# WPF Dialog GUI

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf1:** anvende centrale metoder til at specificere og konstruere algoritmer [...] * **1Pf2**: anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, designmønstre, abstrakte datatyper, datastrukturer, designmodeller anvende centrale faciliteter i programmeringssproget til realisering af algoritmer, designmønstre, abstrakte datatyper, datastrukturer, designmodeller og brugergrænseflader. |
| **Forventet læringsudbytte [SOLO]** | Programmeringssporet:   * [Unistrukturel] Du kan nævne enkelte kendetegn ved **applikationstypen** GUI applikation – WPF og XAML |
| **Din forberedelse** | Programmeringssporet:   * WPF UI-kontrollere:   + Evt. genlæs forberedelsen omkring basale UI-kontrollere fra ”Ex24-WPFSimpleGUI”-opgaven   + [Image](https://www.wpf-tutorial.com/basic-controls/the-image-control/) (vær opmærksom på brugen af BitmapImage- og Uri-klasserne)   + [TextBlock](https://www.wpf-tutorial.com/basic-controls/the-textblock-control/) * [Dialog boxes overview (WPF .NET)](https://docs.microsoft.com/en-us/dotnet/desktop/wpf/windows/dialog-boxes-overview?view=netdesktop-5.0)   + Med fokus på Message boxes, Common dialog boxes og Custom dialog boxes * [Creating a custom input dialog](https://www.wpf-tutorial.com/dialogs/creating-a-custom-input-dialog/) * [UIElement.IsEnabled Property](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement.isenabled?view=windowsdesktop-5.0&viewFallbackFrom=net-5.0) * [WPF IsEnabled Property (Button Example)](https://www.dotnetperls.com/isenabled-wpf) * [What is the difference between a relative and absolute path?](https://www.computerhope.com/issues/ch001708.htm)   + Med fokus på Windows |

I denne opgave skal du arbejde videre med Tusindfryd-casen og implementere en GUI til udvalgte dele af casen, hvor du skal lave en WPF-applikation med din egen dialogboks til at indhente data fra brugeren.

# Dagens ord:

A good interface makes it easy for users to tell the computer what they want to do, for the computer to request information from the users, and for the computer to present understandable information. Clear communication between the user and the computer is the working premise of good UI design.

# Øvelse 1: Terminologi

Del teamet op i to mindre grupper og brug **Ordet rundt** til at reflektere over begreberne ”Message boxes”, ”Common dialog boxes”, ”Custom dialog boxes”, ”IsEnabled property (for en WPF control)”, ”WPF TextBlock *control*” og ”WPF Image control”. Sørg for at alle får mulighed for at tale.

*Tidsramme: 20 minutter*

Overvej også, hvad forskellen er på en WPF TextBlock control og en WPF Label control, inden du går videre

A **message box** is a dialog box that pops up, e.g. save changes. Created from message box class. Message box title, text, icon and buttons are configurable.

**Common dialog boxes** are dialog boxes than are common among windows application so users are familiar with them, common dialog boxes are open file, save file and print.

**Custom dialog boxes** are like common dialog boxes, but domain specific and is created by the developer, e.g. add flower in Tusindfryd case. A window is used for this purpose.

**IsEnabled** is a Boolean control property that determines if the control is active or not(grey). It can be based on logic, e.g. a buttons IsEnabled property can be dependent on if a related textbox string is bigger than 0, i.e. the user has input a string in the box.

**WPF TextBlock control** is used for displaying text to the user, the text can be displayed either adding ellipsis when the text exceeds the window, by explicitly defining where linebreaks occur or by wrapping the text so the text is automatically fit inside the TextBlock control. It is not a control per se because the user cant interact with it and it doesn’t inherit from the control class, but is called a control nonetheless for simplicity.

**WPF** **image control** is a control used to display images in a window. The image can be displayed in different ways, either keeping the aspect and fitting in the control, which is the default(uniform). By changing the aspects and fitting the image completely in the control, Fill. By putting it in the control without changing anything about it except cropping if it doesn’t fit, None. Lastly scaled so it completely fits the control and aspect are not changed, uniform to fill. The property is Stretch=”x”.

**WPF Textblock control** and **WPF Label control** are both used to display text. TextBlock can only contain text whereas Label can contain text and images and can utilize access keys to the related control. TextBlock is used for displaying text. Label is used for a label to another control, e.g. explain that a TextBox is for first name.

**Benyt parprogrammering til alle følgende øvelser.**

# Øvelse 2: Tusindfryd

I opgaven Ex21-CustomExceptionHandling kiggede du nærmere på etablerNyBlomstersort-operationskontrakten i TusindFryd-casen. Denne operationskontrakt definerer forskellige detaljer for en tilsvarende system­operation i SSD’en. Kigger man nærmere på operationskontrakten, er der flere oplysninger, der er nødvendige for at kunne oprette en ny blomstersort:

* Navn (tekststreng med navn)
* Billede (binær data, men for nu en tekststreng, som angiver stien til billedet)
* Produktionstid (heltal)
* Halveringstid (heltal)
* Størrelse (heltal)

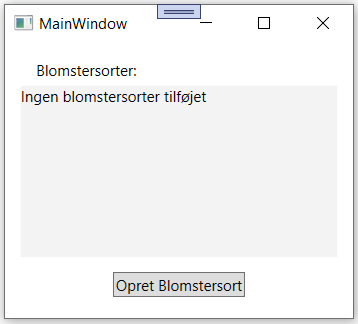
Disse oplysninger skal alle hentes fra brugeren.

I det følgende skal du udvikle en WPF-brugergrænseflade, som henter disse oplysninger via en dialogboks. Dvs. du skal både have:

1. Et hovedvindue (MainWindow) i din løsning
2. En dialogboks (et ekstra vindue) til at indhente oplysninger for en ny blomstersort
3. Kode til at åbne din dialogboks fra hovedvinduet
4. Kode til at gemme de indtastede blomstersort-oplysninger fra dialogboksen i en liste-datastruktur i hovedvinduet, f.eks. List<T> (lige nu behøver du ikke at overveje et repository)
5. UI-kontrol og kode til at vise tilføjelsen af en blomstersort i en passende UI-kontrol i hovedvindue, f.eks. en TextBlock, der viser alle blomstersorter i liste-datastrukturen

## Øvelse 2.1: Klargøring

Udfør følgende:

* Opret en ny WPF-applikation ”TusindfrydWPF” (husk at benytte projektskabelonen ”WPF Application”)
* Tilpas Height og Width under Window i XAML-filen for MainWindow
  + Til for eksempel henholdsvis 250 og 300
* Tilføj følgende UI-kontroller til MainWindow:
  + Indsæt øverst i hovedvinduet en Label kontrol med teksten ”Blomstersorter:”
  + Indsæt herunder en TextBlock. Prøv om du kan sætte baggrunden (Background) på TextBlock’en til lysegrå (LightGray), så den er synlig. Du vender tilbage til indholdet om lidt
  + Indsæt nederst i hovedvinduet en enkel knap (Button) med titlen ”Opret Blomstersort”.  
    Knappen vender du tilbage til om lidt  
    Det skulle gerne nogenlunde ende med billedet til højre ->
* Tilføj en C#-klasse til projektet med navnet ”FlowerSort” til at repræsentere en blomstersort  
  Genbrug evt. den tilsvarende klasse, du lavede i forrige uge, eller opret en helt ny til anledningen, f.eks. ud fra nedenstående software designklasse og ordliste:

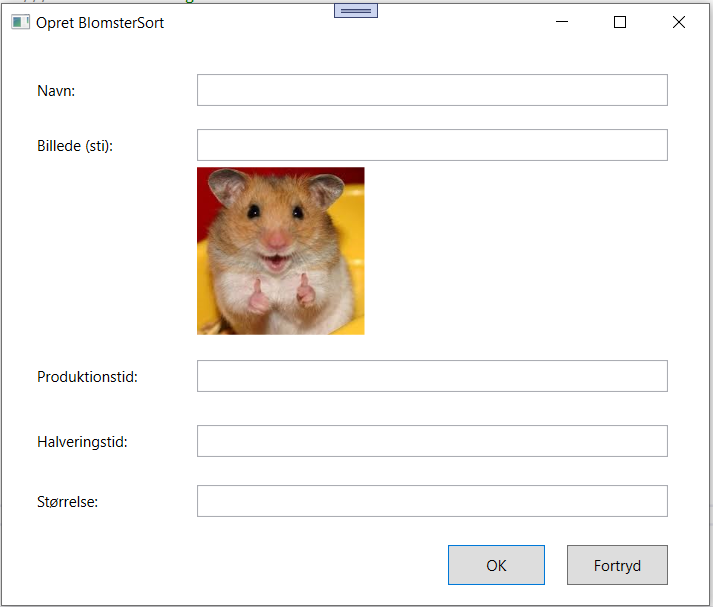
|  |  |  |
| --- | --- | --- |
| Designklasse | Dansk | Engelsk |
|  | Blomstersort | FlowerSort |
| Navn | Name |
| Billede (sti) | PicturePath |
| Produktionstid | ProductionTime |
| Halveringstid | HalfLifeTime |
| Størrelse | Size |

* Opret til sidst en liste i MainWindow codebehind-filen til at gemme alle indtastede blomstersorter, f.eks som angivet med variablen *flowerSorts* forneden:



## Øvelse 2.2: Opret dialogboks til blomstersort

Næste trin er at oprette selve dialogboksen til at indhente oplysninger for en ny blomstersort, hvor du skal lave et layout med Labels, Textbox’e og en enkelt Image UI-kontrol til at indtaste de nødvendige oplysninger samt klikke ’OK’ eller ’Fortryd’. Et eksempel på et GUI-design kunne være:

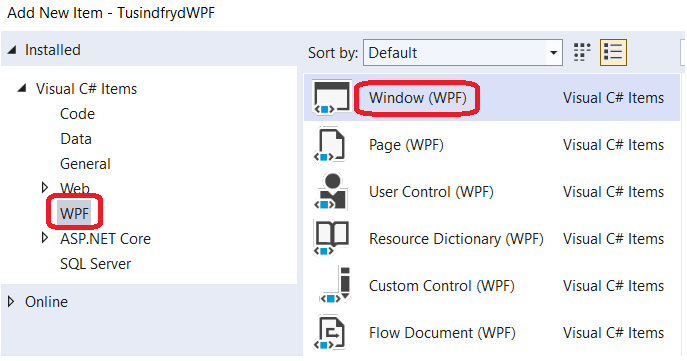


Funktionelle krav:

* Alle felter skal være udfyldt for, at en gyldig blomstersort accepteres ved klik på OK-knappen
* Image UI-kontrollen skal vise det billede, som stien peger på (foroven vises bare et eksempel). Så snart TextBox’en for billedstien mister fokus (LostFocus-eventet) skal det angivne billede vises, hvis muligt. Hvis der opstår en fejl, så håndtér den med brug af exceptions (try-catch-finally).  
  Selve billedet skal ikke gemmes i FlowerSort, kun stien til billedet

Udfør følgende:

* Opret nu et helt nyt WPF-vindue, som skal bruges som dialogboks til at indhente blomstersort-oplysningerne:
  + Højreklik på ”TusindfrydWPF”-projektet i Solution Explorer og vælg ”Add” og derefter ”New Item …”. Herved skulle du gerne se nedenstående dialogboks:



* + Vælg ”WPF” under ”Visual C# Items”, og derefter “Window (WPF)” som vist med rødt
  + Navngiv det nye vindue ”*CreateFlowerSortDialog.xaml*”, og tryk ”Add”-knappen
* Tilpas Height og Width under Window i XAML-filen for CreateFlowerSortDialog
  + Til for eksempel henholdsvis 400 og 400
* Tilføj alle de angivne UI-kontroller til dialogboksen
* Kod den nødvendige funktionalitet for dialogboksen, og afprøv  
  Vent til sidst med billedefunktionaliteten til alt det andet virker.  
  **Husk at debugge med breakpoints, hvis der opstår problemer.**

## Øvelse 2.3: Aktiver dialogboks fra hovedvindue og hent resultat (ny blomstersort)

Du har oprettet hovedvinduet og dialogboksen, men mangler stadig nogle få trin. Når brugeren klikker på ”Opret Blomstersort”-knappen i hovedvinduet, skal dialogboksen åbnes, så blomstersortoplysninger kan indtastes.

Udfør følgende:

* I Click-eventhandler for ”Opret Blomstersort”-knappen, opret en instans af dialogboksen, kald dens ShowDialog-metode, og håndter den returnerede booleske værdi. Find ud af, hvad returværdien betyder.
* Afprøv din kode, og tilret, hvis fejl. Husk debug og exception handling.

## Øvelse 2.4: Overfør resultat fra dialogboks (ny blomstersort)

Du kan nu åbne dialogboksen og indtaste oplysninger for en ny blomstersort, men vil selvfølgelig gerne kunne overføre disse oplysninger, så snart du trykker OK og returnerer til Click-eventhandler’en for ”Opret Blomstersort”-knappen.

Overvej:

* Hvordan får du fat i de 5 oplysninger fra dialogboksen?
* Implementér din løsning på problemet.  
    
  ***Hint:*** en public variabel i dialogboksens klasse kunne måske være vejen frem

## Øvelse 2.5: Opdatér liste af blomstersorter

Du har nu fået oplysninger fra dialogboksen tilbage i hovedvinduet, så det sidste, der mangler, er at opdatere listen af blomstersorter (flowerSorts) og genopfriske hovedvinduets TextBlock, så den viser en tekstlinje for hver blomstersort i listen.

* Implementér dette, og afprøv din applikation

## Øvelse 2.6: Styring af OK-knappen (Enable/Disable)

Du vil nu gerne lave en lidt mere smart styring af OK-knappen i dialogboksen, så knappen kun er aktiveret (Enabled), når alle felter er udfyldt, og ellers deaktiveret (Disabled).

Udfør følgende:

* Tilret koden i dialogboksen, så denne funktionalitet opnås.  
    
  ***Hint:*** der er en eventhandler, som kaldes hver gang teksten i en TextBox ændres.