

**Kursus****1 dag**

Nr. 90322 A

DKK 5.499

ekskl. moms

Dato

06-06-2024

05-09-2024

30-09-2024

07-11-2024

05-12-2024

Sted

Taastrup

Aarhus

Taastrup

Aarhus

Taastrup

SOLID Programming in C#

På kurset får du en introduktion til både teori og praksis bag SOLID-principperne realiseret i moderne C# samt de tilhørende patterns. Vi gennemgår de fem principper teoretisk og viser i praksis, hvordan eksisterende C#-kode kan refaktoreres til disse vha. passende design patterns.

Lær at bruge SOLID til at skrive bedre kode

SOLID er en fællesbetegnelse for fem design-principper til at skrive bedre og mere vedligeholdbar kode i objekt-orienterede programmeringssprog. SOLID understøttes af en række design patterns som gode, velafprøvede garantier for, at principperne i praksis overholdes. Her er bl.a. Abstract Factory, Strategy og Decorator vigtige værktøjer.

Derudover benyttes ofte en Dependency Injection container (som for eksempel Unity eller Autofac) som et framework til at skabe og indskyde klassers afhængigheder automatisk. På denne måde håndteres afhængighederne i praktiske applikationer på en let og overskuelig måde.

I anden halvdel af kurset vil vi også fokusere på de værktøjer og frameworks, der yderligere kan hjælpe os med at skrive SOLID kode. Vi vil især bruge Dependency Injection containers til automatisk at håndtere klassers afhængigheder i praksis, så vi vil få hands-on erfaring med værktøjer som f.eks. Unity eller Autofac. Vi diskuterer gode og dårlige sider af disse containers gennem praktiske eksempler.

Deltagerprofil

Kurset er for C#-udviklere, der gerne vil skrive struktureret, testbar, mere fleksibel og vedligeholdbar kode, og som har lyst til i praksis at programmere de beskrevne teknikker i den nyeste version af C#.

Forudsætninger

For at deltage på kurset skal du have viden om C# og Visual Studio på niveau med [Grundlæggende C# kurset](#).

Udbytte

- Bliv introduceret til teorien bag SOLID-principperne
- Lær at bruge SOLID til at skrive bedre kode, som er lettere at vedligeholde i objektorienterede programmeringssprog
- Lær, hvordan eksisterende C#-kode kan refaktoreres til de fem principper
- Få hands-on erfaring med værktøjer som f.eks. Unity eller Autofac

Det får du på arrangementet

- Øvelser og inddragelse
- Kursusbevis
- Erfaren underviser
- Maks. 10 deltagere
- Fuld forplejning
- Materiale
- Gratis parkering
- Materiale på dansk
- Undervisning på dansk

Indhold

Introduction

What is SOLID and why should we care?

SOLID Principles

- Single Responsibility Principle
- Open/Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

Design Patterns for SOLID in Practice

- Abstract Factory
- Strategy
- Decorator
- Null Object
- Repository
- Singleton
- Facade

Dependency Injection

- Containers
- Patterns and Anti-patterns

Anmeldelser af SOLID Programming in C#

En levende underviser der i den grad havde styr på sit stof

— **Kim Gabriell Nørgaard** Rigspolitiet, Koncern IT

Dygtig og erfaren underviser, med en undervisningsform som aldrig blev trættende
— **Nicolai Elbæk Lindof Lund** Rigspolitiet, Koncern IT

Form

Kurset er struktureret som en workshop, der veksler mellem gennemgang og konkrete programmeringsopgaver, hvor du gennem kurset implementerer en større, sammenhængende løsning vha. SOLID og Dependency Injection. Efter hver gennemgang implementerer du de beskrevne principper i den sammenhængende løsning. Der er stor fokus på, at du tilegner dig praktisk erfaring, der efter endt kursus umiddelbart kan anvendes i konkrete projekter. Tiden på kursusdagene er ligeligt fordelt mellem gennemgang og praktisk SOLID programmering i C#.

Materiale

Kursusmaterialet består af slides og workshop-opgaver med tilhørende løsninger, specialudviklet til kurset.

SOLID-principperne

SOLID-principperne blev oprindeligt formuleret af Robert C. Martin (også kendt som "Uncle Bob") og omfatter bl.a. Single Responsibility Principle og Open/Closed Principle, der er velkendte for deres vigtighed for godt objekt-orienteret håndværk. Sammen med de tre yderligere principper, Liskov Substitution Principle, Interface Segregation Principle og Dependency Inversion Principle, udgør principperne til sammen ikke blot de 5 forbogstaver i SOLID, men også en sammenhængende programmeringsfilosofi med mange overvundne udviklere, der nærmest religiøst har taget SOLID til sig.



UNDERVISER

Jesper Gulmann Henriksen

Jesper har siden 2002 arbejdet med C# og .NET som software-udvikler i meget forskellige områder som overvågningssystemer, software til flyveledere samt betalingsløsninger på mobilen. Derudover har han gennem en årrække været tilknyttet Teknologisk Institut som underviser på C#-kurserne. Han har gennem årene fulgt Microsofts teknologier fra tæt hold.

Har du faglige spørgsmål så kontakt



Mette Rosenløv Vad
+45 72202432
mva@teknologisk.dk