

Opgave: Model-klasse, Interface og Katalog

Du skal fortsætte [opgave om Frugt](#), men skal nu lave repositoret generisk.

Trin 1: lav et Interface generic

- a. Du skal lave en folder 'generic'
- b. Du skal lave et interface 'IRepositoryGeneric', som indeholder de samme fem metoder som 'IFrugtRepository' blot at Det ikke er frugt men en generisk type (fx T).
- c. Du skal refactorere 'FrugtRepository' så det implementere det generiske interface
- d. Husk at 'build'e dit Library
- e. Prøv om dit program stadig virker

Trin 2: lav en abstrakt modelklasse med et id

- f. Du skal i folderen 'generic' lave en abstrakt klasse 'AbstractModelClass', som indeholder
 - i. én property Id (int) (ingen instans felter)
 - ii. to konstruktører (default, samt en med parametre)
 - iii. en ToString-metode
- g. Du skal refactorere din model-klasse frugt, så den arver fra den abstrakte klasse dvs. den 'får' Id fra den abstrakte klasse
- h. Husk at 'build'e dit Library
- i. Prøv om dit program stadig virker

Trin 3: lav et Repository generic

- j. Du skal i folderen 'generic' lave en generisk klasse 'RepositoryGeneric'
- k. Du skal refactorere dit 'IRepositoryGeneric' så den generiske type skal være en AbstractModelClass
fx: where T: AbstractModelClass
- l. Du skal implementere 'RepositoryGeneric', idet modelklassen (T) har en Id-property så alle fem metoder kan implementeres (lidt som i 'FrugtRepository')

- m. Husk at 'build'e dit Library
- n. Prøv om dit program stadig virker

Extra: Lav endnu en model-klasse 'Kunde' med et Id, Navn, Phone og Email
prøv i dit program at lave et Repository til kunder og se om det virker (oprette, get, update, delete)