

# Starthilfe für C#

---

## Inhaltsverzeichnis

Allgemeines .....	2
Bezugsquellen.....	2
SharpDevelop .....	2
.NET Runtime.....	2
.NET SDK .....	2
Installation.....	2
Reihenfolge.....	2
Vorschlag für eine Ordnerstruktur .....	3
Arbeit mit Projekten.....	3
Anlegen eines neuen Projekts .....	3
Beispiel Code: „Hello World“ .....	5
Zusicherungen unter C# .....	5
Kompilieren des Projektes.....	6
Arbeit mit der Entwicklungsumgebung.....	6
Ausführen der Anwendung .....	6
Klassenbrowser .....	6

## Allgemeines

Zur Programmierung in C# wird Sharp Develop empfohlen. SharpDevelop ist eine freie Entwicklungsumgebung und die IDE ist sogar quelloffen verfügbar.

Zur Benutzung der Entwicklungsumgebung wird die Installation der Laufzeitumgebung .NET 2.0 vorausgesetzt. Empfohlen wird außerdem das Software Development Kit von Microsoft, welches noch zusätzliche Werkzeuge, Dokumentationen und Beispiele enthält.

## Bezugsquellen

### SharpDevelop

Die Entwicklungsumgebung ist momentan in der Version 3.0 vorhanden und kann von der folgenden Seite bezogen werden:

<http://www.icsharpcode.net/OpenSource/SD/Download/#SharpDevelop30>

Die Installationsdatei umfasst etwa 18 MB. Für die Bearbeitung der Aufgaben kann auch die Version 2.0 genutzt werden. Diese umfasst lediglich 8 MB

### .NET Runtime

Die .NET Laufzeitumgebung kann über den untenstehenden Link direkt von Microsoft bezogen werden und umfasst etwa 2.8 MB:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=AB99342F-5D1A-413D-8319-81DA479AB0D7&displaylang=de>

### .NET SDK

Das Software Development Kit kann von Microsoft über den untenstehenden Link direkt von Microsoft bezogen werden und umfasst etwa 345 MB:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=FE6F2099-B7B4-4F47-A244-C96D69C35DEC&displaylang=de>

## Installation

### Reihenfolge

Nach der Installation der Runtime (s.o.) kann SharpDevelop direkt installiert werden. Oder man will die Vorteile des SDKs nutzen und installiert in dieser Reihenfolge:

- 1.) Runtime
- 2.) SDK
- 3.) SharpDevelop

Am Schluss der Installation fragt ein Dialog ob eine „Code Completion“ Datenbank angelegt werden soll. Das ist sinnvoll und sollte bewusst durchgeführt werden.

## Vorschlag für eine Ordnerstruktur

Der folgende Vorschlag für eine Ordnerstruktur ist mit Laufwerksangaben der gestellten Rechner versehen. Diese können jedoch auch auf das eigene System (Notebook) mit kleineren Anpassungen übertragen werden.

1. Wie im Abschnitt „**Fehler! Verweisquelle konnte nicht gefunden werden.**“ bereits erwähnt sollten Sie einen Ordner nur für C# - Projekte erstellen.

**N: /inf3praktikum/csharp/**

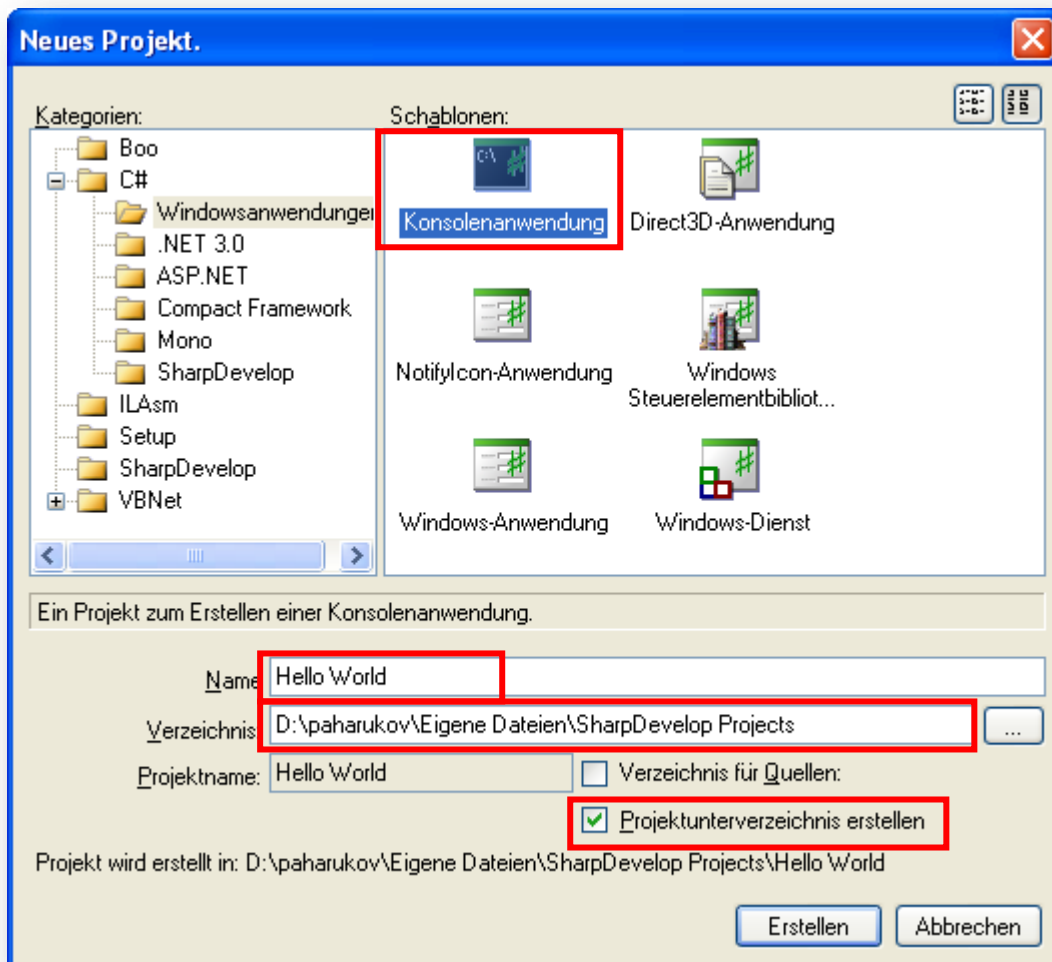
Diesen Ordner sollten Sie nun immer unter dem Punkt „Verzeichnis“ (siehe Abschnitt „Anlegen eines neuen Projekts“) angeben. Weitere Schritte sind nicht Notwendig das Projektmanagement wird vollständig von der IDE übernommen.

## Arbeit mit Projekten

C# arbeitet, wie Visual Eiffel, mit Projektstrukturen. Diese werden hier als Arbeitsmappen bezeichnet. Ebenfalls Eiffel ähnlich ist die Auswahl eines Projekttyps, mit dem Unterschied dass hier viel mehr Anwendungsbereiche abgedeckt werden als bei Visual Eiffel.

### Anlegen eines neuen Projekts

Zum Anlegen eines neuen Projekts klickt man in der IDE auf „Neu -> Projektmappe“. Es erscheint ein Dialogfenster in dem Sie aufgerufen werden der Projektmappe einen Namen zu geben und auszuwählen welche Art von Anwendung sie erzeugen möchten.



**Abbildung 1: C# Anlegen eines neuen Projekts (Hello World)**

Alle anderen Einstellungen können im Normalfall so belassen werden wie sie sind.

Sollten Sie nicht am eigenen Notebook arbeiten, passen Sie bitte das Projektverzeichnis so wie im Abschnitt „Vorschlag für eine Ordnerstruktur“ erklärt.

Weitere Einstellungen in den Projektoptionen sind nicht nötig da diese den Ansprüchen des I3-Praktikums vollkommen genügen.

### Beispiel Code: „Hello World“

Der folgende Codeabschnitt wird von SharpDevelop automatisch erzeugt und wird hier nicht weiter erläutert.

```
using System;

namespace Hello_World
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");

            // TODO: Implement Functionality Here

            Console.Write("Press any key to continue . . . ");
            Console.ReadKey(true);
        }
    }
}
```

### Zusicherungen unter C#

Um Zusicherungen auch unter C# verwenden zu können muss in der jeweiligen Klasse zunächst eine weitere Systembibliothek eingebunden werden. Diese ist in der IDE bereits enthalten. Das Einbinden würde folgendermaßen aussehen:

```
using System.Diagnostics;
```

Der konkrete Befehl lautet dann

```
Debug.Assert()
```

Dieser kann in folgenden Formen aufgerufen werden:

```
Debug.Assert(Boolean)
Debug.Assert(Boolean, String)
Debug.Assert(Boolean, String, String)
```

Hierzu ein Ausschnitt aus der Schnittstellenbeschreibung:

Name	Description
<b>Debug.Assert (Boolean)</b>	Checks for a condition and outputs the call stack if the condition is <b>false</b> .  Supported by the .NET Compact Framework.
<b>Debug.Assert (Boolean, String)</b>	Checks for a condition and displays a message



INF

Studiengang  
Medien- und  
Kommunikationsinformatik



Hochschule Reutlingen

Reutlingen University

Medien- und Kommunikationsinformatik (B.Sc.)  
Alexander Paharukov

### Informatik 3 Praktikum

---

	if the condition is <b>false</b> .
	Supported by the .NET Compact Framework.
<code>Debug.Assert (Boolean, String, String)</code>	Checks for a condition and displays both specified messages if the condition is <b>false</b> .
	Supported by the .NET Compact Framework.

#### Kompilieren des Projektes

Zum Übersetzen des Projekts klicken Sie auf „Erstellen ->Erstelle Mappe neu“ oder drücken Sie „ALT + F8“. Es wird empfohlen das Projekt immer komplett neu zu erstellen um Kompilierungsfehler zu vermeiden.

Bei diesem Vorgang werden in Ihrem Projektordner zwei zusätzliche Ordner erstellt. Dabei enthält der Ordner „bin -> Debug“ die .exe- Datei, die zum Ausführen der Anwendung benötigt wird.

#### Arbeit mit der Entwicklungsumgebung

##### Ausführen der Anwendung

Navigieren Sie in der „CMD“ zu Ihrem Projektordner. Hier befindet sich nun der Ordner „bin“ der einen Unterordner „Debug“ enthält. Navigieren Sie zum Debug-Ordner und führen Sie hier die <Projektname>.exe aus.

##### Klassenbrowser

Die IDE verfügt über einen integrierten Klassenbrowser, der hier als „Dynamische Hilfe“ bezeichnet wird. Dieser kann über „Hilfe -> Dynamische Hilfe“ eingeblendet werden. Um Informationen über ein Objekt aufzurufen genügt es den Text-Cursor in den Bezeichner des Gewünschten Typs zu platzieren. Die Informationen werden dann in der Dynamischen Hilfe angezeigt.

**Vorschläge zur Verbesserung dieser Starthilfe nehme ich gerne entgegen.**