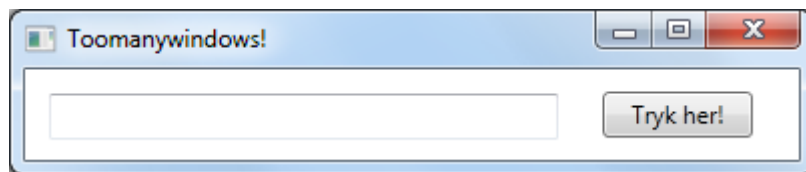


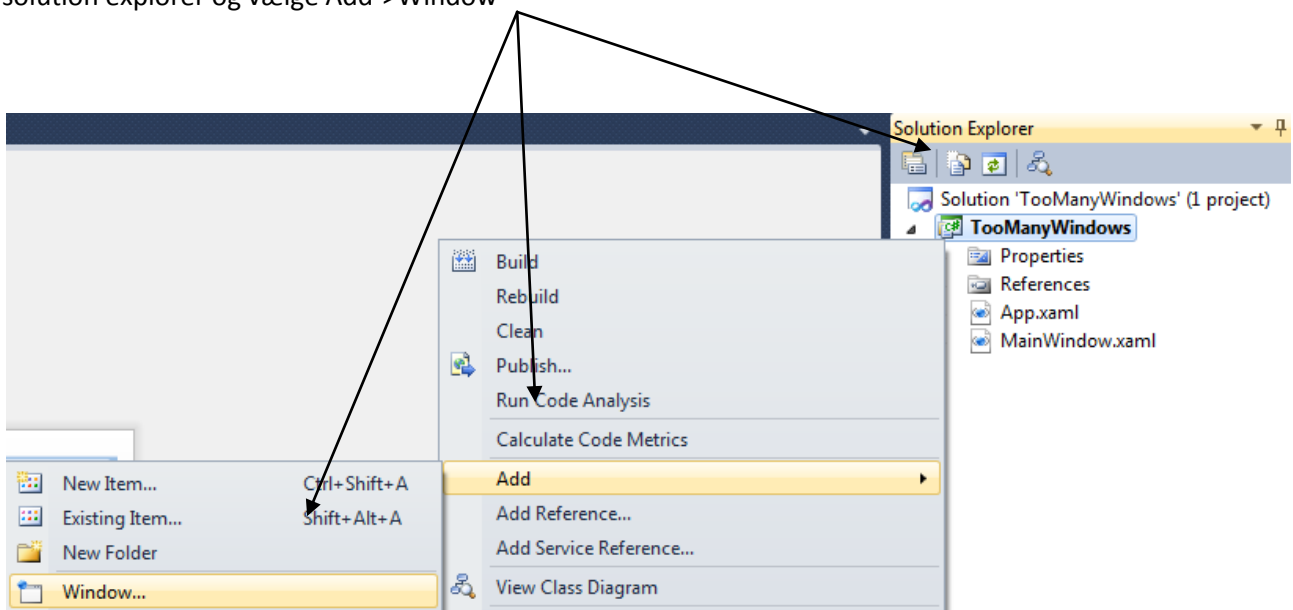
Indtil nu har vi arbejdet med et vindue af gangen. Dette er der ikke noget galt i. Men i nogle tilfælde kan det være ønskeligt at have flere vinduer, som man kan sprede informationer ud på. Og vi vil i det efterfølgende gennemgå hvorledes dette kan gøres i WPF.

Start med at oprette et nyt WPF projekt i VS.

På det indledende vindue skal du tilføje en tekstboks og en button som vist herunder:



Herefter skal du tilføje endnu et vindue til dit projekt. Dette gøres ved at højreklikke på projektet ude i solution explorer og vælge Add->Window

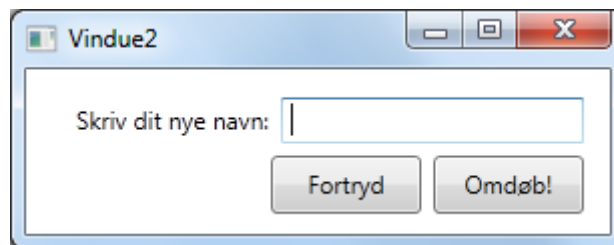
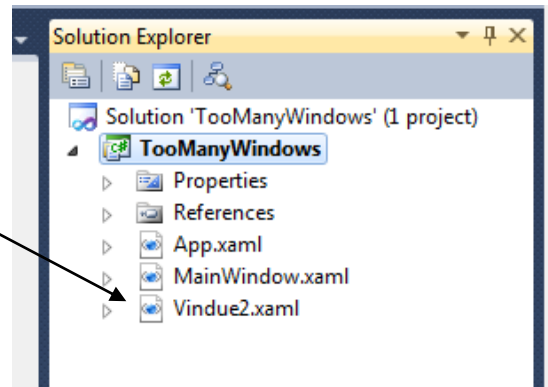


Herefter vil der komme et nyt "Add new item" vindue, hvor du kan skrive navnet på dit nye vindue (Jeg har kaldt mit for vindue2.xaml og jeg foreslår at du gør det samme). Husk at du ikke skal ændre valget der allerede er lavet for dig. Altså at du vil oprette endnu et WPF Window.

Når du er færdig og har trykket OK. vil dit nye vindue være tilføjet dit projekt.

Og vi skal nu prøve om vi kan åbne det ny vindue ved at trykke på knappen på vindue1.

Vi har nu gang i to vinduer og det er jo fint nok, men nogle gange ville det være rart at kunne overføre data imellem to vinduer, og dette kan umiddelbart gøres på to måder som vi vil kigge på i det efterfølgende. Men inden vi går i gang, skal vi lige have tilføjet en tekstboks og to knapper til vindue2, i stil med det herunder:



Nu er vi klar til at kode lidt. Først skal vi have fundet ud af hvordan vi kan åben vindue 2 fra vindue 1... Dette er ren faktisk temmelig lige til, for vores vindue2 er som du sikkert husker, en ganske almindelig class nedarvet fra Window. Så alt vi skal gøre er at oprette classen og bede den om at vise sig...

Start med at tilføje eventhandler for Click eventet på knappen i vindue 1. Herefter tilføj du koden herunder til eventhandleren:

```
Vindue2 vindue = new Vindue2();
vindue.Show();
```

Kør programmet og se om ikke tingene virker? Skift nu den nederste linje ud med følgende :

```
vindue.ShowDialog();
```

Hvad er forskellen?

Så skal vi have overført noget data! Og igen kan vi med simpel grundlæggende OOP teknikker overføre enten via en non default constructor eller via parametre.

Vi starter med en non default constructor til vindue2:

```
// Overloaded constructor
public Vindue2(String navn)
{
    InitializeComponent();
    textBox1.Text = navn;
}
```

Husk at du skal tilskrive navn til tekstboksen EFTER InitializeComponent, ellers eksistere tekstboksen ikke!

Så kan vindue2 initieres fra vindue1 med det data der skal overføres, som vist herunder:

```
Vindue2 vindue = new Vindue2("Thomas");
vindue.ShowDialog()
```

Husk at dette er envejs kommunikation. God programmerings skik siger nemlig at kommunikation umiddelbart kun må gå fra den oprettende class til den oprettede class. Owner->Owned

Alternativt kan vi overføre data via properties. Dette kræver indledningsvis lidt mere arbejde, men det gør det nemt at sende data frem og tilbage imellem de to vinduer.

I vindue2 skal der tilføjes en property, dette gøres lettest ved at skrive "prop" lige over constructoren og trykke tab to gange:

```
public partial class Vindue2 : Window
{
    public string Navn { get; set; }

    public Vindue2()
    {
        InitializeComponent();
        //textBox1.Text = navn;    <-- Skal slettes igen!
    }
}
```

Herefter skal propertyen rettes lidt til, så den overfører navn til og fra tekstboksen.

```
public string Navn
{
    get
    {
        return textBox1.Text;
    }
    set
    {
        textBox1.Text = value;
    }
}
```

Nu kan vi anvende propertyen i vindue2 efter at vi har oprettet vindue2, i vindue1:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    Vindue2 vindue = new Vindue2();
    vindue.Navn = "Thomas";
    vindue.ShowDialog();
    textBox1.Text = vindue.Navn;
}
```

Nu virker tingene forholdsvis efter hensigten. Vi kan i hvert fald overføre data fra vindue1 til vindue2 og tilbage igen. Og du kan jo teste det ved at ændre navnet i tekstboksen på vindue2 og herefter lukke vinduet på krydset i hjørnet?

Nu skal vi have lavet en reel dialog ud af vindue2. Vi skal have tilføjet de to eventhandlers til knapperne på vindue2. Start med at tilføje eventhandleren til "fortryd" knappen og tilføj følgende kode:

```
DialogResult = false;
this.Close();
```

Herefter tilføjer du eventhandleren til "Omdøb" Knappen og tilføjer koden herunder:

```
DialogResult = true;
this.Close();
```

Så er arbejdet færdig i Vindue2

Tilbage i Vindue1 kan vi i click eventet kigger efter i vindue2's DialogResult om brugeren har trykket Omdøb eller fortryd:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    Vindue2 vindue = new Vindue2();
    vindue.Navn = "Thomas";
    vindue.ShowDialog();

    if (vindue.DialogResult == true)
    {
        textBox1.Text = "Hej med dig " + vindue.Navn;
    }
    else
    {
        textBox1.Text = "Du hedder stadig Thomas";
    }
}
```

Opgave:

Lav et nyt projekt med 2 vinduer, der kan overføre et objekt af typen "Person".

Person indeholder 3 properties:

string Fornavn

string Efternavn

int Alder

Disse tre variabler skal tilskrives i MainWindow og overføres til vindue2, hvor der kan rettes i dem. Og herefter skal de føres tilbage til vindue1 og vises i tre tekstbokse hvis brugeren vælger "Ret":

