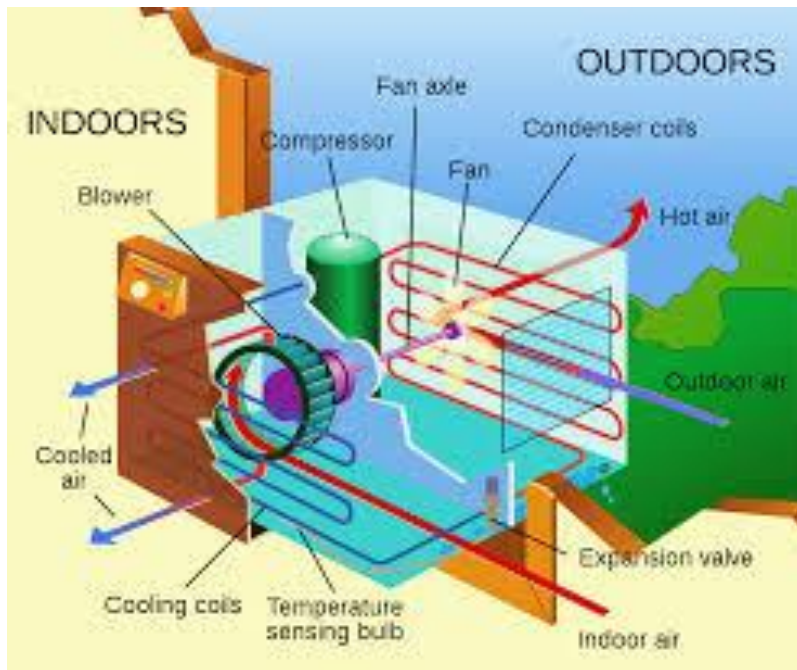


# Zealand Climate System

'Det klimavenlige klasselokale'



## Opgavebeskrivelse

På Zealand - Sjællands Erhvervsakademi, ønsker vi at have behagelige og klimastyrede lokaler. Derfor vil Zealand gerne have udviklet et program '**ZealandClimate**', der kan understøtte klimaet i lokalerne.

I hvert lokale er der installeret sensorer der kan måle temperatur, CO<sub>2</sub> og antal personer i lokalet. Disse sensorer kan senere benyttes som input til systemet **ZealandClimate**.

I den første prototype skal der kunne registreres målinger af temperatur, CO<sub>2</sub> og antal personer i et lokale. Hver måling skal have et unikt Id, et 'timestamp' så tidspunktet for målingen også registreres samt en angivelse af i hvilket lokale målingen er aflæst.

For et lokale skal registreres følgende oplysninger: unikt Id, lokalenummer (fx D. 3.06) og maxAntal (det antal personer lokalet er dimensioneret efter fx 36).

Nedenstående er vist et klassediagram for Målinger og Lokaler:



## Opgave 1 - Implementering

I denne opgave skal du implementere ovenstående klasse-diagram.

1. Opret et nyt *ASP.Net Core console* projekt, **ZealandClimate**
2. Tilføj klassen **Lokale** med de viste properties samt passende konstruktør(e) og *ToString()*-metode.  
**Hint:** Du kan evt benytte et static instance field *nextId* til at initialisere (og incrementere) propertyen: Id
3. Tilføj klassen **Måling** med de viste properties, den viste associering til **Lokale** samt passende konstruktør(e) og *ToString()*-metode  
**Hint:** Du kan evt benytte et static instance field *nextId* til at initialisere (og incrementere) propertyen: Id
4. Afprøv klassen **Måling** i *Main*-metode, hvor der oprettes et nyt **Måling** objekt og hvor målingens properties udskrives.

## Opgave 2 programmets dynamiske forhold.

Tegn et sekvensdiagram som viser din afprøvning fra opgave 1.4, hvor du oprettede en måling i main metoden og udskriv indholdet af måling objektet.

## Opgave 3 (MålingRegister - Array, List eller Dictionary)

Der skal oprettes et register (en klasse) til alle målingerne: **MålingRegister**

Registret skal kunne gemme alle Målingerne i en collection - spørgsmålet er hvilken?

### 1. **Array:**

Giv en kort beskrivelse af et **Array** herunder fordele/ulemper i forhold til at benytte det i klassen **MålingRegister** til at opbevare **Måling**-objekter.

### 2. **List:**

Giv en kort beskrivelse af klassen **List**, herunder fordele/ulemper i forhold til at benytte det i klassen **MålingRegister** til at opbevare **Måling**-objekter.

### 3. **Dictionary:**

Giv en kort beskrivelse af klassen **Dictionary**, herunder fordele/ulemper i forhold til at benytte det i klassen **MålingRegister** til at opbevare **Måling**-objekter.

## Opgave 4 (MålingRegister - Implementering)

Det er besluttet at der skal benyttes en Liste til at opbevare **Måling**-objekter i klassen **MålingRegister**.

1. Opret klassen **MålingRegister** med en List af **Måling** som instans felt.  
Klassen skal have en konstruktør der initialiserer instans feltet med et nyt List objekt.
2. Opret en instanse af **MålingRegister** i Main.

## Opgave 5 (MålingRegister - CRUD metoder)

Det skal være muligt at oprette nye **Målinger** og indsætte dem i Listen samt hente målingerne ud igen.

1. Implementer metoden **OpretMåling(...)**  
**OpretMåling(..)** skal have passende parametre, så metoden kan oprette et nyt **Måling**-objekt og tilføje det til Listen.  
**Hint:** Du kan evt benytte *DateTime.Now*, til at initialisere *DateTime* i **Måling**.
2. Implementer metoden **HentMåling(int id)**  
Metoden skal kunne gennemløbe listen og returnere **Målingen** med det givne id eller null, hvis der ikke findes en måling med dette id.
3. Implementer metoden **PrintAlleMålinger()**  
Metoden skal kunne gennemløbe listen og udskrive hvert enkelt måling.

4. Afprøv programmet i main-metoden.  
I Main skal oprettes mindst tre målinger, som skal tilføjes til MålingRegister. Desuden skal oplysninger om målingerne skrives til konsolvinduet ved at kalde PrintAlleMålinger. Desuden skal du vise hvordan du finder en bestemt måling ud fra et id og udskrive denne.

### **Opgave 6 (Find antal målinger der overskrider max-CO2)**

Ifølge Arbejdstilsynet må CO<sub>2</sub> indholdet i et lokale ikke overstige 1000 ppm.

1. Implementer metoden *AntalUlovligeCO2Målinger()*  
Metoden skal returnere antal målinger med et for højt CO<sub>2</sub> niveau.

### **Opgave 7 (Find alle målinger uden for temperatur-interval) (svær)**

Det skal være muligt at finde alle de målinger der indeholder temperaturer uden for et interval fx 20-22 grader

1. Implementer metoden *AlleTempUnderOver(int min, int max)*, metoden skal returnere en liste med alle de målinger, hvor temperaturen er under min eller over max grader

### **Opgave 8 User story**

Skriv en user story der omhandler metoden *AllTempUnderOver(int min, int max)* fra ovenstående opgave 7

### **Opgave 9 Opdater klassediagram med MålingRegister**

Opdater dit klassediagram med klassen MålingRegister der viser de metoder du implementerede i ovenstående opgaver

### **Opgave 10 User Story**

Skriv mindst 2 nye user stories med acceptance criteria for ZealandClimate.

### **Opgave 11 Exceptions**

Opdater din klasse Måling, så der kastes en *ArgumentException* hvis der er "ulovlige" argumenter fx  $0 < \text{temp} > 50$  ol.

### **Opgave 12 Implementer dine user stories fra opgave 10**

