Book

A Book is a base class of any type of books, it should not be able to be instantiated.

**Data**

Author – string;

Price – double;

Genre - string

**Constructor**

A Supplement should take the following values upon initialization: string Author, double Price

Class Book

it is an abstract class. It has three parameters, as follow:

Author – string;

Title – string

Price – double – it could not be equal to zero or bellow;

Genre – string – it could be only cookbook or novel.

The constructor will initializing Author, Title and Price which child class will receive trough as argument

CoockBook

The class will receive Author, Title and Price. The Genre will be set by default;

NovelBook

The class will receive Author, Title and Price. The Genre will be set by default;

OfficeSupplies

it is an abstract class. It has three parameters, as follow:

string Type – pen or pencile

string Color – the color of scribe. If it is pen the color could be only – red, blue or black. If it is a pencile the color could be red, blue, black, green, purple, yellow, brown

string Manufacturer - Manufacturer's brand;

bool IsPackage – Is it available individually or in a pack of 10;

double Price - The price must be greater than zero.

**Class Store (string name)**

it inherit **IStore Interface**: This non-generic interface contains the shared members (StoreName, Turnover, Profit, Order, and CheckWareHouseCapacity), making it compatible with the Controller class.

 **IStore<T> Interface**: The generic interface now inherits from IStore and includes the WareHouse property specific to T.

It is a base class which could not be instant

In the base class we define two paramteras which should be overwrite in child class

public virtual int WhereHouseMaxLimit => 0;

public virtual int WhereHouseMinLimit => 0;  
  
overwritten as:  
public override int WhereHouseMinLimit => 2;

public override int WhereHouseMaxLimit => 8;

Data

Warehouse - List<T> - all the goods for sale are stored there unarranged.

Turnover – double - it is an amount of all income from sales;

Profit – double it is difference between delivery and sales price;

ProfitTable – Dictionary<string, double> - there are desriptions of all markups, separated by different type of goods. When a good is ordered it should be sale with markups over delivery price;

Behavior

void Order(string kind)

If orderd kind of item is not налично в продажба throw exception message - "The {kind} is out of stock";

If it is in stock to the price of the item should add the profit according to the kind of item, add total price to Turnover, add the profit to Profit, increase number of sales and remove the item from local warehose.   
return the message:  
“{Kind of stock item} was sols on price {salsePrice}”

NB for Store and GWarehose repositories is to watch over their availability and if it is <= of 3 for store warehouse and <= 10 for GWarehose, The store to order delivery from GWarehouse and if the GWarehose is under minimum availability to produce enough for 3 Stores.

We have to monitor what type of store make their order for delivering correct items according the type of the store.

private void CheckWareHouseCapacity() – check if capacity of warehouse id equal to WhereHouseMinLimit throw throw new ArgumentException(String.Format(ExceptionMessages.OutOfStock, storeName, GetType().Name));

NB if controller receive this exception message should execute method for refill of the warehouse of define store.

**Controller class**

Implements IStore<T>, inheriting both the non-generic and generic functionality.

Possible Commands:

CreateStore(string storeType, string storeName) – create a instance of Store class which could be "BookStore" or "OfficeStore". If the type is different than the previous mentioned the program will throe exception message - "The type of the store must be BookStore, OfficeStore.";

Notes:  
in Controller if we have a command related to define class of Store we should deliver a proper type of goods. For example if the command is Delivery Office1, the method should find the type of the store snd deliver correct type of goods

Commands:

CreateStore BookStore MyStore

CreateStore OfficeStore Office2

CreateProduct Pen FC no 2.45

CreateProduct Pen FC yes 2.11

CreateProduct NovelBook Vadim Future 12.45

CreateProduct Pencil Musala no 4.45

CreateProduct Pencil Musala2 no 0.99

CreateProduct Pencil Romana yes 4.45

CreateProduct CoockingBook Vezhdi Tosts 11.45

CreateProduct CoockingBook Nelly Pizza 7.22

CreateProduct Pen Tombow no 122.45

CreateProduct Pen Fabarzhe yes 2.11

CreateProduct NovelBook Shakespeer Dialogs 8.33

CreateProduct Pencil China no 1.45

CreateProduct Pencil China2 no 0.59

CreateProduct Pencil Bulgaria yes 2.76

CreateProduct CoockingBook Vezhdi Salads 16.45

CreateProduct CoockingBook Kalina pasta 7.22

CreateProduct NovelBook Chehoff Rivers 13.93

CreateProduct NovelBook Strugatski Robots 7.73

CreateProduct NovelBook Shakespeer DialogsII 5.23

CreateProduct NovelBook ShakespeerII DialogsII 51.23

CreateProduct Pencil China21 no 9.45

Delivery Office2

Delivery MyStore

GetInventory MyStore

GetInventory Office2

Order MyStore Kalina pasta

Order MyStore Chehoff Rivers

Order MyStore Vezhdi Salads

Order MyStore Vezhdi Tosts

Order MyStore Vadim Future

Order MyStore Shakespeer Dialogs

Report MyStore

Report Office2