

Læseplan for Grundlæggende C#

Indhold

Kurset er et indledende kursus i programmeringssproget C#, der i øjeblikket findes i version 7 som den nyeste udgave. Vi introducerer samtlige elementer af sproget helt fra bunden, så der kræves intet forhåndskendskab til C#.

Kurset baseres på Troelsens og Japikses bog [1]. Vi vil initielt kort skitsere ideerne i Part I, men vil fokusere næsten udelukkende på Part II, III og IV af bogen, da disse dele beskriver C#-sproget. Til sidst vil vi dog berøre en lille del af Part V. Nu og da vil bogen dog gå ned i et detaljeniveau (eksempelvis med analyse af IL-kode), som ikke er i tråd med et grundlæggende kursus. Vi vil derfor ofte præsentere stoffet en smule anderledes end bogen gør det – forhåbentligt med et større udbytte for kursisterne til følge. Part V og de senere dele beskriver de forskellige elementer af .NET-frameworket, hvilket er udenfor rammerne af dette kursus, bl.a. ASP.NET.

Vi vil i dette kursus (med få undtagelser) koncentrere os om at introducere C#-sproget ”så rent som muligt” gennem konsol-applikationer. Vi beskæftiger os derfor ikke meget med de forskellige typer af brugerfladekomponenter, som eksempelvis WinForms, WPF eller ASP.NET. Men vi introducerer løbende features, tips og tricks til at benytte værktøjet Visual Studio 2017, når chancerne byder sig.

Tidsplan

Kursus strækker sig over 6 dage efter følgende forventede tidsplan.

Dag 1 + 2 (Introducerende)	Indledning Module 1: “What is .NET and C#?” Module 2: “Hello, World” Module 3: “Value Types and Expressions” Module 4: “Reference Types and Statements” Module 5: “Methods”
Dag 3 + 4 (Mellemniveau)	Genopfriskning af Dag 1 + 2 Module 6: “Introducing Object-Oriented Programming” Module 7: “Properties and Static Members” Module 8: “Inheritance and Polymorphism” Module 9: “Structured Exception Handling” Module 10: “Interfaces”
Dag 5 + 6 (Avanceret)	Genopfriskning af Dag 3 + 4 Module 11: “Collections and Generics” Module 12: “Delegates, Events, and Lambda Expressions” Module 13: “Advanced C# Language Features” Module 14: “LINQ to Objects” Opsamling Afslutning

Kurset indeholder både instruktorgennemgang af stof samt praktiske øvelser, der relaterer sig til stoffet. Fordelingen mellem gennemgang og øvelser er ca. 65:35, dog med variationer fra modul til modul.

Det vil være en **meget stor** fordel at have skimmet stoffet inden instruktorgennemgangen, da det så bliver lettere at få et udbytte af gennemgangen. Endvidere vil det være **optimalt**, hvis perioderne mellem kursusdagene benyttes til at få en smule hands-on øvelse med C# gennem det stof, som er blevet gennemgået. Den bedste måde at lære et sprog på er at "få det ind gennem fingrene". 😊

Beskrivelse af de enkelte moduler følger nedenfor.

Module 1: "What is .NET and C#?"

Synopsis

.NET-frameworket introduceres i overblik og der gives en kort grundlæggende introduktion til dets hovedbestanddele og karakteristika. Som en del af dette forklares, hvordan programmeringssproget C# passer ind i det større .NET-billede.

Læsestof

- Chapter 1, s. 3 – 33 udgør baggrundsmaterialet for dette modul.

Module 2: "Hello, World"

Synopsis

Et simpelt "Hello, World"-program illustreres i C#, og de enkelte dele af dette diskuteres kort. Dette giver anledning til at berøre Main-metoden, namespaces, input og output, command-line argumenter samt en introduktion til at lave og køre C#-programmer i Visual Studio 2017.

Læsestof

- Elementer fra Chapter 1, s. 3 – 33 indgår også kort i dette modul.
- Chapter 2, s. 35 – 52 udgør baggrundsmateriale for udvikling i Visual Studio 2015, så disse sider skal kun skimmes af *virkeligt* interesserede.
- Chapter 3, s. 53 – 67 udgør stoffet i dette modul.

Module 3: "Value Types and Expressions"

Synopsis

Grundlæggende data-typer introduceres, og opdelingen mellem værdi- og reference-typer beskrives. Vi lægger hovedvægten på værdi-typer og giver et overblik over de indbyggede værdi-typer. Det forklares hvordan variabler erklæres og initialiseres, samt hvordan konstanter og variabler kan kombineres i udtryk vha. operatorer. Dernæst ser vi på hvordan man konverterer mellem data-typer og slutter med at vise, hvordan man kan definere sine egne værdi-typer til at komplementere de indbyggede.

Læsestof

- Chapter 3, s. 68 – 76 og 88 – 98 udgør stoffet i dette modul. Der er dog stor vægt på variabler og operatører i modulet.
- Chapter 4, s. 135 – 141 udgør behandlingen af bruger-definerede enumerationer og structs.

Module 4: “Reference Types and Statements”

Synopsis

Vi dækker her reference-typerne, som er modstykkerne til værdi-typerne fra Module 3. Disse beskrives ved to konkrete slags reference-typer; arrays og strenge, der begge er meget almindeligt brugte typer. Med disse som eksempel gennemgår vi oprettelse og initialisering af reference-typer samt tilordning af variabler af disse typer. De gængse operationer på strenge vises, hvor det fremhæves at strenge er immutable. Efter kort at have diskuteret nullable typer, dækkes sætninger i C#. Startende med selection-sætninger som if og switch, dækkes herefter de fire grundlæggende loop-konstruktioner samt jump-sætninger.

Læsestof

- Chapter 3, s. 77 – 87 behandler strenge i detaljer.
- Chapter 3, s. 98 – 109 omhandler iterations- og selektions-sætninger.
- Chapter 4, s. 111 – 117 introducerer arrays.
- Chapter 4, s. 142 – 150 sammenligner værdi- og reference-typer.
- Chapter 4, s. 150 – 153 forklarer nullable typer.

Module 5: “Methods”

Synopsis

Det vises grundlæggende, hvordan man definerer og kalder metoder i C# med parametre og returværdier. Dernæst diskuteres de fire parameter modifiers, samt hvilken effekt disse har på parameteroverførslerne, og vi kigger på named og optional parametre. Herefter kigger vi kort på, hvordan reference-typer opfører sig mht. metoder, inden vi til sidst beskæftiger os med overloading af metoder.

Læsestof

- Chapter 4, s. 118 – 132 udgør det grundlæggende stof omkring metoder.
- Chapter 4, s. 141 – 143 diskuterer blandt andet, hvor værdi- og reference-typer hhv. opfører sig som parametre til metoder.

Module 6: “Introducing Object-Oriented Programming”

Synopsis

Bogen starter med direkte at introducere syntaksen for klasser i C#. Vi indleder dog med først at give en introduktion til hvad objekt-orienteret programmering (OOP) er, samt hvad der udgør nøgleegenskaberne.

Vi beskriver OOPs første søjle, indkapsling, gennem et klassisk eksempel med bankkonti. Dernæst følger vi bogen og introducerer syntaks og semantik for, hvordan klasser og objekter i praksis benyttes i C#.

Læsestof

- Chapter 5, s. 159 – 174 udgør en introduktion til klasser i C#.
- Chapter 5, s. 188 – 190 beskriver access modifiers.
- Chapter 5, s. 211 – 212 omhandler partielle klasser.

Module 7: “Properties and Static Members”

Synopsis

Vi fuldender bankkonti-eksemplet fra Module 6 ved at indføre statiske members i klasser. Vi fortsætter herefter behandlingen af indkapsling i C# ved at definere properties som mekanismer til at opnå indkapsling i klasser. Vi berører i denne forbindelse object initializers.

Læsestof

- Chapter 5, s. 176 – 188 og 190 – 211 fortsætter behandling af objekt-orientering i C#.

Module 8: “Inheritance and Polymorphism”

Synopsis

Vi fortsætter behandlingen af objekt-orienteret programmering og dets syntaks i C# fortsættes her med gennemgang af OOPs to andre søjler. Efter at have diskuteret filosofien bag nedarvning, gennemgås konstruktionerne i C# gennem en lang række eksempler. Dernæst beskæftiger vi os med polymorfi, og det fremhæves hvordan virtuelle metoder skabes og overskrives i nedarvede klasser. Vi berører dernæst en ny feature, som er pattern matching på objekt-typer. Vi slutter modulet med at gennemgå de vigtigste egenskaber af System.Object, som alle objekter i C# direkte eller indirekte arver fra, samt hvordan der kan konverteres mellem værdi- og reference-typer.

Læsestof

- Chapter 6, s. 213 – 254 udgør beskrivelsen af nedarvning og polymorfi i C#.
- Chapter 3, s. 106 – 109 introducerer pattern matching på typer.
- Chapter 9, s. 330 – 333 omhandler boxing og unboxing.

Module 9: “Structured Exception Handling”

Synopsis

Vi starter med at introducere exceptions som den objekt-orienterede måde at håndtere fejl på, idet vi sammenligner med manglerne ved den ”traditionelle” fejlhåndtering. Det skitseres hvordan exceptions flyder gennem en programudførelse, og vi gennemgår de centrale syntaktiske konstruktioner med try-catch-finally til at håndtere exceptions på. Undervejs ser vi på, hvordan Visual Studio 2015 kan være en stor

hjælp i fejlsituationer. Hvis der er tid til sidst vises hvordan man kan definere egne exception-klasser og et par gode retningslinier for generel exception-håndtering gennemgås.

Læsestof

- Chapter 7, s. 255 – 281 udgør det samlede materiale.

Module 10: “Interfaces”

Synopsis

Interfaces introduceres som en konstruktion til at definere kontrakter eller ”opførsler”, som en klasse eller struktur kan udvide eller implementere. Det vises, hvordan man implementerer eksisterende interfaces samt konstruerer egne interface-definitioner, som kan implementeres. Interfaces er reference-typer, og vi undersøger derefter, hvordan interfaces forholder sig mht. forskellige aspekter som f.eks. konverteringer og metoder. Dernæst berøres nogle af de vigtigste interfaces indbygget i .NET, og der gives konkrete eksempler på, hvordan disse benyttes i praksis. Vi vil til sidst se på et interface til oprydning af objekter og kort tale om garbage collection i .NET.

Læsestof

- Chapter 8, s. 283 – 321 udgør det samlede materiale, hvor vi dog ikke vil bruge tid på s. 311 – 315.
- Chapter 13, s. 479 – 500 beskriver objekters levetid i detaljer. Den resterende del af kapitlet ser vi bort fra.

Module 11: “Collections and Generics”

Synopsis

Der indledes med et meget overfladisk overblik over collection-relaterede interfaces, hvorefter de ”klassiske” collection-typer nævnes. Vi vil med baggrund i disse motivere, hvorfor der er brug for de type-sikre, fleksible og effektive generics. Der forklares, hvad generics er, og de vigtigste indbyggede generiske collection-typer introduceres. Til sidst ser vi på, hvordan man selv kan definere generiske typer, hvilket er et uundværligt værktøj for C#-udviklere.

Læsestof

- Chapter 9, s. 323 – 363 udgør materialet til dette modul. Vi har dog allerede behandlet stoffet vedrørende boxing og unboxing, s. 330 – 333 under Module 7.

Module 12: “Delegates, Events, and Lambda Expressions”

Synopsis

Indledningsvis undersøges delegates, der i sin simpleste form blot er en reference til en metode. Vi ser hvordan delegates virker, samt hvordan man selv kan implementere sine egne delegate-typer. Herefter beskæftiger vi os med events som en indkapsling af delegates og ser eksempler på brugen af events til løsning af almindelige programmeringsopgaver. Dette leder over til at kigge på syntaksen for anonyme

metoder og senere de såkaldte lambda-udtryk, der basalt set blot er en præcis notation for anonyme metoder.

Læsestof

- Chapter 10, s. 365 – 406 udgør hoveddelen af stoffet. Vi vil dog slet ikke gå i helt den samme dybde som i bogen.

Module 13: “Advanced C# Language Features”

Synopsis

Vi gennemgår kun udvalgte af de avancerede C# features. Vi definerer indeksere, der er en særlig type properties, som kan benyttes til at give klasser ”array-lignende” egenskaber, og ser eksempler på at konstruere disse. Herefter kigger vi på extension-metoder, der udvider eksisterende typer med nye metode-egenskaber. Endeligt gennemgås anonyme typer og de nye værdi-tupler. De resterende avancerede features berøres ikke. Vi slutter med at berøre lokale metoder.

Læsestof

- Chapter 11, s. 407 – 412 omkring indexers.
- Chapter 11, s. 425 – 430 omkring extension-metoder.
- Chapter 11, s. 431 – 436 omkring anonyme typer.
- Chapter 4, s. 154 – 158 beskæftiger sig med tupler.
- Chapter 4, s. 133 – 134 beskriver lokale metoder.

Module 14: “LINQ to Objects”

Synopsis

Vi kan nu introducere LINQ, der er en kraftfuld mekanisme til at forespørge og søge på data i forskellig form. Vi koncentrerer os her om den del af LINQ, der er ren C# – dvs. uden eksempelvis komponenterne, der tilgår data fra XML og SQL, hvilket vi dog skitserer til sidst i modulet. Vi præsenterer de praktiske forudsætninger man skal igennem for at kunne benytte LINQ, og gennemgår herefter LINQs query-udtryk og query-operatorer. Vi ser disse benyttet på collections og arrays, og ser ved flere eksempler til sidst, hvordan LINQ kan benyttes mod andre datakilder end objekter i hukommelsen.

Læsestof

- Chapter 12, s. 445 – 478 udgør alt stof. Vi vil dog kun i overblik bruge tid på s. 473 – 477.
- Chapter 22, s. 857 – 906 er en grundig gennemgang af Entity Framework, der indeholder mulighed for LINQ mod databaser såsom SQL Server, hvilket vi kun meget kort skitserer.

Referencer

- [1] Andrew Troelsen og Philip Japikse: *“Pro C# 7”* (Eighth Edition).
Apress (2017)