**Opgave: Rejsebureau**

# Introduktion

Formål med opgaven:

* Problemformulering og kravspecifikation i denne opgave ligner dem i eksamensopgaven.
* At løse denne opgave forbereder dig godt på eksamen.

Fremgangsmåde:

* Læs opgavestillingen nøjagtigt!
* Find detaljerne hvordan man gør i vores isbutik-opgave. Vi løser stort set de samme problemer som i isbutik-opgaven men nu får du ingen detaljerede anvisninger og heller ingen eksempel-kode.
* Kopiere skamløst kode fra din isbutik-løsning eller fra lærerens sidste isbutik-milepæl :-)
* Kommenter din kode rigeligt så du forstå den stadig når du kommer til at genbruge den. Dine kommentarer vil også være meget nyttige, når du skal forklare din kode til eksamen.

Hvis du sidder fast med noget:

* Læs altid hele delopgaven før du begynder at løse den.
* Research selv (google, w3schools.com, stackoverflow.com, …).
* Spørg de andre elever.
* Spørg læreren. Senest når du har siddet fast i en opgave i en halv time, spørg læreren eller send læreren en Teams Chat besked: ”*jeg sidder fast i opgave <nr>*”.

# Problemformulering

Vi skal udvikle et program for et rejsebureau.

Den skal have en brugergrænseflade til håndtering af oprettelse og vedligeholdelse af kundekartoteket og af de rejsearrangementer kunderne kan booke i rejsebureauet. Derudover kan man i programmet gemme, redigere og slette bookinger.

# Kravspecifikation

De følgende oplysninger skal gemmes i en SQL-database og håndteres af vores program:

Om Rejsebureauets Kunder:

* Efternavn
* Telefonnummer

Om de rejser som kunderne kan booke:

* Feriemål (bynavn)
* Startdato
* Slutdato
* Pris (i danske kroner, for en person)
* PladsKapacitet (for så mange personer er der plads ved denne rejse)

Om bookinger:

* KundeId
* RejseId
* Pladser

Alle nævnte oplysninger kan man oprette, redigere og slette i GUIen. (Sletning kunne være fuldstændigt eller logisk. Du beslutter hvad giver mere mening.)

Før man kan gemme en ny booking skal det tjekkes om rejsen har nok frie pladser tilbage.

Ved klik på Button ”Beregn salg” vises i en felt med overskrift ”Salg” den opsummerede beløb af alle bookinger.

# Opret et nyt WPF-projekt

* WPF App .NET Framework
* Project name: Rejsebureau

Det er en god idé at åbne din isbutik-solution eller lærerens sidste isbutik-milepæl parallelt in en separat Visual-Studio-instans så du kan sammenligne og genbruge kode.

# Planlæg databasen

Åbn [isbutik database udkast](https://intranet.aspit.dk/Fagets_Univers/Softwarekonstruktion/Documents/Materialer/ULSC/S2/isbutik%20database%20udkast.xlsx?d=w5089b46dbb4e4939b37b1bce722f5549) og benyt det som en forlæg for din udkast af rejsebureau-databasen.

* Hvilke tabeller har du brug for? (Teksten i den næste delopgave hjælper.)
* Hvilke kolonner har du brug for i tabellerne?
* Hvilke datatyper har kolonnerne i SQL?
* Hvilke datatyper har de tilsvarende properties i C#?
* Hvilke forbindelser er der mellem tabellerne (via fremmednøgler)?

Send læreren dokumentet via Teams Chat. Snak med læreren om din udkast før du opretter databasen.

# Opret databasen

* Database Name: RejsebureauDB
* Tabel dbo.Kunder
* Tabel dbo.Rejser
* Tabel dbo.Booking

Send læreren en Teams Chat besked: *jeg er færdig med ”opret databasen”*

# Opret lagdeling

* GUI-lag: MainWindow
* Funktionalitets-lag: RejseFunc
* Data-lag: RejseData

Send læreren en Teams Chat besked: *jeg er færdig med ”opret lagdeling”*

# Tilføj dataset fra database

Tilføj dine tabeller til datasettet.

Skaf dit data-lag adgang til datasettet.

Send læreren en Teams Chat besked: *jeg er færdig med ”opret dataset”*

# Opret klasser til data

* Kunder
* Rejser
* Booking

Deklarer også lister som kan indeholde objekter af disse klasser som variabler i klassen MainWindow.

Send læreren en Teams Chat besked: *jeg er færdig med ”opret klasser til data”*

# Skriv CRUD-funktioner

* Skriv de CRUD-metoder du har brug for.
* Tilføj test data manuelt i database tabeller.
* Test dine databasefunktioner.

Send læreren en Teams Chat besked: *jeg er færdig med ”skriv CRUD-funktioner”*

# Opret GUI

Følg kravspecifikationerne nøjagtig!

En god fremgangsmåde er:

1. Blev klar over hvilke GUI-objekter du har brug for.
2. Byg GUI med disse objekter.
3. Opret (tomme) funktioner, som bliver udført ved GUI-interaktioner (Klik, Udvalg, …).
4. Tilføj kode til disse funktioner og skriv funktioner til funktionalitets-laget.

Send læreren en Teams Chat besked: *jeg er færdig med ”opret GUI”*

# Gør dit program mere robust

Find et eller to eksempler hvor brugeren kunne gøre noget forkert og ville crashe programmet. Forhindre at programmet crasher ved denne fejlbetjening.

Send læreren en Teams Chat besked: *jeg er færdig med ”gør programmet mere robust”*

# Opret unit testing

Skriv nogle unittests til dit projekt efter AAA-princippet og kør dem.

# Færdig!

Tillykke :)

Send læreren en Teams Chat besked: ”*jeg er færdig med rejsebureauprojektet”*