

# IDENTIFIKATION: Datastrukturer / PELE

## Overordnede mål

Ideen er at du skal afprøve forskellige datastrukturer.

## Baggrunds Materiale:

C# note of Per Laursen from 1<sup>st</sup> semester: (<https://github.com/perl-easj/OOProg>)

Kapitlerne: Prog2 – Introduction to Data structures og Prog3 – Data Structures revisited

Slides: [AbstractDataTypes.pdf](#)

## Opgave-Beskrivelsen: Brug af forskellige Datastrukturer

### Formål

Formålet med denne opgave er at du kender flere datastrukturer og kan anvende dem, samt ved hvornår den ene er at foretrække for den anden.

### Konkret

Du skal lave:

- a. Implementere en simpel model klasse i et C#-Class-Library
- b. Implementere en Konsol-applikation, hvor forskellige datastrukturer afprøves.

### Domæne beskrivelse

Du skal implementere forskellige datastrukturer til håndtering af biler.

## Opgave 1: Lav en model klasse i et C# Library

Du skal lave en solution 'Datastrukturer' der har to projekter det første er et Library (Class Library).

I Library projektet lav en 'folder' model, hvori du implementerer en klasse 'Car'.

En 'Car' har følgende egenskaber (properties, der behøver IKKE være instans felter):

- ID – et heltal
- Color – en tekst streng (string)
- RegistrationNo – en tekst streng (string)
- Price en double

Alle egenskaber skal have både en get- og en set-metode, der skal være en konstruktør, samt en 'ToString'-metode.

### Domæne beskrivelse

Bilforhandler B.Lakket.Ry ønsker at holde styr på sine biler. Men er usikker på hvilken datastruktur han skal benytte. Han ønsker desuden følgende (**slags** funktions liste / *userStories*):

1. B.Ry ønsker at alle biler kan udskrives.
2. B.Ry ønsker at en bil med et ID kun findes en gang i datastrukturen (ingen dubletter)
3. B.Ry ønsker at finde alle røde biler og få dem udskrevet
4. B.Ry ønsker bilen med ID = xx fundet og udskrevet
5. B.Ry ønsker bilen med RegistreringsNr = yy fundet og udskrevet
6. B.Ry ønsker bilen der er ældst, den der først er ankommet (dvs. sat ind i listen) fundet og udskrevet
7. B.Ry ønsker bilen der er nyest, den der sidst er ankommet (dvs. sat ind i listen) fundet og udskrevet

## Opgave 2: Lav et Interface

I dit Library skal du lave endnu en folder 'RyCars', hvor du skal lave et interface 'IRyCars'

Som indeholder følgende metoder (til at understøtte ovenstående domæne beskrivelse):

```
public ICollection<Car> GetAllCars();
public void Add(Car car);
public ICollection<Car> FindRedCars();
public Car FindCarById(int id);
public Car FindCarByRegNo(string regNumber);
public Car FindOldestCar();
public Car FindNewestCar();
```

## Opgave 3: Lav en klasse, der implementerer IRyCars-interfacet med en Liste

I dit library i folderen RyCars lav en klasse 'RyCarsList', der implementerer IRyCars interfacet vha. en Liste

## Opgave 4: Lav en Konsol Applikation

I din solution lav et projekt (Console Application) 'DatastrukturerApp'.

Sørg for at refererer til dit Library.

Du skal lave et objekt af din klasse 'RyCarsList' og prøv de forskellige metoder.

Hint: Lav din variable som

```
IRyCars cars = new RyCarsList(); // så kan du i senere opgaver 'bare' skifte new-delen ud
```

Kald metoderne og udskriv resultatet på skærmen (Konsollen).

## Nu skal du prøve de andre datastrukturer

### Opgave 5: Implementerer IRyCars-interfacet med en LinkedList

I dit library i folderen RyCars lav en klasse 'RyCarsLinkedList', der implementerer IRyCars interfacet vha. en LinkedList

Prøv klassen af i din consol-applikation

### Opgave 6: Implementerer IRyCars-interfacet med en Kø (Queue)

I dit library i folderen RyCars lav en klasse 'RyCarsQueue', der implementerer IRyCars interfacet vha. en kø (Queue)

Prøv klassen af i din consol-applikation.

### Opgave 7: Implementerer IRyCars-interfacet med en stak (Stack)

I dit library i folderen RyCars lav en klasse 'RyCarsStack', der implementerer IRyCars interfacet vha. en stak (Stack)

Prøv klassen af i din consol-applikation.

### Opgave 7: Implementerer IRyCars-interfacet med en HashSet

I dit library i folderen RyCars lav en klasse 'RyCarsHashSet', der implementerer IRyCars interfacet vha. en HashSet

Prøv klassen af i din consol-applikation.

### Opgave 8: Implementerer IRyCars-interfacet som et Dictionary

I dit library i folderen RyCars lav en klasse 'RyCarsDictionary', der implementerer IRyCars interfacet vha. et Dictionary

Husk ved metoden GetAllCars at lave Values om til fx en Liste

Prøv klassen af i din consol-applikation.

### Opgave 9: Hvilken datastruktur passer bedst

For hver af de 7 ønsker fra B.Ry vurder hvilke datastrukturer, der er gode hhv. dårlige og hvilke kan slet ikke benyttes.

**Konklusion: Set i en samlet vurdering, hvilken datastruktur skal B.Ry anvende?**