

IDENTIFICATION: Brushup#1 / PELE

Overall Purpose

The overall purpose for the group of 'Brushup' assignments is to be able to quick, safe and sure in creating and using model classes, methods using collections and basic string operations.

The whole group of assignments consist of XX steps:

- 1. A simple model class with constraints. (this assignment)**
2. More advanced and complex model classes.
3. Testing a model class.
4. Methods deal with collections.
5. String manipulations.
6. More ???.

Background Material:

C# note of Per Laursen from 1st semester

(<https://github.com/perl-easj/Teaching-materials/tree/master/CSharpProgramming/Notes>)

Chapters: Getting started, Prog1 and OOP1.

This Assignment: Brushup#1

Purpose

The purpose of this assignment is to be able to safely creating model classes.

Mission

You are to

- a. create simple model classes in a C#-library
- b. make use of these classes in a simple console-application

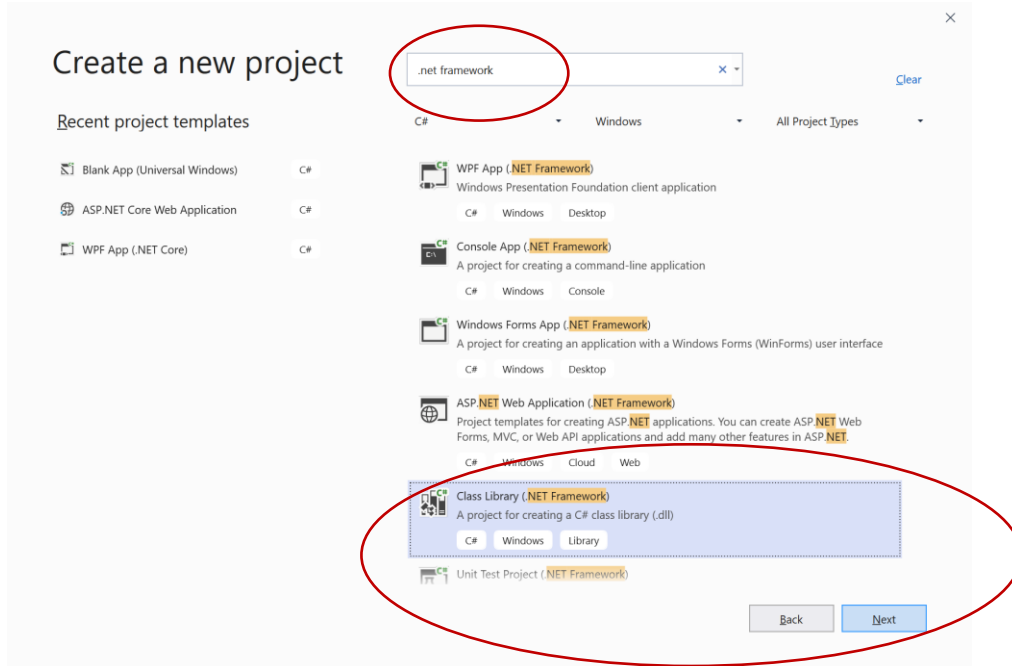
Domain description

In overall, you are to design and implement classes for a restaurant's menu card.

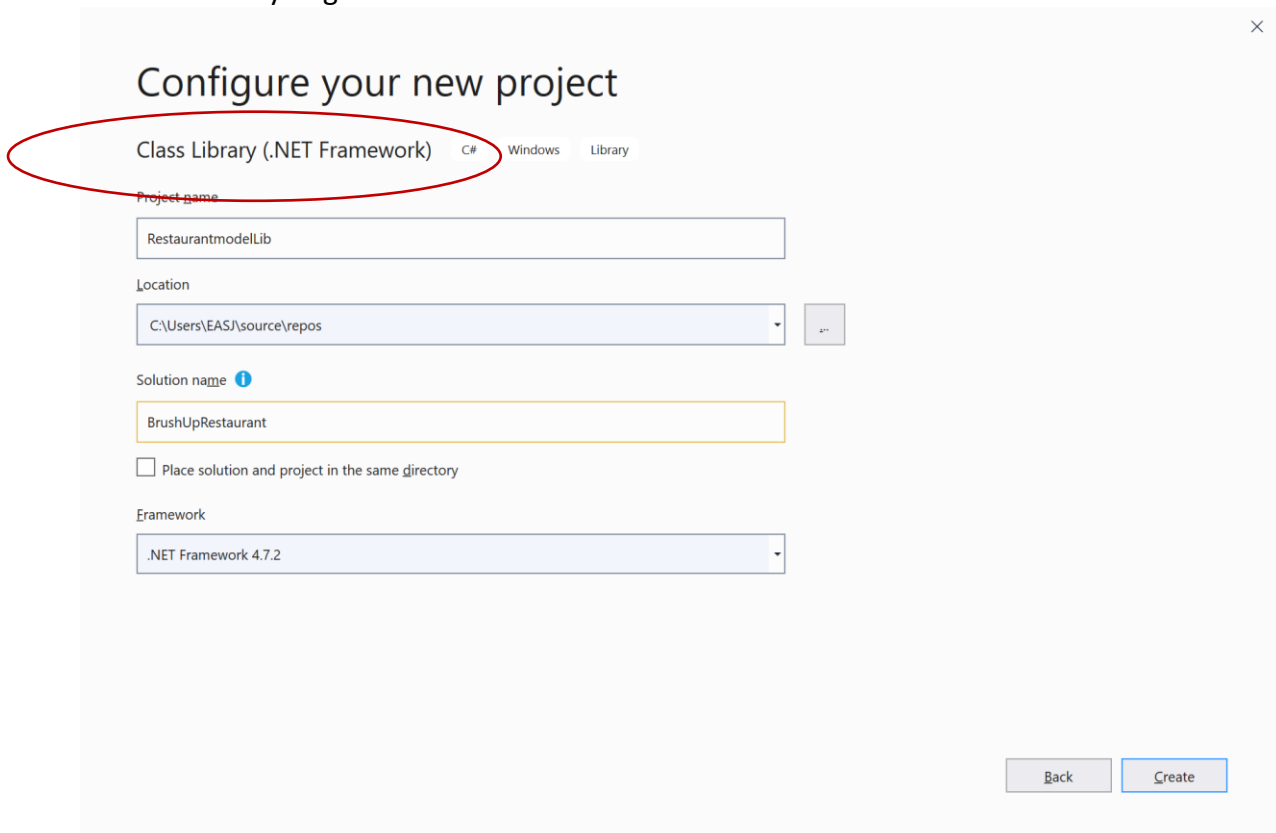
You are to create the classes drink, dish and menu card

Assignment 1: Create a C# Library

You are to open Visual Studio and create a new solution 'BrushUpRestaurant' with a project which is a **.Net Framework Library**.



Click next and you get:



Give the project the name 'RestaurantModellLib' and the solution 'BrushUpRestaurant'

Assignment 2: Create the classes

In the library create a folder named 'model' and delete/remove the default inserted 'Class1'.
In the model folder:

In the following implement, the code without any help from Visual Studio / Resharper i.e. do it all by hand ☺

Create a class '**Drink**'

- a. Make the class public
- b. Give the Drink class following information

Name, **TypeOfDrink**, both strings, **IsAlcoholic** a bool, **Price** a double

Make them as instance fields and as properties as get + set methods

Name the name of the drink

TypeOfDrink softdrink, redwine, beer etc.

IsAlcoholic if the drink have alcohol in

Price the cost of the drink

- c. Implement two constructors a default constructor and a constructor with values
- d. Implement a ToString method

Create a class '**Dish**'

- a. Make the class public
- b. Give the Dish class following information

Name, **TypeOfDish**, both strings, **Price** a double

Make them as instance fields and as properties as get + set methods

Name the name of the drink

TypeOfDish starter, main course, dessert etc.

Price the cost of the drink

- c. Implement two constructors a default constructor and a constructor with values
- d. Implement a ToString method

Create a class '**MenuCard**'

- a. Make the class public
- b. Give the MenuCard class following information

Drinks a list of Drink, **Dishes** a list of Dishes

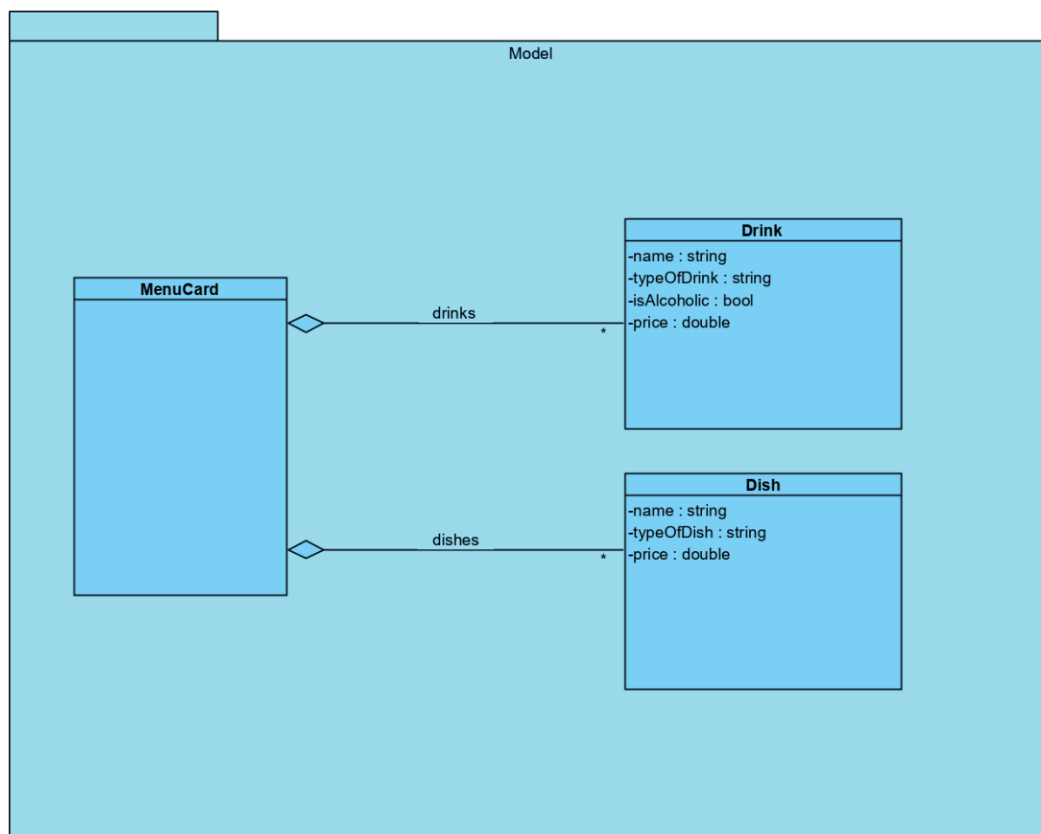
Make them as instance fields and as properties **but only as get methods**

Drinks all possible drinks

Dishes all possible dnameishes

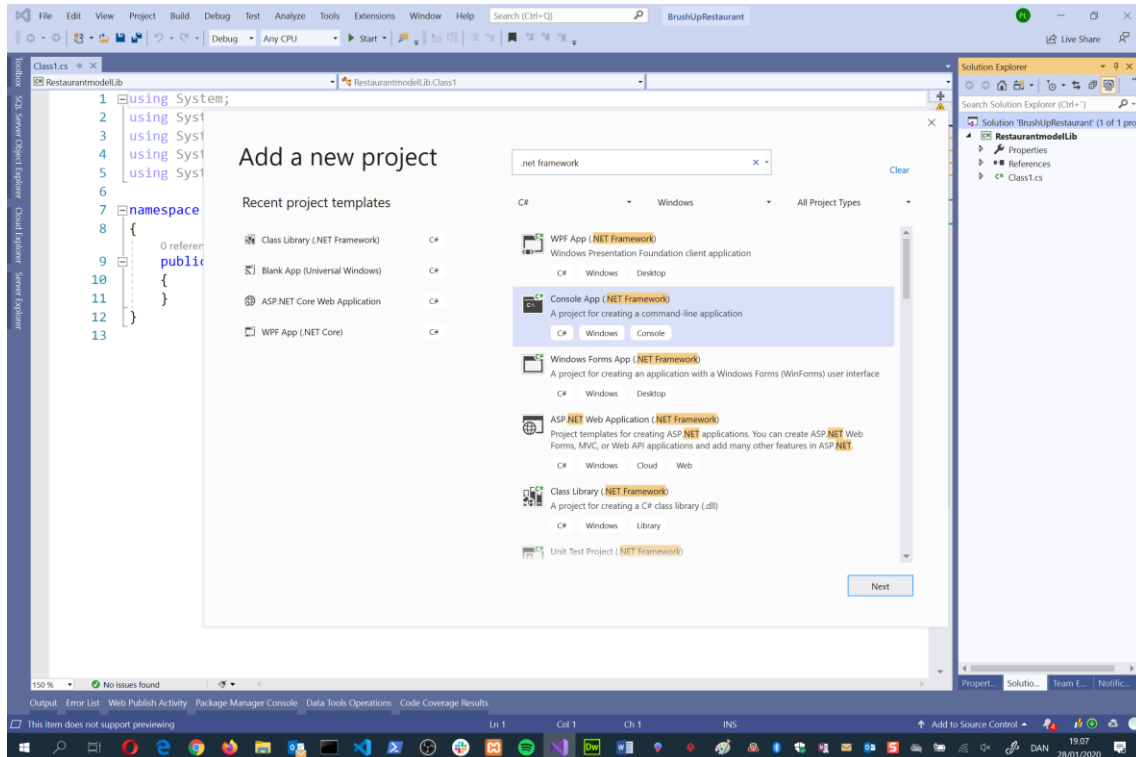
- c. Implement one constructors a default constructor
- d. Implement a ToString method

A class diagram of these classes:



Assignment 3: Create a Console Application

In the solution, create a new project 'BrushupConsoleApp' as a Console App (.Net Framework)



Add the library to your Console Application

Before you add the library to your project – build (or rebuild) the library, this create the dll-file we are going to use in this console application.

In the console application at the references (right click) add a reference to the RestaurantModelLib, whereby you now can access the three classes from the previous assignment.

Set up the console application

In the main method insert the lines:

```
BrushupWorker worker = new BrushupWorker();
worker.Start();
```

```
Console.ReadLine();
```

Create a class BrushupWorker with a Start-method like:

```
class BrushupWorker
{
    public void Start()
    {
        ...
    }
}
```

Instantiate objects

In the Start - method implement:

- a. Make two different objects / instances of the Drink class.
- b. For the first object
 - i. print the whole object i.e. use the ToString
 - ii. print only name and price
- c. For the second object
 - i. Print the whole object
 - ii. Set the price to be 10 % higher
 - iii. Print out the object again

- d. Make an object of the MenuCard-class
- e. Add the two Drink objects to the Drinks in MenuCard

- f. Make two different objects / instances of the Dish class.
- g. Add the two Dish objects to the Dishes in MenuCard
- h. Print out the menuCard object