

C#-programmering - 4 dagar

kurser 419

- Du får lära dig att**
- Skapa, kompilera och köra objektorienterade C#-program med hjälp av Visual Studio
 - Skriva och förstå C#-språkets konstruktioner, syntax och semantik
 - Producera program för skrivbords- och webbapplikationer med hjälp av stabila flerlayersarkitekturer
 - Utveckla återanvändbara .NET-komponenter med hjälp av gränssnittsrealisering och standard-designmönster
 - Använda namnutrymmen och klasser i .NET Framework
 - Komma åt databaser med Language Integrated Query (LINQ) och Entity Framework (EF)
- Sammanfattning** C# är ett modernt objektorienterat programmeringsspråk vars syfte är att skapa enkla men kraftfulla program. Det har utformats för att utnyttja CLI-funktioner. C# är kärnspråket inom Microsofts .NET-ramverk. På den här kursen får du lära dig det som behövs för att använda språket C# samt .NET-ramverket för att utveckla program för många typer av skrivbords- och webbapplikationer.
- Vem bör delta** Alla som är intresserade av programmering med C#. Erfarenhet av ett modernt programmeringsspråk som t ex VB, Java, Pascal eller C/C++ förutsätts. De som endast har erfarenhet av COBOL, RPG, SQL, HTML eller liknande bör överväga att gå kurs 502, "Programmering med .NET: introduktion".
- Praktiska övningar** Du får praktisk erfarenhet av att skapa din egen C#-applikation. Bland övningarna märks:
- Skriva och kompilera C#-program med hjälp av Visual Studio
 - Bygga C#-klasser och arvshierarkier
 - Skriva skrivbords- och webbapplikationer med Windows Forms och Web Forms
 - Konstruera och driftsätta specialkomponenter i .NET
 - Implementera datafrågelogik för databaser som använder LINQ och EF
 - Påskynda utvecklingen med .NET Framework-biblioteket

C#-programmering - 4 dagar

kurser 419

Introduktion till C#-språket

Utvecklingen av C#

- Jämförelse mellan olika versioner av C#
- Uttrycka C#-modeller i UML

C# och .NET-infrastrukturen

- Common Language Infrastructure (CLI)
- Filosofi för hanterad kod
- Common Intermediate Language (CIL) och metadata

Grundläggande om språk

Datatyper och styrkonstruktioner

- Deklarera implicita och explicita variabler
- Värde och referenstyper
- Unicode-tecken och strängar

Definiera och anropa metoder

- Specifikation av "Main"-metod
- Överföra argument och returvärdet
- Variablers omfattning och livslängd
- Namnmetoder och symboliska metoder
- Hantera undantag
- Återställa resurser

Använda .NET-biblioteksklasser

- Undvika kollisioner genom att använda "namespaces"
- Utföra I/O med "stream"-klassen och serialisering
- Standardsamlingar och generiska samlingar

Utveckla C#-klasser

Definiera klasser

- Kapsla in attribut med metoder och egenskaper
- Använda konsekvent initiering med konstruktörer
- Överlagrade metoder och konstruktörer
- Återanvändning genom arv och polymorfism

Skapa och använda objekt

- Tilldela objektsminne med `new`
- Överföra initialvärden till konstruktörer
- Välja värde- eller referenstilldelning
- Rutor eller inte
- Starta metoder och komma åt egenskaper

Sammankoppla objekt

Sammankoppla klasser

- Manipulera referenser
- Fysisk kontra logisk motsvarighet
- Välja samlingsbiblioteksklasser
- Öka driftsäkerheten med generiska funktioner

Visa gränssnitt

- Definiera en gränssnittsspecifikation
- Implementera ett gränssnitt i en klass
- Polymorfism för gränssnitt
- Händelser och delegater

Förenkla utveckling av komponenter

Komponentfunktioner i .NET

- "Manifests" och "assemblies"
- Driftsätta komponenter och applikationer
- .NET assembly metamodell

Skriva .NET-komponenter i C#

- Skapa och anropa klientkomponenter
- Utöka `System.ComponentModel.Component`

Binda ihop befintliga komponenter

- Åtkomst av COM/DCOM
- Verktyg för framåt- och bakåt-kompatibilitet
- Anropa befintliga komponenter

Integrera C# med andra språk

- Samordna komponenter via CLI
- Åtkomst till metadata
- Hantera skillnader mellan olika språk

Implementera och förbättra

C#-lösningar

Bygga program med flera nivåer

- Använda stabila arkitekturer (MVC och EDM)
- Ersätta användargränssnittet
- Koda branschstandardiserade designmönster i C#
- Distribuera en C#-applikation

Arbeta med relationsdatabaser

- Åtkomst till databaser med Entity Framework (EF)
- Integrera utökade funktioner hos C# med LINQ

Standarder och versioner

- Standardisering med ECMA/ISO
- Funktioner i olika C#-standarder

Avancerade tekniker

- Automatisera dokumentation med XML
- Implementera gränssnittet `IEnumerable<T>`
- Anropa utökningsmetoder
- Använda händelser, delegater och lambda-uttryck
- Specificera utvecklingsattribut