

# Test af viden og færdigheder

Obligatorisk opgave 2022 januar

Vagtplan System



# Vagtplan System

## Case beskrivelse

Covid19 test centret i Roskilde skal have et system til at håndtere forskellige vagter for personer der skal foretage test.

Overordnet skal vagtplan systemet kunne håndtere forskellige personer. Desuden skal systemet kunne holde styr på forskellige vagter på de forskellige dage. Vagterne for de enkelte personer kan have forskellige længder. Testcentret har åbent alle dage mellem 8 og 20.

## Overordnede System

På baggrund af denne overordnede beskrivelse skal du lave tre opgaver, senere kommer flere opgaver til en første version af systemet.

### Opgave 1: User stories (swd)

Du skal skrive 3 user stories med acceptance kriterier til Vagtplan systemet.

### Opgave 2: Domain model (swd)

Du skal lave en domain model for centrale dele af Vagtplan systemet, hvor der mindst indgår 5 klasser.

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

### Opgave3: Design klasse diagram (swd)

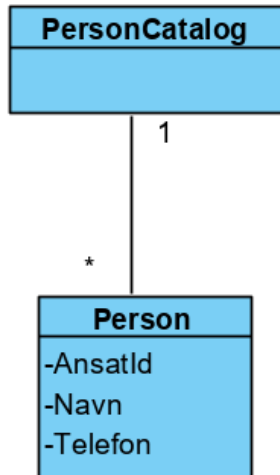
Du skal nu lave et Design klasse diagram ud fra ovenstående domain model.

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

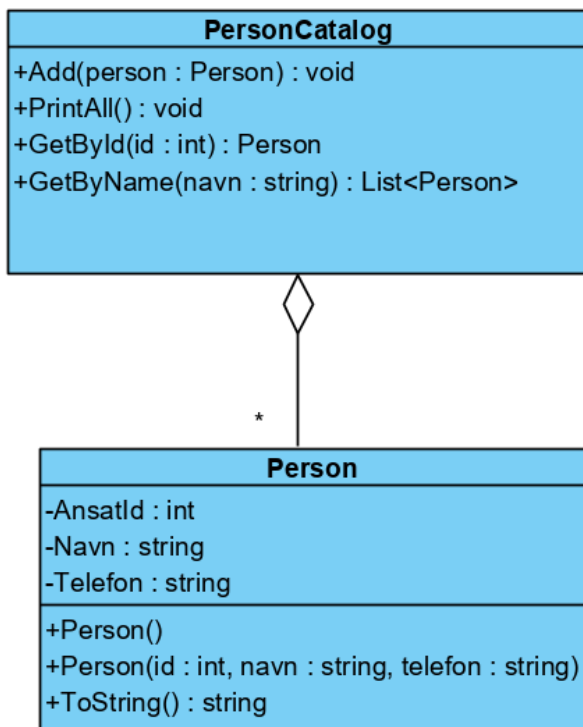
## Første version af Vagtplan Systemet

Det er fra product owner besluttet at lave en første version af systemet som indeholder håndteringen af personer og dertil er der udarbejdet en domain model og et design klasse diagram

Første version: domain model



Første version: Design Klasse Diagram



#### Opgave 4: Implementer klassen Person (swc)

Du skal implementere klassen Person

1. Opret en consol application (.Net Core) 'VagtplanApp'
2. Lav en folder (underkatalog) 'model'
3. I 'model' folderen opret en klasse 'Person'
4. Klassen skal indeholde de tre properties 'AnsатId', 'Navn' og 'Telefon'  
dog skal det kun være muligt at læse (get) 'AnsатId'
5. Lav to konstruktører
  - a. En default konstruktør (uden parametre)
  - b. En konstruktør til at sætte de tre properties
6. Lav en ToString metoder der returnere property værdierne i en string.

#### Opgave 5: Opret objekter af Person (swc)

Du skal i Main oprette tre forskellige objekter af klassen Person, samt udskrive disse på skærmen.

#### Opgave 6: Implementer klassen PersonCatalog (swc)

Du skal implementere klasse PersonCatalog i folderen 'model'

1. Opret klassen PersonCatalog
2. Klassen skal indeholde en collection til at opbevare personer  
Du vælger selv om du vil benytte List eller Dictionary
3. Lav en default konstruktør, hvor collection'en initialiseres
4. Du skal implementere de fire metoder
  - a. Add(Person nyPerson) som tilføjer en ny person til collection'en.
  - b. PrintAll() som udskriver alle personer på skærmen en ad gangen.
  - c. GetById(int id) som finder og returnere en person med den specificerede id  
(hvis der ikke findes en person med id'en kastes en exception – dette behandles senere i opgave 10).
  - d. GetByName(string navn) som finder og returnere alle personer med det specificerede navn. (hvis der ikke findes ingen person med navnet kastes en exception – dette behandles senere i opgave 10)

#### Opgave 7: Anvend PersonCatalog (swc)

Du skal anvende klassen PersonCatalog i main

1. Du skal i Main oprette et objekt af klassen PersonCatalog.
2. Du skal tilføje de tre objekter fra opgave 5 til PersonCatalog-objektet.
3. Du skal udskrive alle personerne til skærmen ved hjælp af PersonCatalog-objektet.
4. Du skal finde og udskrive en person med et givet id.
5. Du skal finde og udskrive personer med et givet navn.

### Opgave 8: design Arv (inheritence) (swd)

Du skal udvide den første version så det bliver muligt at have nogle specielle personer 'Planner', som skal have adgang til at lægge vagtplaner. En 'Planner' har de samme attributter/properties som 'Person', desuden har de en property 'TestSted', der angiver hvilket test center de tilhører.

Du skal vise i et design klasse diagram, hvorledes forholdet mellem 'Person' og 'Planner' vises i diagram form. Du kan udvide design klasse diagrammet for første version som er vist tidligere.

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

### Opgave 9: Implementer Arv (swc)

Du skal implementere klassen Planner

1. Lav en klasse Planner, der arver fra klassen Person
2. lav en property 'TestSted'
3. Lav en konstruktør til at sætte relevante værdier
4. Lav en ToString metode til at returnere et planner objekts værdier.

Du skal desuden i main oprette et objekt af klassen Planner og tilføje det til PersonCatalog, samt udskrive alle objekter i PersonCatalog.

### Opgave 10: Exceptions (swc)

Du skal understøtte at dit program ved fejlsituationer kaster en passende exception.

I de to metoder GetById og GetByName skal metoderne kaste en passende exception (du vælger selv hvilken), hvis der ikke findes en person med det specificerede id hhv. navn.

Du skal i main vise; hvordan du kan håndtere, at du kalder de to metoder med et id hhv. et navn, der ikke findes, samt udskrive en passende fejl meddelelse på skærmen.