

Opgave: Repository (liste)

I denne opgave skal du programmere en Repository klasse der bruger en liste.

Kom i gang

I Visual Studio lav et nyt projekt af typen Class Library. Navn *AktiehandelRepositoryLib*.

Model-klasse

Lav en simpel model-klasse, *AktieHandel*, med flg properties:

- HandelsId, et tal som er unikt
- Navn, en string, må ikke være null, og mindst 4 tegn langt
- Antal, et tal
- HandelsPris, decimal

Tilføj valideringsmetoder, der kaster relevante exceptions, hvis ovenstående betingelser ikke overholdes.

Aktie klassen kan bruges til både køb og salg. Et køb er antal et positivt tal, hvis salg et negativt tal.

Unit-test af model-klasse

Lav en unit-test af din model-klasse.

Testen skal dække både de normale (gode) tilfælde og de tilfælde, der ender med en exception.

Testen skal have et godt Code Coverage.

Repository-klasse

Lav en klasse med navnet *AktieHandelRepository*.

Klassen skal have et *private* field af typen `List<AktieHandel>`

Repository-klassen skal tildele Id'er til Aktiehandel-objekter ved indsættelsen.

Lav et private field `int _nextId`

Metoder

- `GetById(int id)`, returnerer et Aktiehandel object med pågældende id
- `GetAll()`, returnerer alle Aktiehandel objekter i listen

Aktierepository

- Get(), returnerer alle Aktiehandel objekter i listen (skal bruges til filtrering og sortering senere)
- Add(Aktiehandel Aktiehandel), tilføjer et nyt id til Aktiehandel objektet og sætter det ind i listen
Returnerer det nyt Aktiehandel objekt
- Delete(int id), sletter Aktiehandel objektet med det angivne id.
Returner det slettede Aktiehandel objekt - eller null, hvis der ikke er noget Aktiehandel objekt med det angivne id.
- Update(int id, Aktiehandel data)
Opdaterer Aktiehandel objektet med det angivne id, med de data der er i *data* parameteren.
Returnerer det opdaterede Aktiehandel objekt - eller null, hvis der ikke er noget Aktiehandel objekt med det angivne id

Obs: Husk at give sigende og brugbare Xml Comments !

Unit test af repository-klassen

Lav en unit-test af din repository-klasse.

Testen skal have et godt Code Coverage.

Prøv dit Repository DLL i en konsolapplikation

"Build" din repository DLL

Opret et nyt ConsoleProject

Kopier DLL + XML ind i projektet (husk dependency til DLL'en)

Kald dine forskellige metoder i dit AktiehandelsRepository fra din konsolApp

Søgning: Parametre til Get() metoden

Tilføj søge-parametre til Get() metoden:

- Get(decimal kurs, string? navn), returnerer alle aktiehandelobjekter med navn = navn og kurs > kurs.

NB: overvej LINQ for at løse disse

Unit test Get metoden, nu med søge-parametre

Test at dine søge-parametre virker

Sortering: Flere parametre til Get() metoden

Tilføj endnu **en** parameter til Get metoden, *sortBy*

NB: overvej LINQ for at løse disse

så du kan sortere data efter Id, name eller BirthYear (både faldende og stigende).

Tip: En [switch](#) sætning er anvendelig

EKSTRA: Unit test Get metoden, nu med sorterings-parameter.