

## IDENTIFIKATION: Sommerhus#1 / PELE

### Overordnede mål

Ideen er at du skal afprøve checkbokse og radio-knapper, samt søgning og sortering i RazorPages. I første omgang bliver det med filtrering.

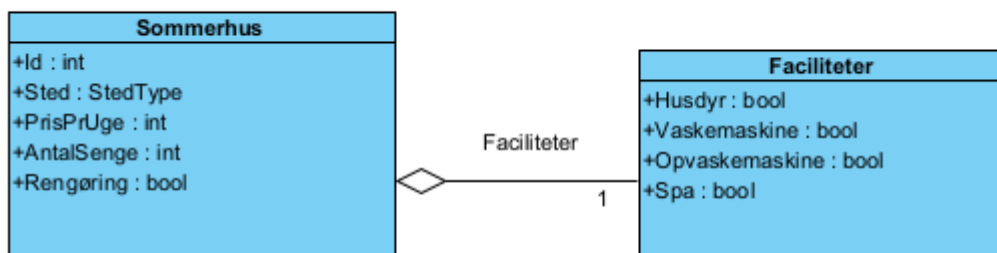
### Opgave-Beskrivelsen: Lav en applikation af Sommerhus-udlejning

Fra Github skal du downloade et model library der allerede indeholder to klasser, samt en enum-type. Desuden er der i MockData lavet 40 sommerhuse.

#### Du skal downloade fra dette Github:

<https://github.com/rf22da2b1-1b-DAT-1B-E22/Sommerhus>

Klasser kan du se i dette Design Klasse Diagram:



Enum typen er defineret som:

```
public enum StedType
{Bornholm, Nordjylland, Vestjylland, Sønderjyllend, Fyn, Vestsjælland, Falster, Øvrige}
```

#### Opgave 1: Lav en Sommerhus Applikation (RazorPage)

1. Lav et nyt C# projekt fx 'SommerhusApp' af typen 'ASP.NET Core Web App'
2. Lav en reference til dll-filen i det library du har downloadet (Husk at 'bylde' libreriet først)
3. Lav et services folder/mappe hvor du definerer et interface ISommerhusRepositoryService med én metode:  

```
public List<Sommerhus> GetAll(),
```

samt en klasse SommerhusRepositoryServic, der implemterer interfacet.

Du skal benytte SommerhusMockData.GetSommerhuse() til at initialisere din liste.

4. I Program.cs lav en dependency injection til din service dvs.

```
builder.Services.AddSingleton<ISommerhusRepoistoryService,  
SommerhusRepoistoryService>()
```

5. Lav en folder/mappe under pages 'sommerhuse'
6. I denne mappe lav en ny RazorPage (Index)
7. Sørg for at din controller-del i konstruktøren
  - a. får 'injected' din ISommerhusRepositoryService og gemmer den i en private instans felt.
  - b. Lave en property Sommerhuse af typen List<Sommerhus>
  - c. I OnGet initialiser Sommerhuse propertien med din service
8. I view-delen
  - a. Lav en tabel der viser alle sommerhusene
9. Prøv om din applikation kører  
(ellers få den til at køre 😊)

## Opgave 2: Lav en filtrering

Du skal nu lave at din side kan understøtte at brugeren kan skrive en Max Pris og kun se dem der opfylder dette.

Det skulle se nogenlunde sådan her ud:

Id	Sted	Antal Sengepladser	Rengøring	Pris pr uge	Husdyr tilladt	Vaskemaskine	Opvaskemaskine	Spa
1	Bornholm	4	Inkluderet	7600	Ja	Ja	Ja	Nej
2	Bornholm	3	Uden	7810	Ja	Nej	Ja	Ja
4	Nordjylland	3	Inkluderet	5100	Nej	Nej	Nej	Nej
6	Vestsjælland	6	Uden	6000	Nej	Ja	Ja	Nej
7	Vestsjælland	5	Inkluderet	5600	Ja	Nej	Nej	Nej
9	Nordjylland	5	Uden	8120	Nej	Ja	Ja	Nej
11	Vestjylland	3	Uden	6600	Nej	Nej	Ja	Ja
13	Vestsjælland	4	Inkluderet	5540	Ja	Nej	Ja	Nej
15	Vestsjælland	3	Uden	4600	Nej	Nej	Nej	Nej
17	Nordjylland	6	Uden	7700	Ja	Ja	Ja	Nej
20	Vestsjælland	3	Uden	4400	Nej	Nej	Nej	Nej
22	Vestsjælland	8	Uden	6000	Ja	Nej	Ja	Nej
30	Vestjylland	5	Inkluderet	7900	Nej	Ja	Ja	Nej
31	Nordjylland	6	Inkluderet	8300	Nej	Ja	Ja	Nej
33	Vestsjælland	3	Uden	6300	Ja	Nej	Ja	Nej
34	Øvrige	5	Inkluderet	4400	Nej	Nej	Ja	Nej
35	Vestsjælland	2	Uden	3900	Nej	Nej	Nej	Nej

### 1. I controller-delen

- Lav en BindProperty MaxPris af typen int.
- Lav en metode OnPostFilterMax  
Den skal tildele property Sommerhuse (fra sidste opgave) de filtrerede huse.  
Til dette benyt `_service.GetAll` til at få alle sommerhuse.  
Benyt Linq-udtrykket 'where' til at filtrere  
(fx `Where( s ) => s.PrisPrUge <= MaxPris` )  
Husk så at lave det til en Liste (dvs. `ToList<Sommerhus>()` )

### 2. I View-delen

- Lav en form der går til `OnPostFilterMax` (dvs: `asp-page-handler="FilterMax"` )
- I denne form lav et input felt – bundet til property MaxPris, samt en knap

### 3. Prøv om din applikation kører (ellers få den til at køre 😊)

**Ekstra:**

- Lav desuden et felt **MinPris** det er den nedre grænse
- Lav en knap der igen viser alle sommerhuse