

Въведение в MVC

Реализация на програма



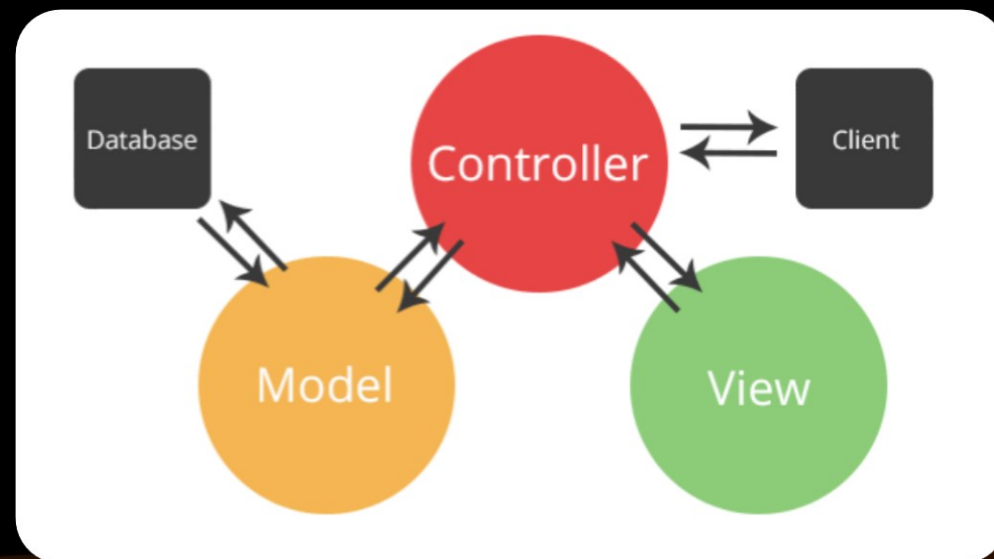
Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Разработка на софтуер



Съдържание

1. Program.cs
2. Model/Tip.cs
3. Views/Display.cs
4. Controllers/TipCalculatorController.cs



Класът Program

Ще използваме Program класът като входна точка. Тук ще създадем контролера, който ще управлява всичко. Възможно е да се наложи да добавите using ConsoleMVC.Controllers;

```
class Program
{
    static void Main(string[] args)
    {
        TipCalculatorController t =
            new TipCalculatorController();
    }
}
```

Модел

- Две полета за сума и процент
- Свойство за сумата
- Свойство за процент с валидация на процент
- Конструктор с 2 параметъра; конструктор по подразбиране без параметри
- Метод CalculateTip()
- Метод CalculateTotal()

Класът Tip

- Полета

```
private double amount;  
private double percent;
```

- конструктори по подразбиране

```
public Tip(double amount, double percent) {  
    Amount = amount;  
    Percent = percent;  
}
```

```
public Tip() : this(0, 0) { }
```

Верижно извикване на горния конструктор, подавайки му нули като параметри по подразбиране

Класът Tip(2)

- Свойство за сумата

```
public double Amount  
{  
    get { return amount; }  
    set { amount = value; }  
}
```

Класът Tip(3)

- Свойство за процента

```
public double Percent
{
    get { return percent; }
    set
    {
        if (value > 1)
        {
            value /= 100.0;
        }
        percent = value;
    }
}
```

Въведеният процент може да е цяло число или десетична дроб по-малка от 1. Ако е по-голямо от 1, се дели на 100, така най-голямата възможна стойност е 100%

Класът Tip(4)

- публични методи

```
public double CalculateTip()  
{  
    return Amount * Percent;  
}
```

```
public double CalculateTotal()  
{  
    return CalculateTip() + Amount;  
}
```



Общо за плащане

Изглед

- Клас Display.cs
- Полета / свойства за: процент, сума, бакшиш, обща сума
- Конструктор по подразбиране

Класът Display

- Свойства:

```
public double Percent { get; set; }
```

```
public double Amount { get; set; }
```

```
public double Total { get; set; }
```

```
public double TipAmount { get; set; }
```

Класът Display(2)

- Конструктор:

```
public Display()  
{  
    Percent = 0;  
    Amount = 0;  
    Total = 0;  
    TipAmount = 0;  
    GetValues();  
}
```

Класът Display(3)

- Метод за въвеждане

Частен метод за въвеждане на данни, извикван от конструктора

```
private void GetValues()  
{  
    Console.WriteLine("Enter the amount of the meal:");  
    Amount = double.Parse(Console.ReadLine());  
    Console.WriteLine("Enter the percent you want to tip: ");  
    Percent = double.Parse(Console.ReadLine());  
}
```

Класът Display(4)

- публични свойства

методът е публичен, така може да се извеждат данни с извикване от контролера

```
public void ShowTipandTotal()
{
    Console.WriteLine("Your tip is {0:C}", TipAmount);
    Console.WriteLine("The total will be {0:C}", Total);
}
```


Контролер

- Свързва изгледа и модела.
- Създава инстанция на изгледа
- Създава инстанция на модела
- Предава данни към изгледа
- Забележка: Тук трябва да добавите /имайте предвид, че името на проекта е ConsoleMVC/:
using ConsoleMVC.Model;
using ConsoleMVC.Views;

Класът TipCalculatorController

- полета

```
private Tip tip;  
private Display display;
```

Тук се създават полета за всички модели и изгледи, които желаем да се използват в рамките на контролера

Класът TipCalculatorController (2)

Конструктор по подразбиране

```
public TipCalculatorController()
{
    display = new Display();
    tip = new Tip(display.Amt, display.Percentage);
    display.TipAmount = tip.CalculateTip();
    display.Total = tip.CalculateTotal();
    display.ShowTipandTotal();
}
```

Какво научихме в този час?

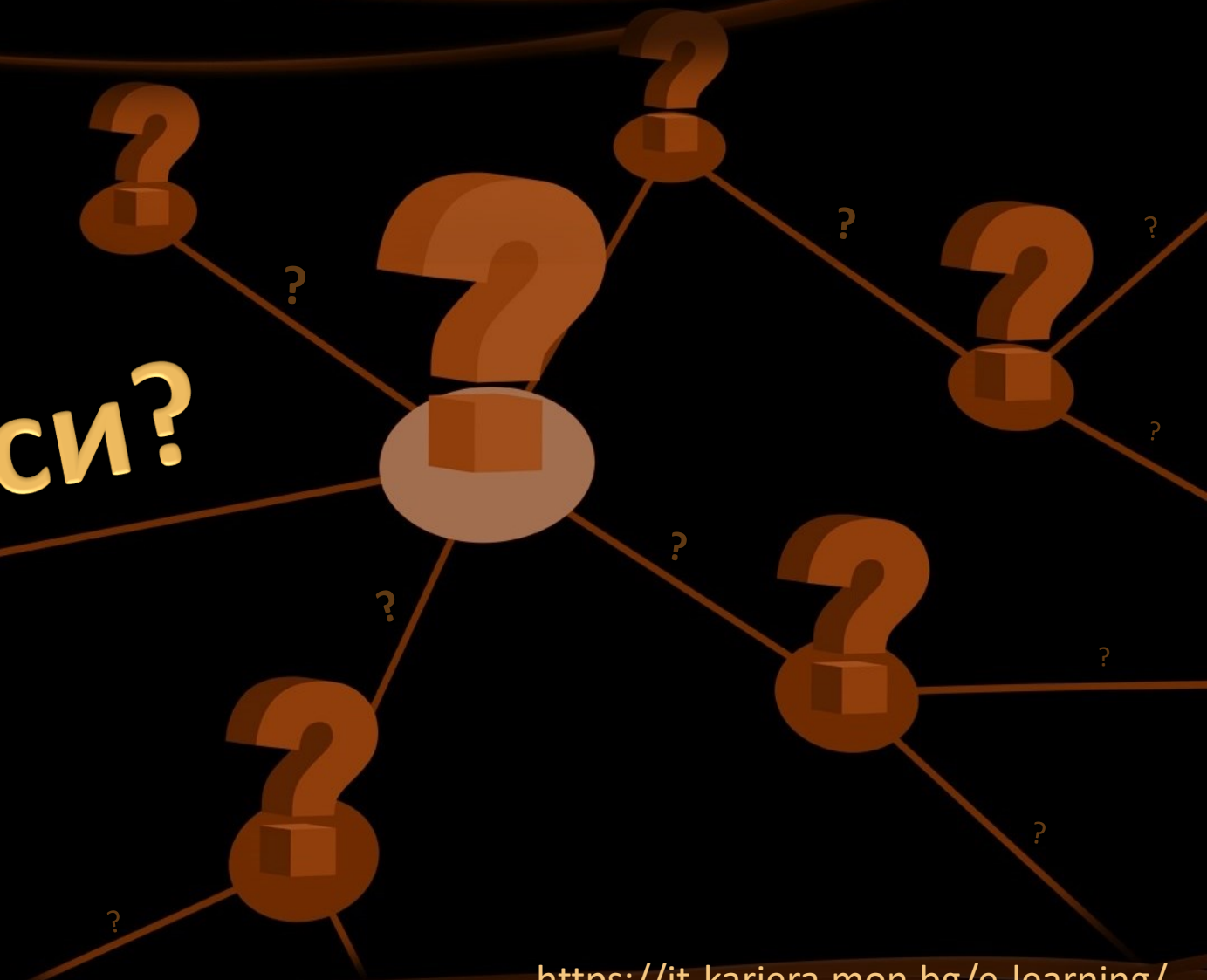
- Структурата на MVC проекта е добре да е разпределена в 3 отделни подпапки:
 - Controllers – съдържаща контролер класовете
 - Model – съдържаща класовете за моделите
 - Views – съдържаща изгледите
- Разгледахме примерен код за мини MVC приложение



Въведение в MVC. Реализиране на MVC програма



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

