

OPGAVEVEJLEDNING TIL UML 1 SPM. 5

1 Opgaven

Her starter vejledningen til spørgsmål 5 *Implement Design Class* i UML1 opgaven:

- A. *Create a new Console application PizzaStore*
- B. *Implement all the classes from your Design Class Diagram.*
 - a. *Every class should have properties and a constructor*
- C. *In the first version of the PizzaStore application the associations/aggregations/compositions should be implemented as 1:1 connections.*
- D. *All the classes should implement a ToString method*
- E. *Create a function, CalculateTotalPrice, that calculates the total price for the order. The function should return the total price for the order including tax and delivery costs (40 kr).*
- F. *To test your application you should create a class Store with a method Start.*
 - a. *Call the Start method from the main method in the class Program.*
 - b. *In the Start method you should:*
 - i. *Create 3 Pizza objects,*
 - ii. *3 Customer objects and*
 - iii. *Order objects each with a different pizza.*
- G. *Print out order information*
 - a. *Using the object reference to each Order object, you should print out the pizza name, the customer name and the total price for each order.*
 - b. *Test the method CalculateTotalPrice*

Bemærk: Det vil være en fordel, hvis du har klasserne Customer, Order, Pizza og Order med starten af UML1 opgaven i Software Design lektionerne

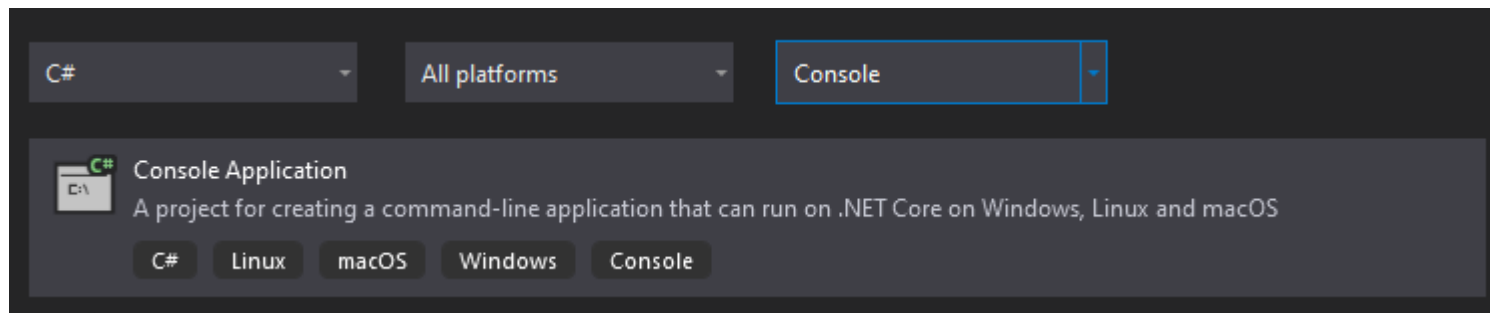
2 Løsningstrin

2.1 Opret GitHub konto

1. Gå til <https://github.com/>
2. Indtast e-mail adresse
3. Vælg Sign up for GitHub, hvis du ikke allerede oprettet en konto
4. Forbliv logget på

2.2 Opret projekt

1. Åben Visual Studio 2019
2. Vælg *File -> New -> Project*
3. Vælg "C#", "All platforms" og "Console" og vælg "Console Application" som vist herunder.



4. Klik på *Next*
5. Angiv PizzaStore som *Project name* og unchecked *Place solution and project in the same directory* (folder)

Configure your new project

Console Application C# Linux macOS Windows Console

Project name

PizzaStore

Location

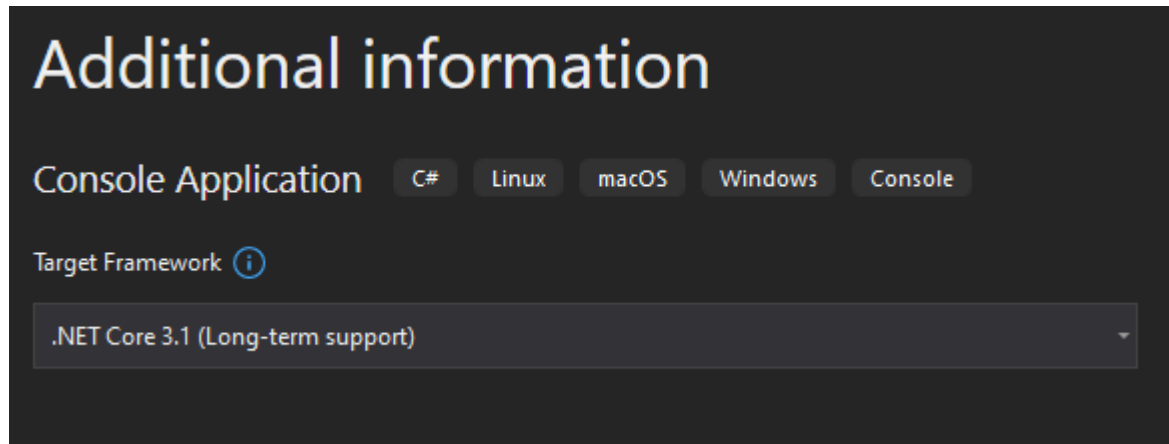
C:\Users\EASJ\source\repos

Solution name ⓘ

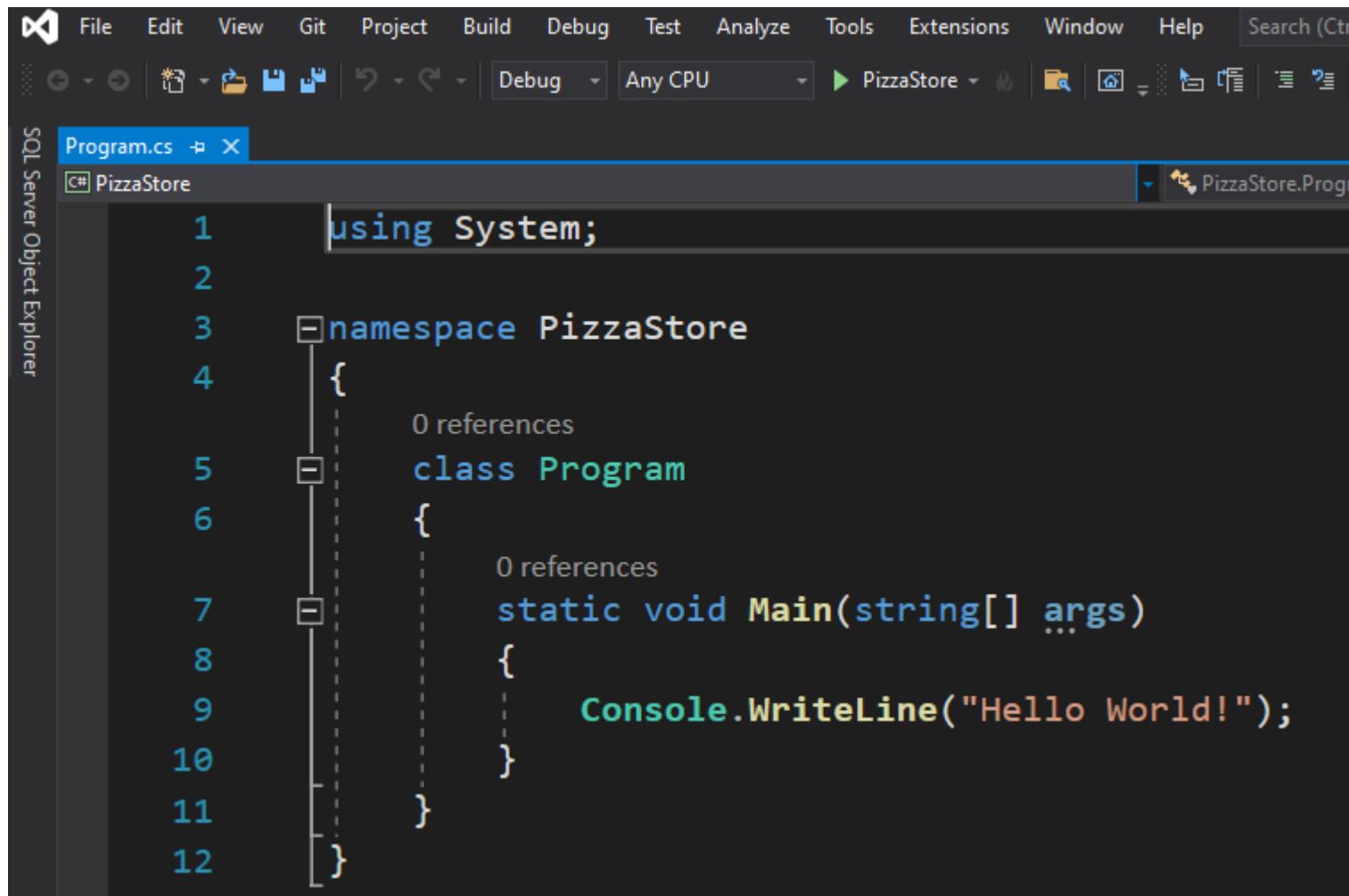
PizzaStore

☐ Place solution and project in the same directory

6. Klik på *Next*
7. Vælg *Target Framework* helst .NET Core 3.1 (Long-term support)



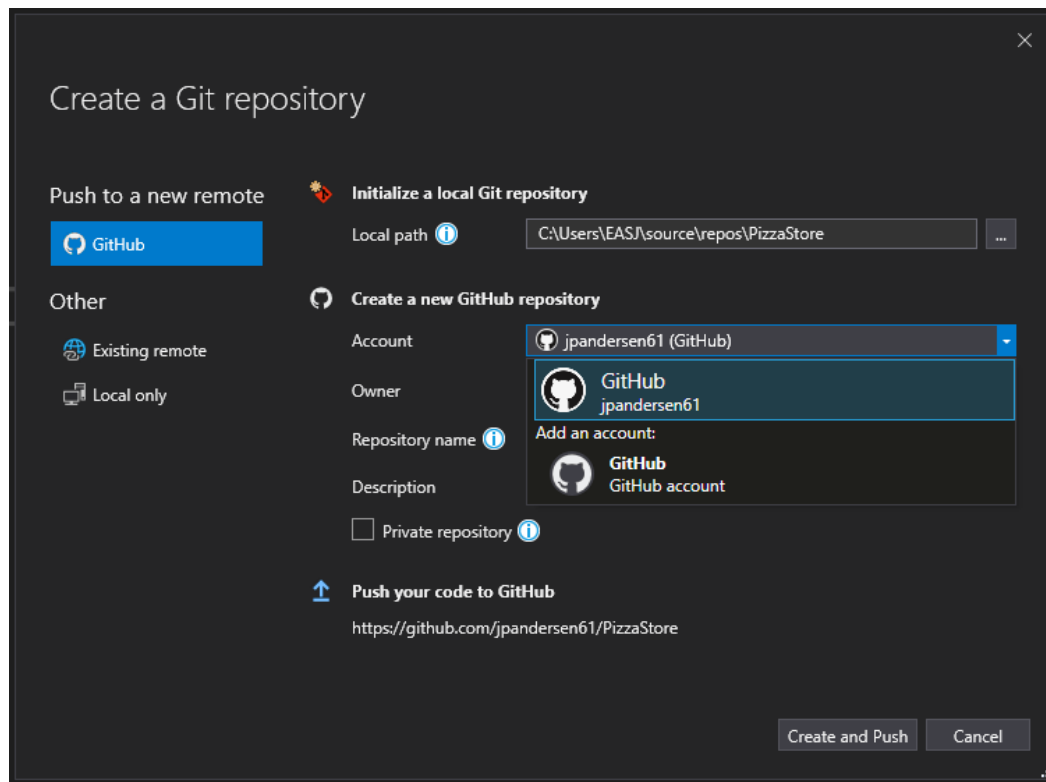
8. Klik på *Next*



```
1  using System;
2
3  namespace PizzaStore
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             Console.WriteLine("Hello World!");
12         }
13     }
14 }
```

2.3 Opret et GitHub repository til dit project

1. Vælg *Git* -> *Create Git Repository*
2. Vælg *Add an account*
3. "Autoriser" din nye konto i forhold til Visual Studio
4. Uncheck *Private repository*

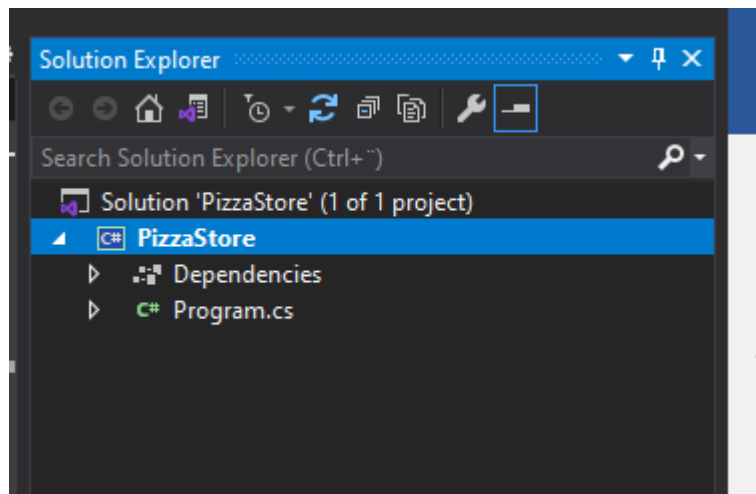


5. Klik *Create and Push*

6. Klik på *Save*
7. Observer ændringerne i GitHub

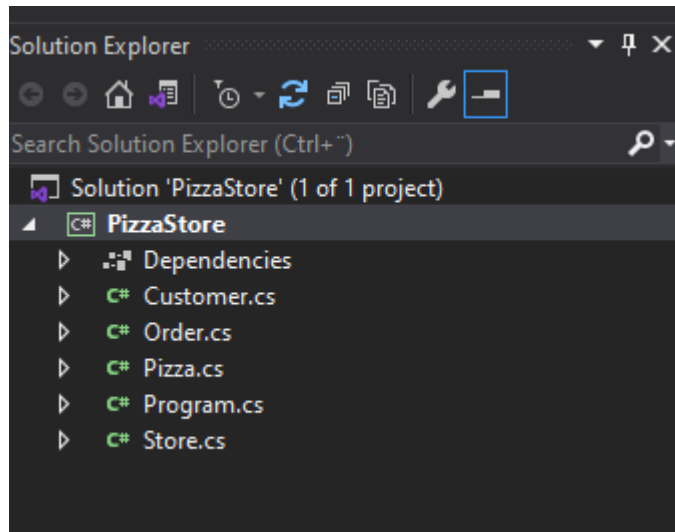
2.4 Opret klasser

1. Vælg *View -> Solution Explorer*



2. Højre-klik på "PizzaStore" projekt
3. Vælg *Add -> Class...*
4. Angiv "Store.cs" som *Name* for at oprette klassen "Store"
5. Klik på *Add*

6. Gentag oventående for klasserne "Customer", "Order" og "Pizza"



2.5 Implementer klasser

Gør følgende:

1. Indfør instansfelter (instance fields), properties og constructors på klasserne Store, Customer, Order og Pizza
2. Gør det samme for de øvrige klaser, som du ellers har defineret.

Bemærk: Dette som man gør i opgaverne OOP 1.3 og 1.4

Eksempel:


```
class Pizza
{
    string _name;
    int _price;

    1 reference
    public Pizza()
    {
        _name = "";
        _price = 0;
    }

    0 references
    public string Name
    {
        get { return _name; }
        set { _name = value; }
    }

    1 reference
    public int Price
    {
        get { return _price; }
        set { _price = value; }
    }
}
```

2.6 Indfør ToString metoder

Gør følgende:

1. Hver klasse skal have en ToString() metode som vist herunder:

```
0 references  
public override string ToString()  
{  
    return $"Name: {Name} - Price: {Price}";  
}
```

Name og *Price* er blot eksempler på properties, som anvendes i ToString() metoden. Eksemplet kunne være taget fra Pizza klassen.

2. Prøv at angive et objekt af klassen som parameter i Console.WriteLine() – f.eks. Console.WriteLine(<Pizza object>)

Hvad sker der?

Se også Power Point: [Link](#)

2.7 Hvordan er dine objekter og klasser organiseret i forhold til hinanden

Gør følgende:

1. Skab dig overblikket og find ud af hvordan Store, Order, Pizza og Customer skal være organiseret i forhold til hinanden.

Eksempel:

```
class Order
{
    double _taxPct;
    double _deliveryCosts;
    Pizza p;

    0 references
    public Order()
    {
        p = new Pizza();
        TaxPct = 0.0;
        DeliveryCosts = 0.0;
    }

    2 references
    double TaxPct
    {
        get { return _taxPct; }
        set { _taxPct = value; }
    }

    2 references
    double DeliveryCosts
    {
        get { return _deliveryCosts; }
        set { _deliveryCosts = value; }
    }

    0 references
    public double CalculateTotalPrice()...
}
```

2.8 Beregnings-og printmetoder

Færdigør metoderne

2.9 Push dine ændring til GitHub

1. Vælg *Git -> Commit or Stash*
2. Skriv en kommentar vedr. dine ændringer.
3. Observér ændringerne i GitHub

4. Vælg *Commit All and Push*

