

# Afleveringsopgave 1 - Navneliste

Link til GitHub:

<https://github.com/Mark5900/Vidergaaende-programmering-opgaver/tree/main/Uge%20%20-%20Afleveringsopgave%201>

## Redegørelse

Jeg har valgt at bruge WPF Application template til at lave projektet, da dette er en mere moderne måde at lave GUI på en Windows Forms. Men også for at udfordre mig selv, da jeg aldrig har brugt WPF før.

Da det er relativt simple kode har jeg valgt at lægge alt C# koden i MainWindow klassen.

Der er 3 variabler i starten af klassen:

- `bool bTest`: Bruges til at indsætte noget test data i `ListBox`, man kan sætte variable til `false`, hvis man ikke ønsker at der bliver indsat test data. Dog skal man lige være opmærksom på det kan give problemer at bruge test data, hvis `iMaxNamesCount` er mindre end 10.
- `int iMaxNamesCount`: Bruges til at sætte, hvor mange navne der max må være i `ListBox`'en. Så hvis `iMaxNamesCount` = 10 må der max være 10 navne i listen.
- `List<string> INames`: Er den liste der bruges til at opbevare data i som vises i `ListBox`'en i GUI'en.

## Konstruktøren

Her har jeg tilføjet, så når GUI bliver vist at der er tilføjet test data, hvis `bTest` = `true` og at `ListBox`'en sorter alfabetisk.

## Tilføj navn knap (`btnRemovePosition_Click`)

Jeg har tilføjet noget fejlhåndtering, der tjekker om brugeren har indtastet noget tekst i tekstboksen og om der er plads til at man kan tilføje flere elementer til listen.

Der vil blive vist en `MessageBox`, der siger om brugeren har glemt at indtaste et navn eller om der ikke er plads i listen.

## Fjern position knap (`btnRemovePosition_Click`)

Da koden er skrevet om fra array til liste, vil den første position i `ListBox`'en være 0.

Jeg har tilføjet noget fejlhåndtering, der tjekker om brugeren har indtastet en position, om det er et tal og positionen findes i listen.

Det vil blive vist i en `MessageBox`, hvis brugeren har indtastet noget ugyldigt.

Når der skal fjernes en position i listen `INames` findes først det navn der står i `ListBox`'en og så fjernes det navn i listen.

Dette gøres da det på gælden navn ikke nødvendigvis har den samme position i både listen og ListBox'en.

### Fjern markeret navn knap (btnRemoveSelectedName\_Click)

Der tjekkes om der er markeret en item i ListBox'en og ud fra dette vil pågældende navn blive fjernet fra listen, ligesom der gør når man trykker på "Fjern position knap (btnRemovePosition\_Click)" knappen.

### Sorterings knapper (btnSortAscending\_Click og btnSortDescending\_Click)

For at ændre sorteringen skal man først fjerne den nuværende SortDescriptions før man kan sætte en ny.

Det er ikke nødvendigt at man sorterer listen i C# koden, data vil automatisk blive sorteret i ListBox'en.

### Metode: UseTestData

Bruges til at indsætte test data i listen.

For at gøre det læsevenligt bruges der et array der loopes igennem og indsætter data i listen.

### Metode: SetListBoxItems

Bruges til at indsætte data i ListBox'en i GUI'en.

For ikke at gentage de samme to linjer kode flere steder, har jeg centraliseret det i en metode man kan kalde, hver gang listen skal opdateres.