

# Grundlagen in C# und .net

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- [mwilhelm@hs-harz.de](mailto:mwilhelm@hs-harz.de)
- Raum 2.202
- Tel. 03943 / 659 338

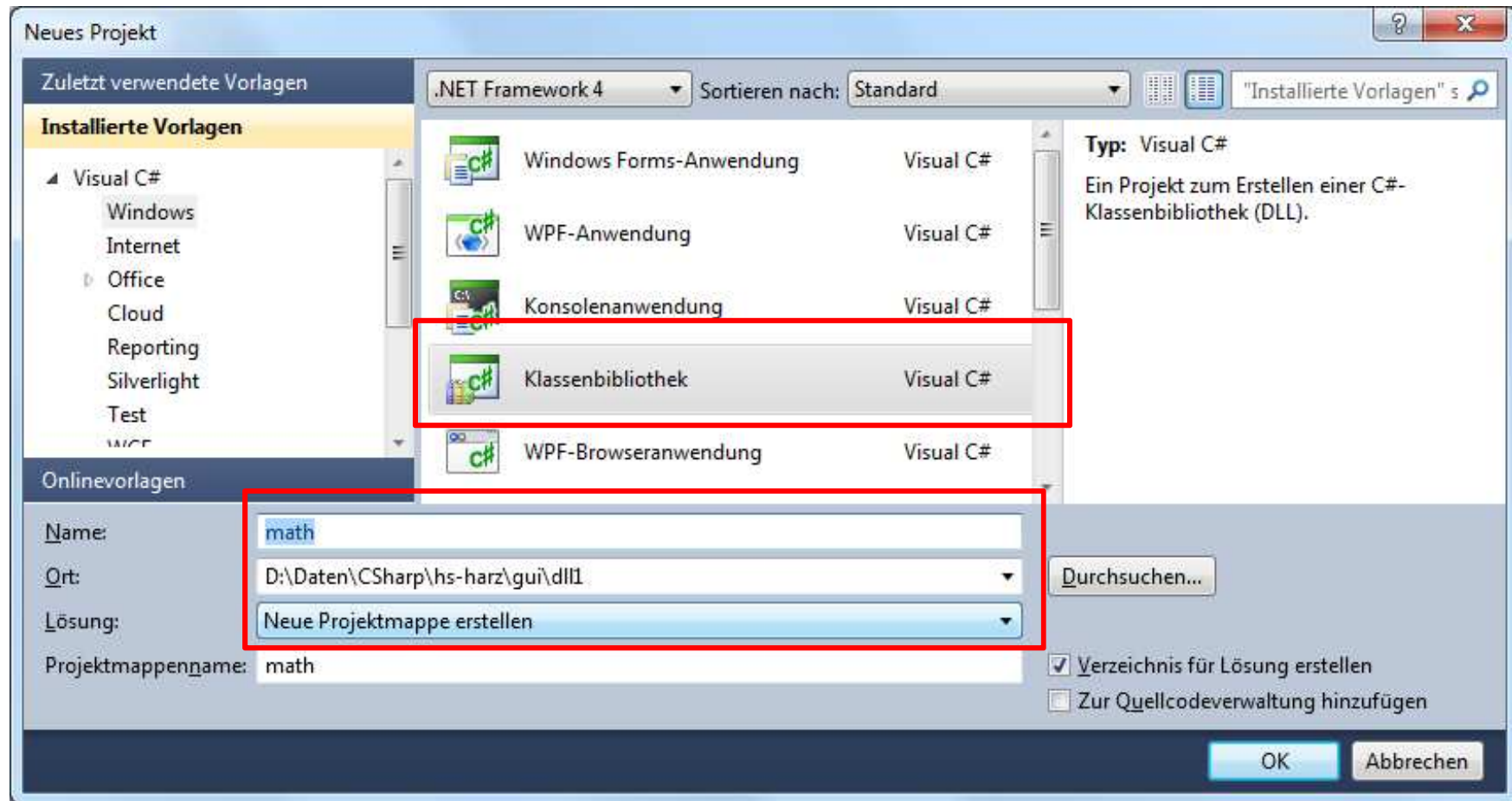
# Inhalt

- TabbedPane (Register)
- ListView
- Tree
- Tabelle
- MDI-Programme
- Erweiterte Grafik
- Threads und Semaphore
- Datenbanken
- **DLL**

# Dynamik Link Library

- Gemeinsamen Code in ein Projekt zusammenfassen
  - Gemeinsame Verwendung in mehreren Projekten
  - Spart Speicherplatz auf der Festplatte
  - Spart Speicherplatz im Hauptspeicher
  - Wird erst auf „Aufforderung“ in den Hauptspeicher geladen
  - Ideal für mehrere Teams
  - Debuggen ist aber schwieriger
  - Statische und dynamische DLL (je nach Methoden)
- Nachteile
  - Versionsprobleme
- Steigerung als COM-Objekte

# 1. DLL Projekt erstellen: gemeinsamer Ordner



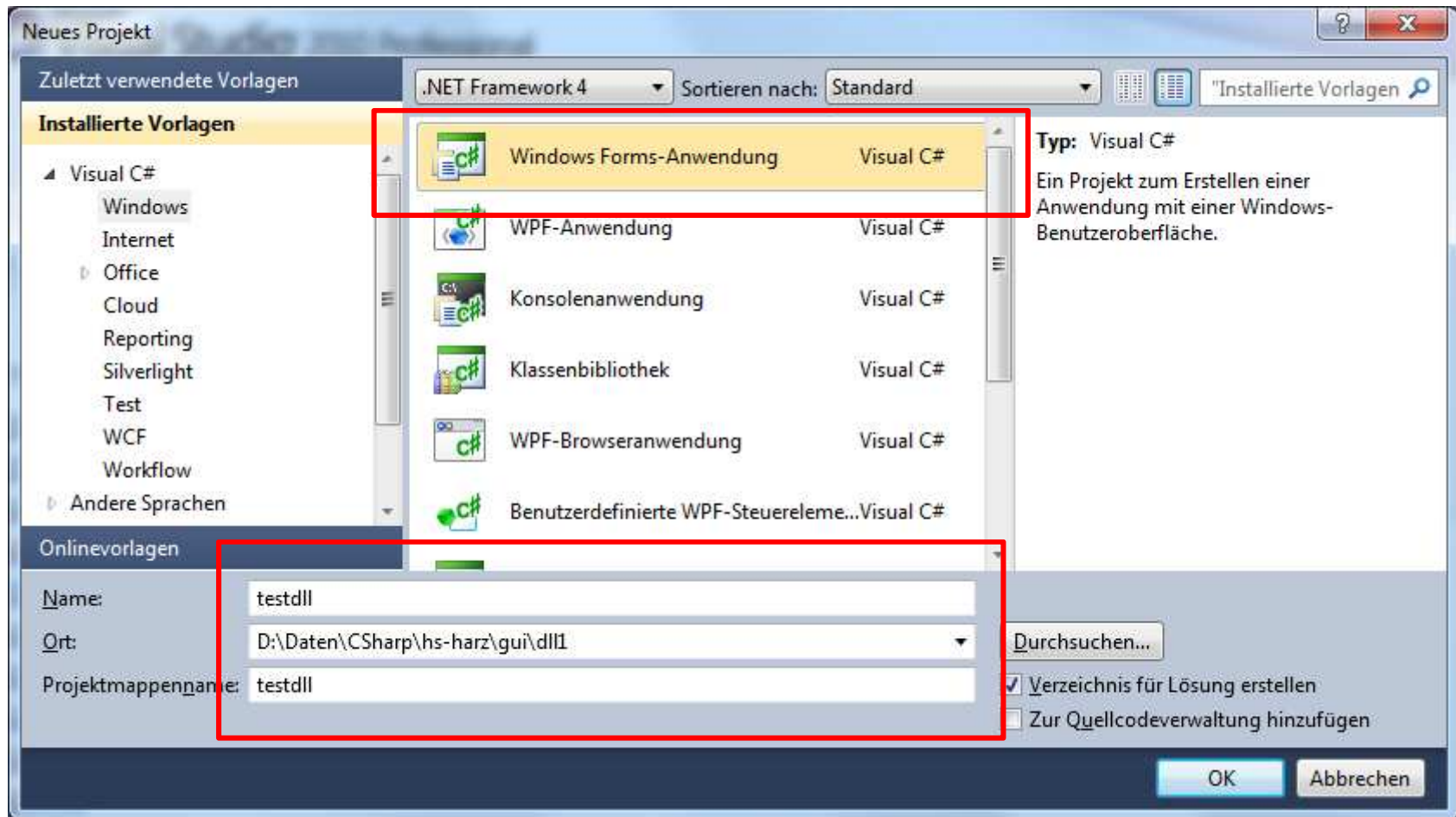
Umbenennen der Klasse „class1“ nach math

# DLL-Quellcode

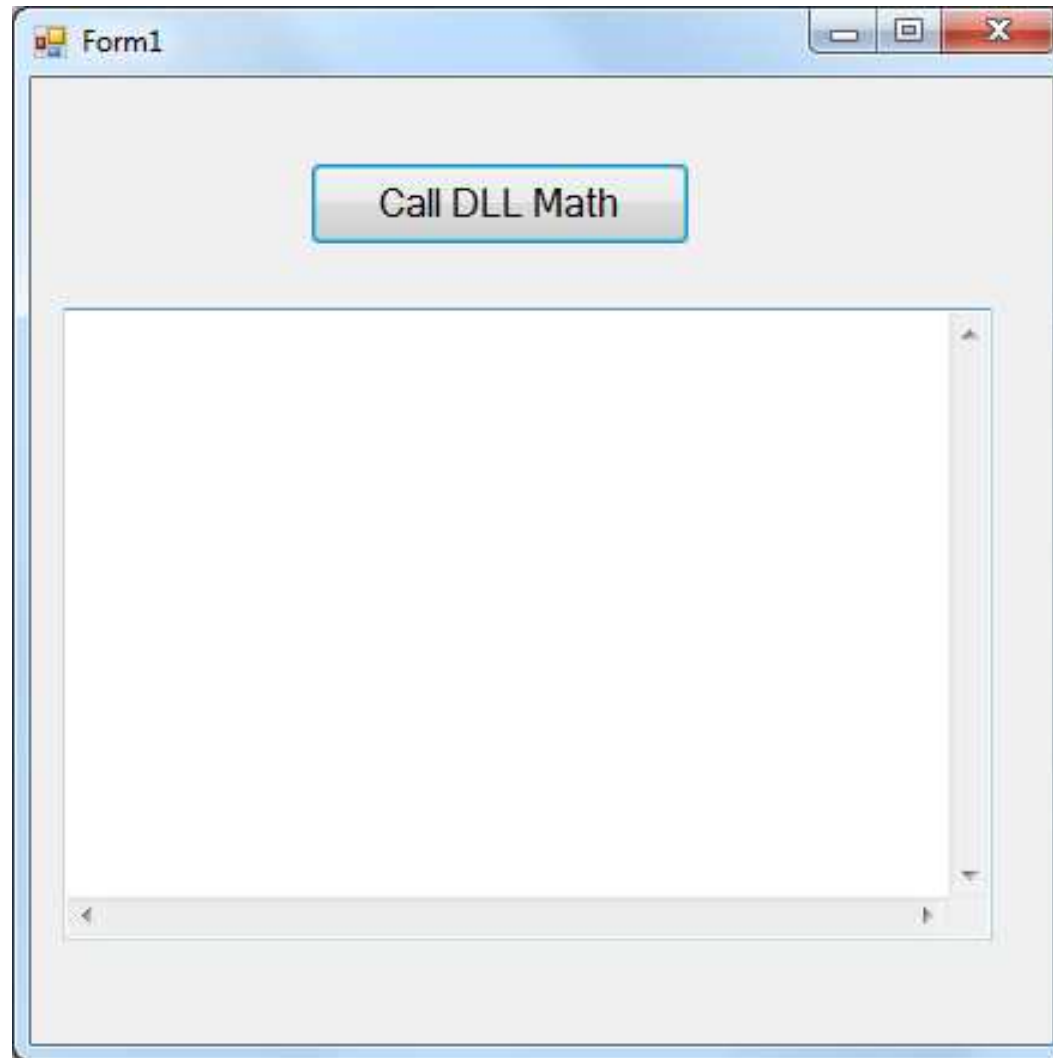
```
namespace math
{
    public class Math
    {
        public static long Add(long i, long j)
        {
            return (i + j);
        }
        public long Mult(long i, long j)
        {
            return (i * j);
        }
    }
}
```

# Aufrufprogramm erstellen: gemeinsamer Ordner

## Projektname: testdll



# Dialog aufbauen



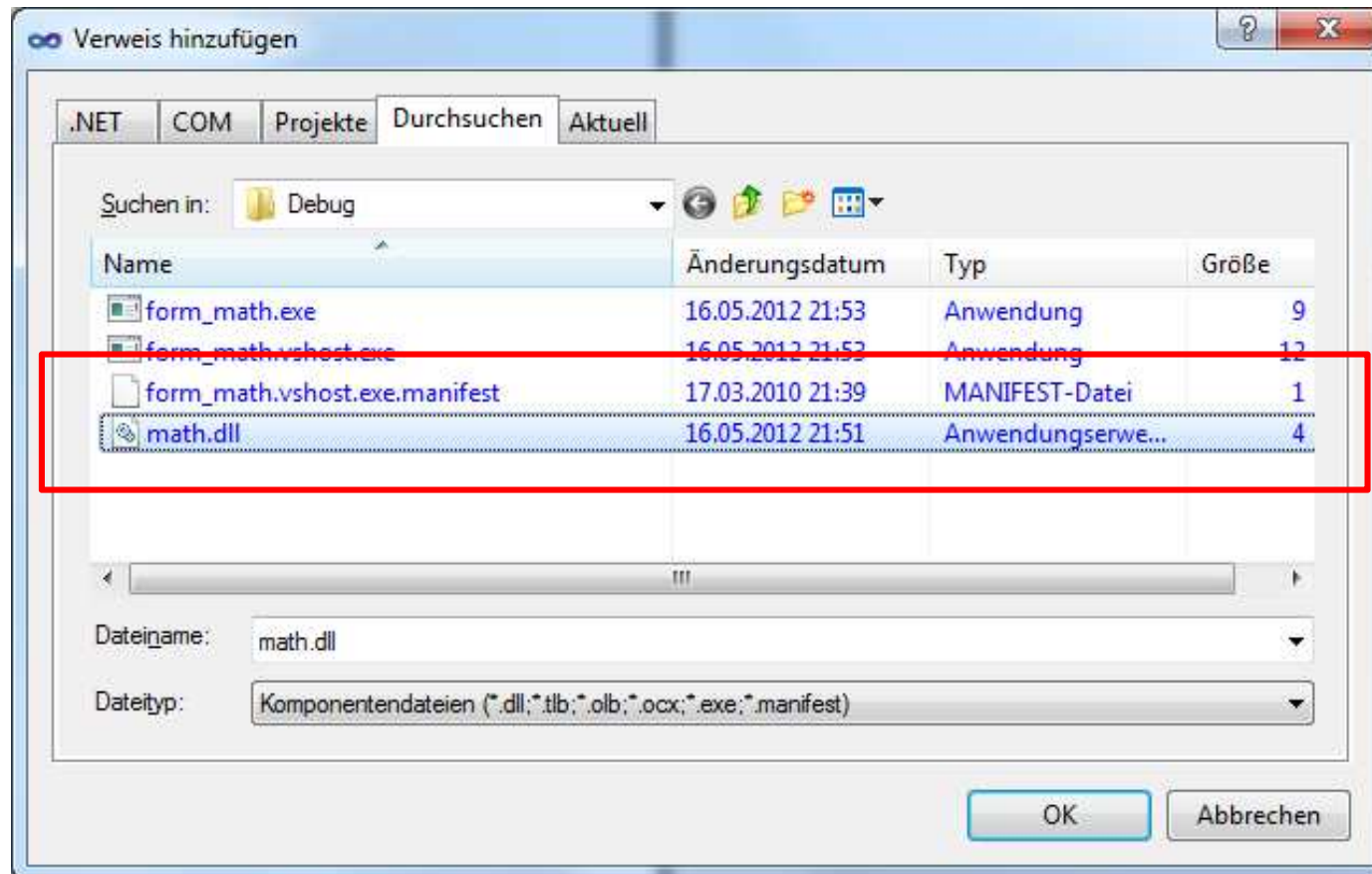
using math;

```
private void BnCallDLL_Click(object sender, EventArgs e)
{
    long a, b, c;
    a = 12;
    b = 13;
    textBox1.Text="";
    c = math.Math.Add(a, b);           // wenn eindeutig Math.Add(a, b);
    textBox1.AppendText( "Add C: " + c.ToString()+"\r\n" );
}
}
```



- Menü Ansicht: Projektmappen-Explorer

- Eintrag Verweise
- Verweis hinzufügen



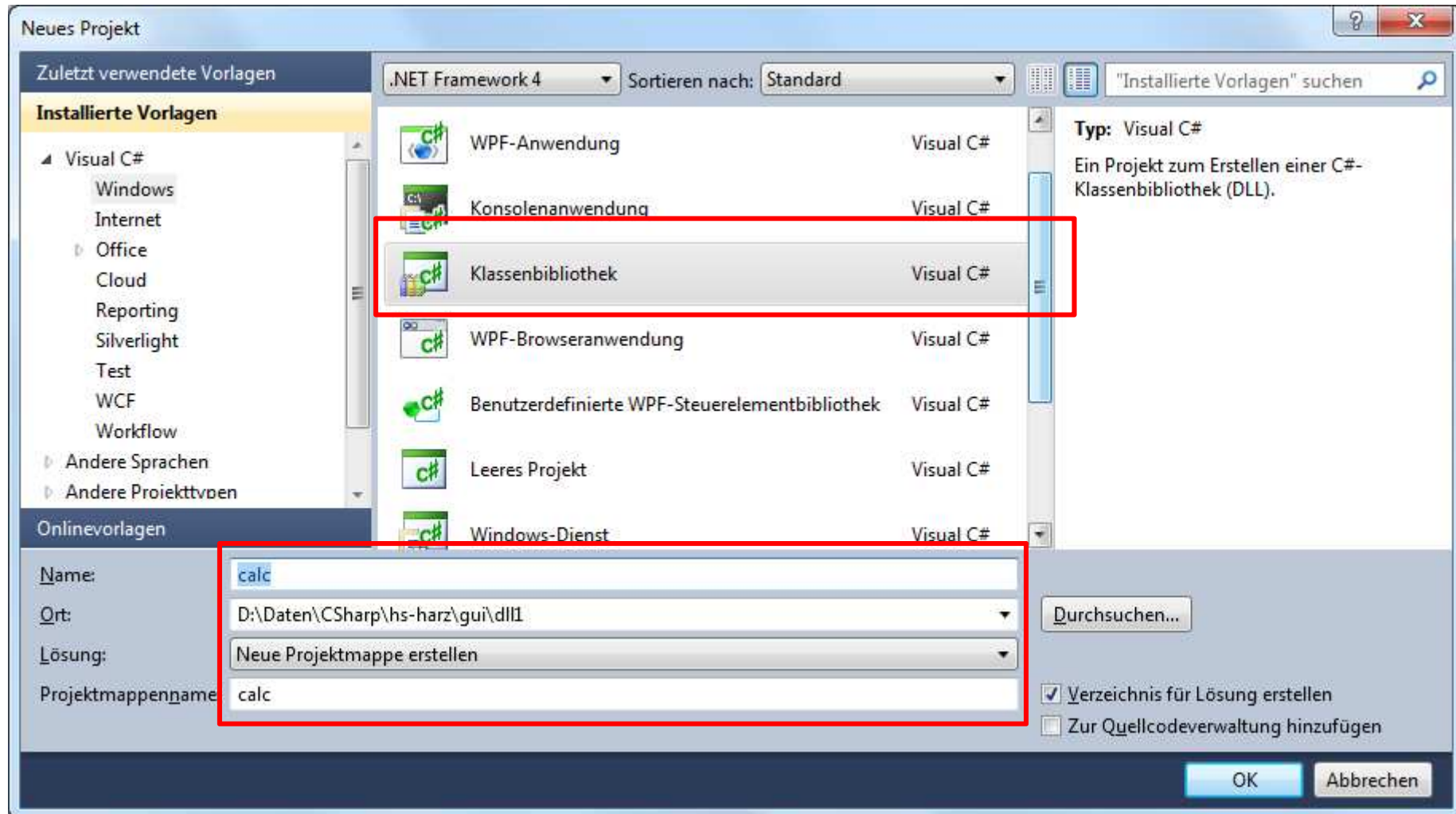
using math;

```
private void BnCallDLL_Click(object sender, EventArgs e)
{
    long a, b, c;
    a = 12;
    b = 13;
    textBox1.Text="";
    c = math.Math.Add(a, b);           // wenn eindeutig Math.Add(a, b);
    textBox1.AppendText( "Add C: " + c.ToString()+"\r\n" );
    c = math.Math.Mult(a, b);
    textBox1.AppendText( "Mult: C: " + c.ToString()+"\r\n" );
}
}
}
```

using math;

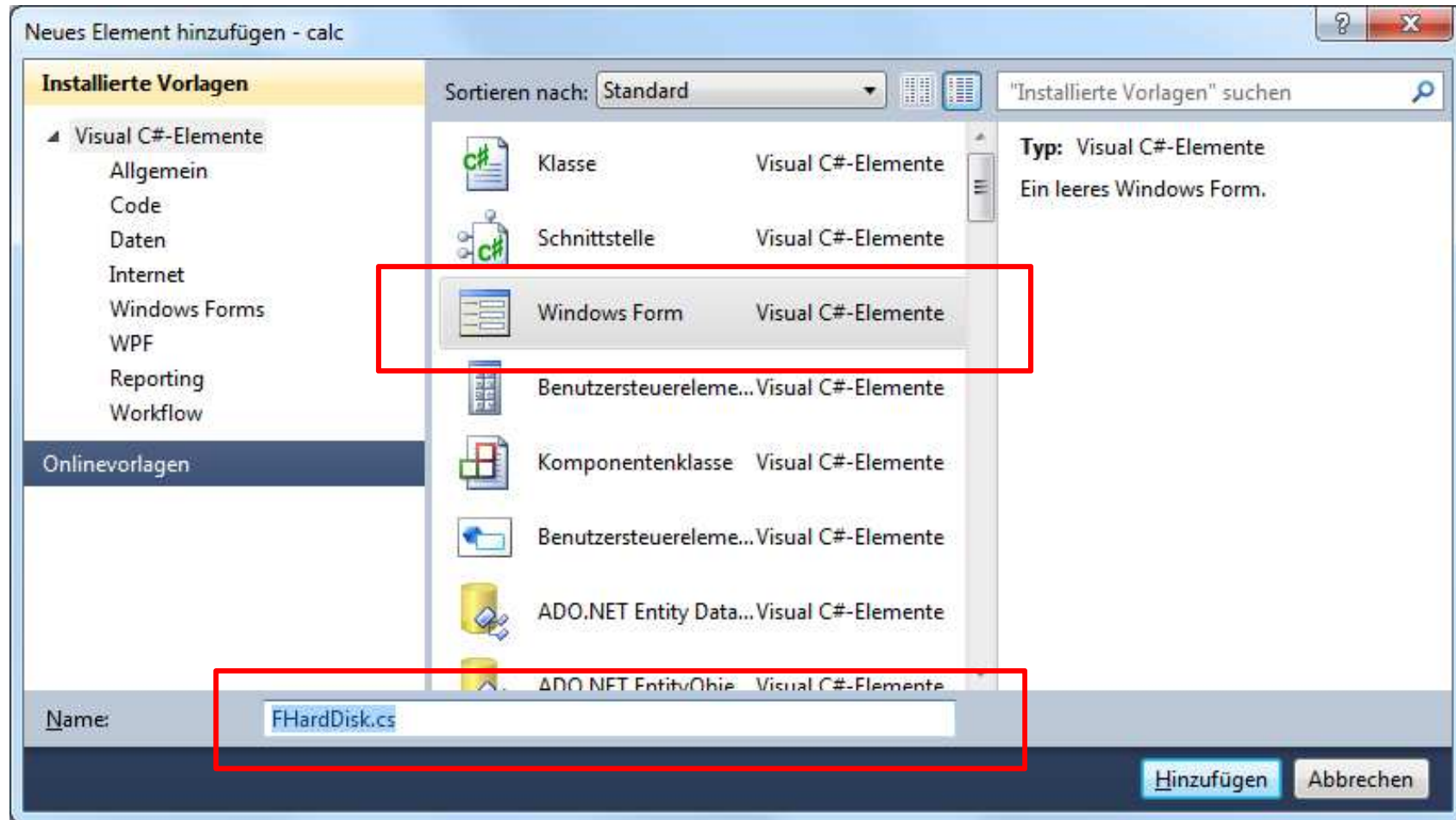
```
private void BnCallDLL_Click(object sender, EventArgs e)
{
    long a, b, c;
    a = 12;
    b = 13;
    textBox1.Text="";
    c = math.Math.Add(a, b);
    textBox1.AppendText( "Add  C: " + c.ToString()+"\r\n" );
    math.Math m = new math.Math();
    c = m.Mult(a, b);
    textBox1.AppendText( "Mult: C: " + c.ToString()+"\r\n" );
}
}
}
```

## 2. DLL Projekt erstellen: gemeinsamer Ordner

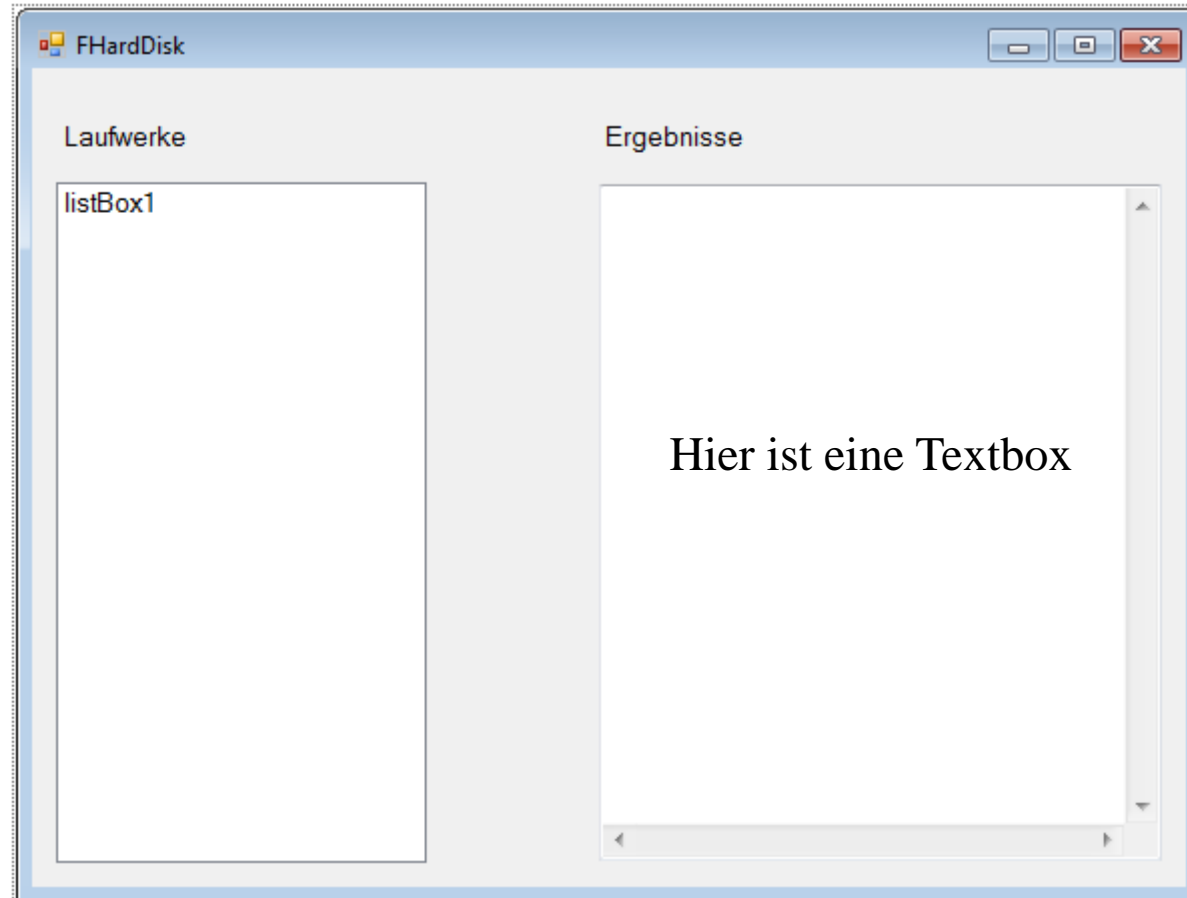


Umbenennen der Klasse „class1“ nach form1

# Interne Form erstellen



# Interne Form erstellen



# Form Harddisk

## ■ Methode FHardDisk\_load

- Eintragen der vorhandenen Laufwerke
  - `DriveInfo[] lws = System.IO.DriveInfo.GetDrives`
  - `String[] lws = System.Environment.GetLogicalDrives();`
- Ausgabe der Daten des Rechners in der TextBox
  - `System.Environment.GetFolderPath(Environment.SpecialFolder.CommonDocuments)`
  - `System.Environment.GetFolderPath(Environment.SpecialFolder.CommonPictures)`
  - `System.Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFiles)`
  - `System.Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFilesX86)`
  - `System.Environment.GetFolderPath(Environment.SpecialFolder.CommonPrograms)`
  - `System.Environment.MachineName`
  - `System.Environment.OSVersion`
  - `System.Environment.SystemDirectory`
  - `System.Environment.UserName`
  - `"CPU´s:" + System.Environment.ProcessorCount`

# Form Harddisk

## ■ Methode SeletedIndex Change

- Index vorhanden
- Holen des DriveInfo
- Ausgabe der Daten des Laufwerkes in der TextBox
  - "Name: "+lw.Name
  - "Typ: "+lw.DriveType.ToString()  
CDROM, REMOVABLE, FIXED, NETWORK, RAMDISK, UNKNOWN.
  - "Lable: "+lw.VolumeLabel
  - "Free: "+(lw.AvailableFreeSpace / 1024 / 1024).ToString()+" MB "
  - "Total: ...
  - „Used: ...



# Externe DLL, (C++, Delphi, Assembler), aufrufen

```
using System;
```

```
using System.Runtime.InteropServices;
```

```
class Program
```

```
{
```

```
    [DllImport("DP_BSP.dll")]
```

```
    static extern double Add(double a, double b);
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine( Add( 23.6, 18.4 ) );
```

```
    }
```

```
}
```