-b+sqrt(b*b-4*a*c))/(2*a);  
        Label10->Caption=(-b-sqrt(b*b-4*a*c))/(2*a);  
    }  
    if(b*b-4*a*c==0)  
    {  
        Label8->Caption="Eine";  
        Label9->Caption=-b/2*a;  
        Label10->Caption="- ";  
    }  
    if(b*b-4*a*c<0)  
    {  
        Label8->Caption="Keine";  
        Label9->Caption="- ";  
        Label10->Caption="- ";  
    }  
}  
  
//-----  
-
```

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Close();
}
//-----
-
```

Download

[05QuadratischeGleichungen.zip](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi_201718:4_visualcplusplus:05ueb



Last update: **2018/04/06 13:30**