

**Færdighedsprøve**  
**i Fagene**  
**Software Construction**  
**og**  
**Software Design**

**1 semester 8. juni 2021**

*Domæne beskrivelse*

FDF –k19 Vanløse, der hvert år holder sommerferie på Klinteborg for 170 børn, unge og voksne, ønsker et system kaldet **Tilmeldingsystemet**, der kan tilmelde børn, unge og voksne til sommerlejren enten i en uge eller 2 uger.



I systemet skal der være en klasse **Tilmeldingskatalog**. Klassen skal indeholde følgende properties:

Property	Type	Beskrivelse
LejrID	string	F.eks. "Julelejr 2021", "Sommerlejr 2022"...
UgeRegel	bool	<ul style="list-style-type: none"><li>• Hvis <b>true</b>, er ugereglen aktiv: Man kan være tilmeldt 1 eller 2 uger.</li><li>• Ellers hvis <b>false</b>: Reglen gælder ikke.</li></ul>
Startdato	DateTime	Startdato for lejren
Slutdato	DateTime	Slutdato for lejren

Ligeledes skal der oprettes en klasse **Person**, der indeholder oplysninger om den enkelte tilmeldte person.

Klassen **Person** skal indeholde følgende properties:

Property	Type	Beskrivelse
Id	int	ID på Deltager
Navn	string	Navn på deltager
Adresse	string	Adresse på deltager
Patrulje	string	Deltagers patrulje
Fødselsdato	DateTime	Deltagers fødselsdato

EnUge	bool	Der skal kun tages hensyn til denne property, hvis UgeRegel er <b>true</b> i det <b>Tilmeldingskatalog</b> , som en <b>Person</b> tilhører. <ul style="list-style-type: none"><li>• Hvis <b>true</b>, skal den pågældende person være tilmeldt for en uge.</li><li>• Ellers hvis <b>false</b>, skal den pågældende <b>Person</b> være tilmeldt i 2 uger.</li></ul>
-------	------	--

Klassen **Tilmeldingskatalog** skal indeholde en liste af **Personer**.

Bemærk: En **Person** oprettes påny, hvis denne bliver tilknyttet et **Tilmeldingskatalog**.

### Opgave 1. Opret Klassen Person.

- Opret et nyt Console App (.NET Core) projekt, **Tilmeldingsystemet**
- Implementer klassen **Person** med instansfelter, properties og passende konstruktører.
- Klassen skal have en ToString() metode.
- Test klassen ved at oprette nogle instanser af den i Main metoden. Skriv hver enkelt instans ud til konsollen.

### Opgave 2. Opret klassen Tilmeldingskatalog

En instans af **Tilmeldingskatalog** klassen skal indeholde en Dictionary (af typen Dictionary) af **Person** objekter. Denne liste skal instantieres i konstruktøren.

- Implementer klassen **Tilmeldingskatalog** med instansfelter, properties og passende konstruktører.
- Klassen skal have en ToString() metode, som returnerer alle oplysninger i et **Tilmeldingskatalog**. I ToString() metoden skal du anvende en foreach-løkke, hvormed du gennemløber **Personerne** til **Tilmeldingskataloget**.

### Opgave 3: Lav et sekvensdiagram

Tegn et sekvensdiagram for et kald af metoden AddPerson. Udgangspunktet for sekvensdiagrammet er Main metoden og sekvensdiagrammet skal kun vise et enkelt kald af AddPerson og vise, hvorledes der oprettes et objekt af typen **Person** og at det tilføjes til listen i **Tilmeldingskataloget**.

#### Opgave 4: Udvid Tilmeldingskatalog klassen med nogle CRUD metoder

- Tilføj metoden *void AddPerson(Person person)* på klassen **Tilmeldingskatalog**.
- Tilføj metoden *void DeletePerson(int id)* på klassen **Tilmeldingskatalog**. Metoden sletter **Personen** med det pågældende Id (= parameteren id).
- Test klassen **Tilmeldingskatalog** ved at oprette 2 instanser af den i Main metoden. Skriv hver enkelt instans ud til konsollen. Husk at teste begge metoder - både *AddPerson* og *DeletePerson* metoderne

#### Opgave 5. UML – Tegn et klassediagram

Lav et klassediagram og redegør for de valgte associationer

#### Opgave 6. Exceptions

Klassen **Tilmeldingskatalog** skal kaste exceptions i følgende tilfælde:

1. Hvis StartDato er senere end SlutDato
2. Hvis UgeRegel er **true** og periode fra StartDato til SlutDato ikke dækker præcis 2 uger

Udvid klassen så der kastes en *ArgumentException()* med en passende meddelelse, hvis ovenstående ikke overholdes. Overvej at udvide de nævnte properties med exception handling.

Test exception håndteringen i Main metoden

#### Opgave 7 Implementation af arv

Man kan tilmelde sig som **PatruljeFører**, som også er en **Person**:

- Implementér en **PatruljeFører** klasse som arver fra klassen **Person**.
- Implementer en property *bool OK* i **PatruljeFører** klassen, som checker **PatruljeFørerens** forudsætninger. Der checkes indtil videre kun på alder:
  - OK er **true**, hvis **PatruljeFøreren** er fyldt 18
  - Ellers er OK **false**
- Test **PatruljeFører** klassen i Main metoden

#### Opgave 8 Lav dine egne user stories

Lav et par user stories, der relater sig til **Tilmeldingssystemet**

**Opgave 9 Implementér dine egne user stories**

Implementer de user stories, du definerede i opgaven ovenover.

**Opgave 10 Anvendelse af for-løkke**

Implementér ToString() metoden i klassen **TilmeldingsKatalog** vha. en for-løkke.

**Opgave 11 Brug Dictionary i stedet for List**

Implementér klassen vha. en Dictionary af **Personer** i stedet for en Liste af **Personer**.