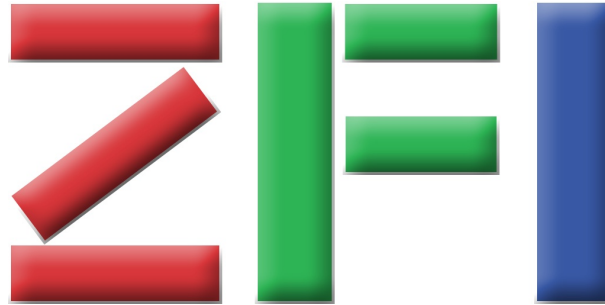


Kurs-Dokumentation



Zentrum für Informatik ZFI AG

Programming with C# using MS .NET Framework 4.0 (NPC1)

<http://www.zfi.ch/NPC1>

Weitere Infos finden Sie unter www.zfi.ch oder via Adresse:

Zentrum für Informatik ZFI AG
Zentralsekretariat
Rütistrasse 28
CH-8952 Zürich-Schlieren
Telefon: 044 732 40 00
Telefax: 041 530 31 68

Zürich, Basel, Bern, ZÄ¼rich, Schweiz

Titel	Programming with C# using MS .NET Framework 4.0
Untertitel	Einführung ins .NET Framework mit der Sprache C#
Einleitung	<p>Mit Visual Studio 2008/2010 hat Microsoft die Möglichkeit zur Erstellung von Windows-Programmen nochmals erheblich verbessert. Damit können die Bedürfnisse der Anwender noch besser als früher umgesetzt werden. Die dazu notwendigen Technologien sind im .NET Framework vorhanden. Die (neue) Sprache Visual C# wird dabei optimal unterstützt und in diesem Kurs als Programmiersprache für die Beispiele verwendet. Dieser Kurs vermittelt dem C#/C++-Programmierer einen erweiterten Einblick in .NET und bereitet den Teilnehmer damit optimal auf den professionellen Einsatz als Programmierer von C#.NET-Applikationen vor. Es wird mit der jeweils aktuellen Programmversion gearbeitet. Ein Muss für den angehenden .NET Entwickler. Dieses neue Microsoft-Seminar berücksichtigt bereits auch die Neuerungen des .NET Frameworks 4.0. Dieser Kurs wird durch einen Microsoft-zertifizierten Kursleiter geleitet.</p>
Ihr Nutzen	<p>After completing this course, students will be able to:</p> <ul style="list-style-type: none"> - Explain the purpose of the .NET Framework, and understand how to use C# and Visual Studio 2010 to build .NET Framework applications. - Understand the syntax of basic C# programming constructs. - Create and call methods in a C# application. - Catch, handle and throw exceptions. - Perform basic file IO operations in a C# application. - Create and use new types (enumerations, classes, and structures), and understand the differences between reference types and value types. - Control the visibility and lifetime of members in a type. - Use inheritance to create new reference types. - Manage the lifetime of objects and control the use of resources. - Define properties and indexers to encapsulate data, and define operators for this data. - Decouple an operation from the method that implements an operation, and use these decoupled operations to handle asynchronous events. - Use collections to aggregate data, and use Generics to implement type-safe collection classes, structures, interfaces, and methods. - Implement custom collection classes that support enumeration. - Query in-memory data by using LINQ. - Integrate code written by using a dynamic language such as Ruby and Python, or technologies such as COM, into a C# application
Voraussetzungen	<p>Before attending this course, students must have:</p> <ul style="list-style-type: none"> - At least 12 months experience working with an Object Oriented language - Have C++ or Java knowledge: <ul style="list-style-type: none"> - Creating Classes - Inheritance and Abstraction - Polymorphism - Interfaces - Exceptions - Knowledge of the Visual Studio IDE.

Teilnehmerkreis	This course is intended for experienced developers who already have programming experience in C, C++, Visual Basic, or Java and understand the concepts of object-oriented programming. This course is not designed for new programmers; it is targeted at professional developers with at least 12 months experience of programming in an object-oriented environment.
Unterlagen	Original Microsoft-Kursunterlagen (englischsprachig, nach Wunsch und Verfügbarkeit deutschsprachig)
Folgekurse	- «Advanced Foundations of Microsoft .NET Development» (Fortgeschrittene Themen) (NCWI) - «Introduction to Web Development with MS Visual Studio 2010» (NWEB)
Inhalt	- Introducing C# and the .NET Framework - Introduction to the .NET Framework Creating Projects Within Visual Studio 2010 Writing a C# Application Building a Graphical Application Documenting an Application - Running and Debugging Applications by Using Visual Studio 2010 - Using C# Programming Constructs Declaring Variables and Assigning Values Using Expressions and Operators - Creating and Using Arrays Using Decision Statements Using Iteration Statements - Declaring and Calling Methods Defining and Invoking Methods Specifying Optional Parameters and Output Parameters Handling Exceptions Handling Exceptions Raising Exceptions Reading and Writing Files Accessing the File System Reading and Writing Files by Using Stream Creating New Types Creating and Using Enumerations Creating and Using Classes Creating and Using Structs Comparing References to Values Encapsulating Data and Methods Controlling Visibility of Type Members Sharing Methods and Data

Inheriting From Classes and Implementing Interfaces

Using Inheritance to Define New Reference Types
Defining and Implementing Interfaces
Defining Abstract Classes

Managing the Lifetime of Objects and Controlling Resources

Introduction to Garbage Collection
Managing Resources

Encapsulating Data and Defining Overloaded Operators

Creating and Using Properties
Creating and Using Indexers
Overloading Operators

Decoupling Methods and Handling Events

Declaring and Using Delegates
Using Lambda Expressions
Handling Events

Using Collections and Building Generic Types

Using Collections
Creating and Using Generic Types
Defining Generic Interfaces and Understanding Variance
Using Generic Methods and Delegates

Building and Enumerating Custom Collection Classes

Implementing a Custom Collection Class
Adding an Enumerator to a Custom Collection Class

Using LINQ to Query Data
Using the LINQ Extension Methods and Query Operators
Building Dynamic LINQ Queries and Expressions

Integrating Visual C# Code with Dynamic Languages and COM Components
Integrating C# Code with Ruby and Python

Beitrag

Accessing COM Components from C#

Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).