

Solutions og projects i .net

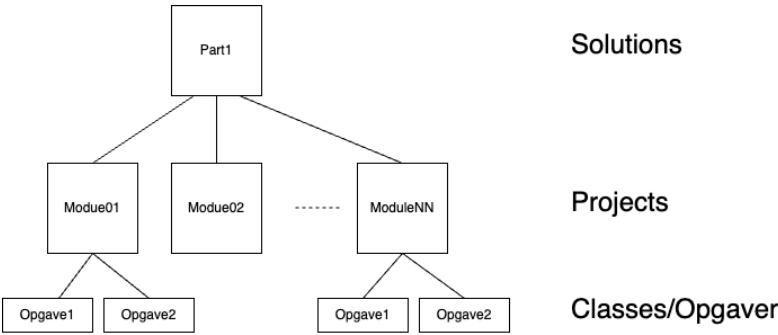
Denne note vil kort skitsere hvordan man kan organisere sin kode her i den første del af faget. Her skal vi lave såkaldte konsol-applikationer, dvs. programmer hvor input/output sker gennem terminalen/kommando-prompten. Yderst primitivt og kun brugt til programmer som bare startes og kun har en sparsom interaktion med brugeren. Fx er mange af git kommandoerne lavet som konsol programmer.

I .net organiserer man sin kode i solutions - dette skal I tænke på som en paraply for en række programmer, som tilsammen udgør et system. En solution består af et antal projekter. Et projekt svarer til et program eller et bibliotek. Et program er noget man kan køre, mens et bibliotek (library) er noget kode som et program eller et andet bibliotek kan anvende. d3 er et eksempel på et bibliotek.

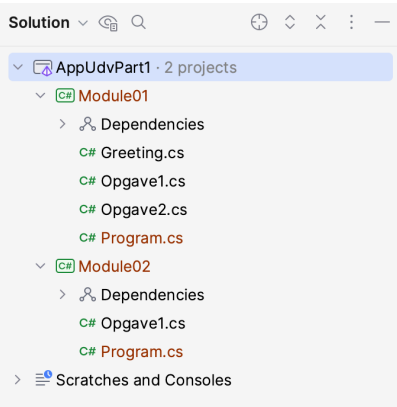
Opgave 1: I ovenstående blev brugt 4 begreber: solution, projekt, program og bibliotek. Konstruer en begrebsmæssig model der viser hvordan disse begreber hænger sammen.

Jeg anbefaler, at I laver en solution for den første del af faget, hvor vi skal lave konsol programmer. Vi kunne kalde den for Part1. Og under den vil jeg så have et projekt for hver modul (dag). En simple navngivning kunne være Modul01, Modul02, osv.

Under moduleerne vil jeg så oprette en klasse for hver opgave eller eksempel.



I Rider kommer det til at se således ud:



Som I kan se laves altid en fil med navn Program.cs når man opretter et projekt som konsol program (Console Application). I denne fil er programmets startpunkt; det er den kode der udføres når man kører programmet.

En nem måde at kunne kode og teste sine opgaver på er, at klassen for opgaven indeholder en offentlig (public) og statisk (static) metode med navn Run. Så kommer klassen til at se således ud:

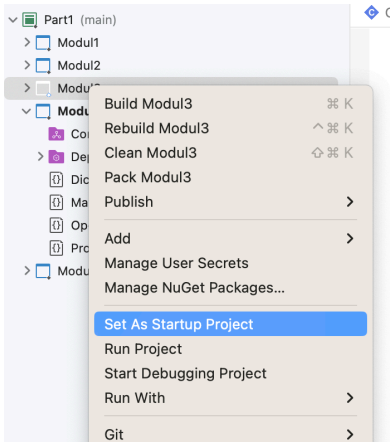
```
public class Opgave1
{
    public static void Run()
    {
        Console.WriteLine("It works");
    }
}
```

For at få opgaven udført, kan man skrive følgende i Program.cs (som er entry-point):

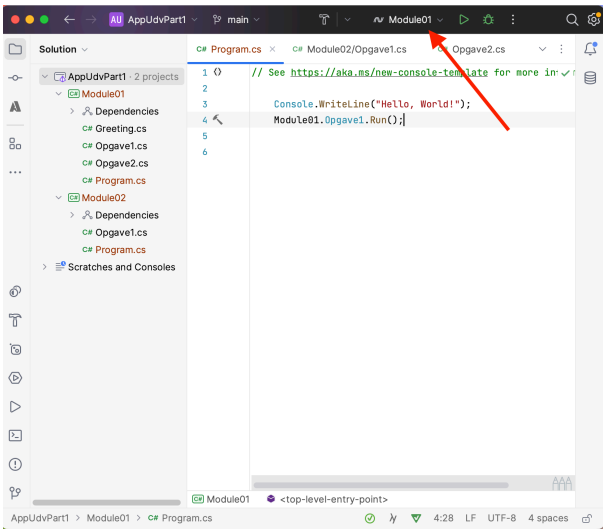
```
Opgave1.Run();
```

Som I kan, se er det blot Opgave1 der skal ændres til fx. Opgave2 når en anden opgave skal køres.

I skal dog huske på, at når der i en Solution ligger flere programmer (fordi der er et per modul/dag), skal man huske at vælge hvilket program der skal køres. Hvordan dette gøres, afhænger af hvilken IDE der brgues. I Visual Studio gøres ved at vælge det modulet man skal køre som Default Project. Se figures herunder:



I Rider vælger man hvilket projekt der skal køres øverst til højre, som vist herunder.



Denne måde at organisere kodebasen på benyttes i den første del af faget - og al den kode jeg har vist jer (og lidt til) ligger på Github [her](https://github.com/oleeriksen/AppUdvPart1) ➞ [. \(https://github.com/oleeriksen/AppUdvPart1\)](https://github.com/oleeriksen/AppUdvPart1).