IDENTIFIKATION: Datastrukturer / PELE

Overordnede mål
Ideen er at du skalafprøve forskellige datastukturer.

Baggrunds Materiale:
C# note of Per Laursen from 1st semester
(https://github.com/perl-easj/Teaching-materials/tree/master/CSharpProgramming/Notes)
Kapitlerne: Prog2 – Datastructures I og Prog3 – Datastructures II

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#-library
b. Implementere en Konsol-applikation, hvor forskellige datastrukturer afprøves.

Domæne beskrivelse
Du skal implementere forskellige brug af samlinger 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 .Net Core).

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

En 'Bil' har følgende egenskaber (properties):

- ID – et heltal
- Farve – en tekst streng (string)
- RegistreringsNr – en tekst streng (string)
- Pris en double

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

Opgave 2: Lav en Konsol Applikation

I din solution lav et projekt (Console Application .Net Core) 'DatastrukturerApp'.
Sørg for at refererer til dit Library.

Evt: lav en worker klasse 'DataWorker' med en Start metode.

Bilforhandler B.Lakket.Ry ønsker at holde styr på sine biler. Men er usikker på hvilken datastruktur han skal benytte. Han ønsker desuden følgende:

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 3: Lav en Metode til at behandle en Liste

Lav en metode 'BilerIListe'. I denne metode skal du oprette en liste af biler (List<Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

Opgave 4: Lav en Metode til at behandle en Dictionary

Lav en metode 'BilerIDictionary'. I denne metode skal du oprette en liste af biler, ID er nøglen (Dictionay<int, Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

Opgave 5: Lav en Metode til at behandle en LinkedListe

Lav en metode 'BilerILinkedListe'. I denne metode skal du oprette en liste af biler (LinkedList<Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

Opgave 6: Lav en Metode til at behandle en Kø (Queue)

Lav en metode 'BilerIKø'. I denne metode skal du oprette en liste af biler (Queue<Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

Opgave 7: Lav en Metode til at behandle en Stak (Stack)

Lav en metode 'BilerIStak'. I denne metode skal du oprette en liste af biler (Stack<Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

Opgave 8: Lav en Metode til at behandle en HashSet

Lav en metode 'BilerISet'. I denne metode skal du oprette en liste af biler (HashSet<Bil>) samt indsæt nogle biler i listen.

Herefter implementer og undersøg de 7 ønsker fra B.Lakket.Ry.

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.

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