

▮ Øvelse 6 – Grundlæggende OOP med C#

▮ Information

Formålet med denne øvelse er at opnå en grundlæggende forståelse af C# og objektorienteret programmering heri.

Øvelsen tager udgangspunkt i Microsofts officielle tutorial og lægger vægt på, at du anvender dokumentationen aktivt og reflekterer undervejs.

▮ [Tutorial: Introduktion til klasser i C# \(Microsoft\)](#)

▮ *Læs og implementér først – læs derefter refleksionspunkterne for hvert afsnit.*

▮ Praktiske læringsmål

- ▮ Oprette og anvende en klasse i C#
 - ▮ Bruge konstruktører til at initialisere objekter
 - ▮ Forstå begrebet indkapsling med `private` accessors
-

▮ Instruktioner

1. ▮ Gennemfør tutorialen ved at implementere en konsolapplikation
 2. ▮ Brug læseguiden nedenfor til at reflektere over nøglebegreber i hvert afsnit
-

▮ Læsevejledning og refleksionspunkter

▮ *Prerequisites*

Du introduceres til de værktøjer, du skal bruge: - [Visual Studio](#) - [Visual Studio Code](#)

▮ SDK = *Software Development Kit* → inkluderer compiler (Roslyn), CLI-værktøjer m.m.

▮ *Create your application*

Du opretter et projekt med `dotnet new console` eller via GUI i Visual Studio.

▮ **Projektet skal ligge i en mappe med navnet `classes`.**

Typisk lægger jeg alle mine projekter i en rodfolder på det drev, jeg arbejder på. I dette tilfælde kunne det f.eks. være `C:\source\classes` eller `/source/classes`.

▮ *Define the bank account type*

Implementér klassen `BankAccount` i filen `BankAccount.cs`.

Overvej:

- Hvad er `Number`, `Owner` og `Balance`?
 - Hvad gør metoderne `MakeDeposit` og `MakeWithdrawal`?
 - Hvorfor omtales klasser også som *typer*?
-

▮ *Open a new account*

Implementér en konstruktør og en privat statisk variabel.

Overvej:

- Hvorfor bruge en konstruktør med parametre?
 - Kan objektet oprettes uden parametre?
 - Hvad betyder det, at `s_accountNumberSeed` er privat?
-

▮ *Create deposits and withdrawals*

Tilføj en ny klasse `Transaction` og brug den til at håndtere transaktioner.

Overvej:

- Hvad er fordelene ved at transaktioner har deres egen klasse?
 - Hvorfor er listen `allTransactions` privat?
 - Hvad sker der, hvis der kastes en `ArgumentOutOfRangeException`?
 - Hvorfor kaldes `MakeDeposit` fra konstruktøren?
-

▮ *Challenge – Log all transactions*

Tilføj en metode, der returnerer hele transaktionshistorikken som tekst.

Overvej:

- Hvordan anvendes `allTransactions`?
 - Hvorfor er det en god idé at holde transaktionerne private?
-

□ Selvevaluering – Centrale begreber fra øvelsen

Når du har gennemført øvelsen, brug denne korte selvevaluering til at reflektere over de vigtigste tekniske begreber:

1. Hvad er et **Objekt**, og hvilken rolle spiller den i applikationen?
 2. Hvordan kan **klasser** bruges til i praksis?
 3. Hvad er anvendes en **Metode** til?
 4. Hvad er en metodes argumenter?
-

□ Ekstra links

- [Microsoft Docs – Classes and Objects](#)
 - [Konsolapplikation i Visual Studio](#)
-

Last update: 2025-10-30 12:07:19