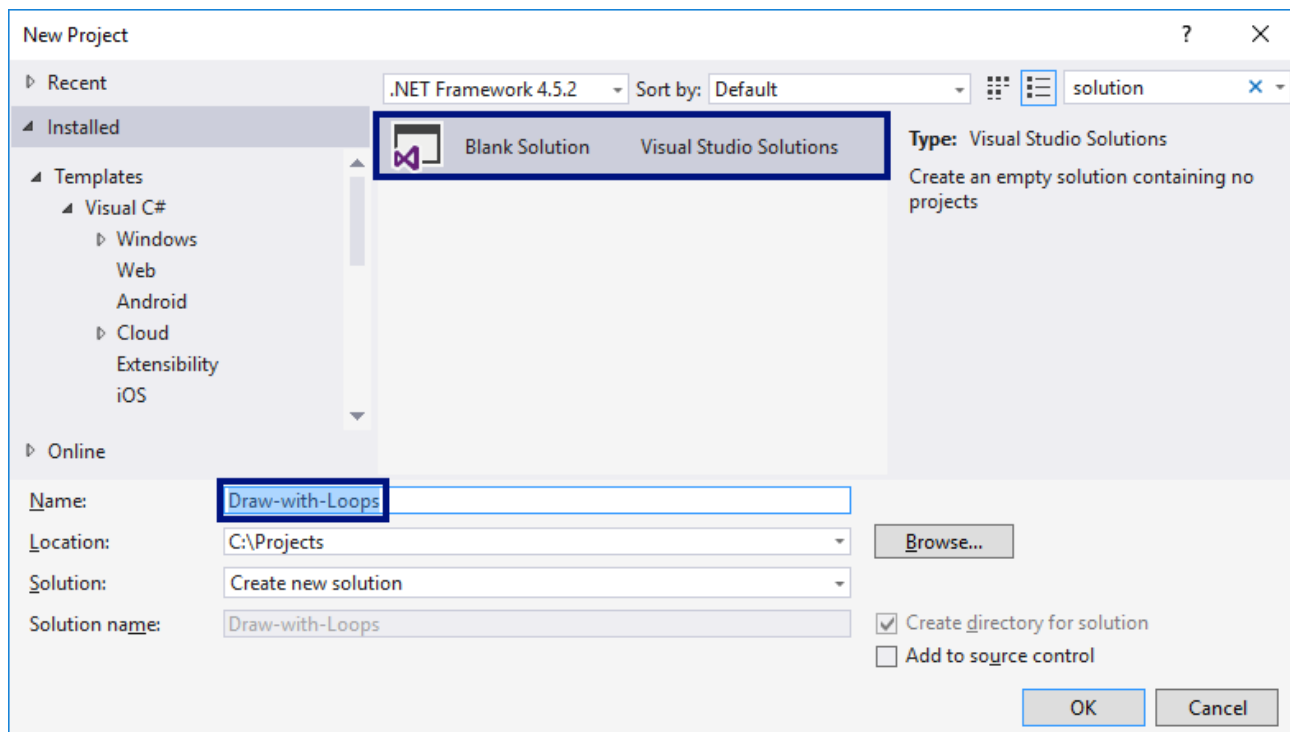


# Упражнения: Чертане с цикли

Задачи за упражнение в клас и за домашно към курса „[Основи на програмирането](#)“ @ СофтУни.

## 0. Празно Visual Studio решение (Blank Solution)

1. Създайте празно решение (**Blank Solution**) във Visual Studio за да организирате кода от задачите за упражнение. Целта на този **blank solution** е да съдържа **по един проект за всяка задача** от упражненията.



2. Задайте **да се стартира по подразбиране текущия проект** (не първият в решението). Кликнете с десен бутон на мишката върху **Solution 'Draw-with-Loops' → [Set StartUp Projects...] → [Current selection]**.

## 1. Правоъгълник от 10 x 10 звездички

Напишете програма, която чертае на конзолата **правоъгълник от 10 x 10 звездички**:

вход	изход
(няма)	***** ***** ***** ***** ***** ***** ***** ***** ***** *****

Подсказки:

1. Създайте **нов проект** в съществуващото Visual Studio решение – конзолна C# програма. Задайте подходящо име на проекта, например **“Rectangle-of-10x10-Stars”**.

- Отидете в тялото на метода **Main(string[] args)** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:

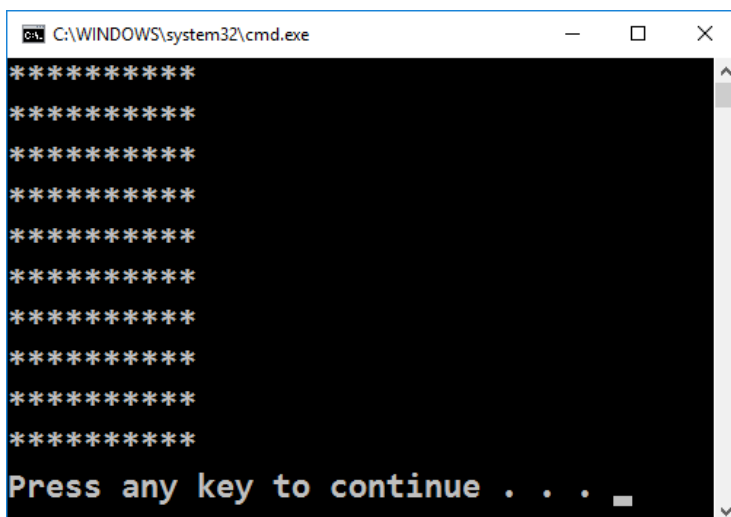
```

using System;

class Program
{
    static void Main(string[] args)
    {
        for (var i = 1; i <= 10; i++)
        {
            Console.WriteLine(new string('*', 10));
        }
    }
}

```

- Стартирайте програмата с [Ctrl+F5] и я **тествайте**:



- Тествайте решението си в **judge** системата: <https://judge.softuni.bg/Contests/Practice/Index/155#0>. Трябва да получите **100 точки** (напълно коректно решение).

## 2. Правоъгълник от N x N звездички

Напишете програма, която въвежда цяло положително число **n** и печата на конзолата **правоъгълник от n \* n звездички**. Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	** **	3	*** *** ***	4	**** **** **** ****

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#1>.

**Подсказка:** отпечатайте **n** звездички в цикъл **n** пъти, точно както в предната задача.

### 3. Квадрат от звездички

Напишете програма, която чете от конзолата число **n** и чертае **квадрат от n \* n** звездички. Разликата с предходната задача е, че между всеки две звездички има по един интервал. Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	* * * *	3	* * * * * * * * *	4	* * * * * * * * * * * * * * * *

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#2>.

**Подсказка:** завъртете два вложени цикъла. Може да си помогнете с кода по-долу (част от кода е нарочно замъглен, за да не преписвате механично, а да се замислите малко):

```
for (var r = 1; r <= n; r++)  
{  
    Console.Write("*");  
    for (var c = 1; c < n; c++)  
    {  
        Console.Write(" ");  
    }  
    Console.WriteLine();  
}
```

### 4. Триъгълник от долари

Да се напише програма, която въвежда число **n** и печата **триъгълник от долари** като в примерите:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	\$ \$ \$	3	\$ \$ \$ \$ \$ \$	4	\$ \$ \$ \$ \$ \$ \$ \$ \$ \$	5	\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#3>.

**Подсказка:** завъртете два вложени цикъла: за първия **row = 1 ... n**; за втория **col = 1 ... row**.

### 5. Квадратна рамка

Напишете програма, която въвежда цяло положително число **n** и чертае на конзолата **квадратна рамка** с размер **n \* n** като в примерите по-долу:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
3	+ - +   -   + - +	4	+ - - +   - -     - -   + - - +	5	+ - - - +   - - -     - - -     - - -   + - - - +	6	+ - - - - +   - - - -     - - - -     - - - -     - - - -   + - - - - +



## 8. Слънчеви очила

Напишете програма, която въвежда цяло число  $n$  ( $3 \leq n \leq 100$ ) и печата **слънчеви очила** с размер  $5*n$  х  $n$  като в примерите:

вход	изход
3	<pre> *****  ***** */////*    */////* *****  ***** </pre>
4	<pre> *****  ***** *////////*     *////////* *////////* *////////* *****  ***** </pre>
5	<pre> *****  ***** */////////* */////////* *////////*      *////////* *////////* *////////* *****  ***** </pre>

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#7>.

Подсказки:

- Отпечатайте **най-горния ред** от очилата:
  - $2*n$  звездички;  $n$  интервала;  $2*n$  звездички
- Отпечатайте **средните  $n-2$  реда**:
  - звездичка;  $2*n-2$  наклонени черти; звездичка;  $n$  интервала; звездичка;  $2*n-2$  наклонени черти; звездичка
  - когато редът е  $(n-1) / 2 - 1$ , печатайте  $n$  вертикални черти вместо  $n$  интервала
- Отпечатайте **най-долния ред** от очилата:
  - $2*n$  звездички;  $n$  интервала;  $2*n$  звездички

## 9. Къщичка

Напишете програма, която въвежда число  $n$  ( $2 \leq n \leq 100$ ) и печата **къщичка** с размер  $n$  х  $n$ :

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
2	<pre> **    </pre>	3	<pre> _ *_ ***  *  </pre>	4	<pre> _ ** _ ****  **   **  </pre>	5	<pre> _ _ * _ _ _ *** _ *****  ***   ***  </pre>	6	<pre> _ _ * _ _ _ **** _ *****  ****   ****   ****  </pre>

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#8>.

Подсказки:

- Отпечатайте в цикъл **покрива** на къщичката:
  - Той съдържа  $(n + 1) / 2$  реда.
  - На първия си ред съдържа **1** звездичка при нечетно  $n$  или **2** звездички при четно  $n$ .
  - На всеки следващ ред съдържа с **2** звездички повече.
- Отпечатайте в цикъл **основата** на къщичката:  $n / 2 - 1$  реда.

## 10. \* Диамант

Напишете програма, която въвежда цяло число  $n$  ( $1 \leq n \leq 100$ ) и печата диамант с размер  $n$  като в примерите по-долу:

вход	изход
1	*

вход	изход
2	**

вход	изход
3	- *- * _* - *_

вход	изход
4	- *** - * _ _* - *** -

вход	изход
5	- - * - - _ * _* * _ _ _* _ * _* _ - - * _ -

вход	изход
6	- - *** - _ * _ _* _ * _ _ _* _ * _ _* _ - - *** -

вход	изход
7	- - - * - - - - * _* - - * _ _* _ * _ _ _* _ * _ _* _ - - * _* _ - - - * - -

вход	изход
8	- - - *** - - - - * _* _* - - * _ _ _* _ * _ _ _ _* _ * _ _ _* _ - - * _* _* - - - - ** - -

вход	изход
9	- - - - * - - - - - - * _* _ - - - * _ _* _ - _ * _ _ _* _ - * _ _ _ _* _ - * _ _ _* _ - - - * _ _* _ - - - - * _* _ - - - - - * - - -

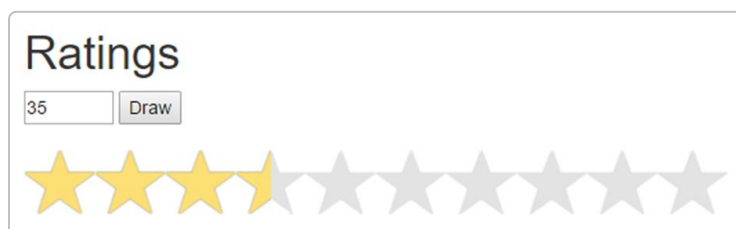
Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/155#9>.

Подсказки:

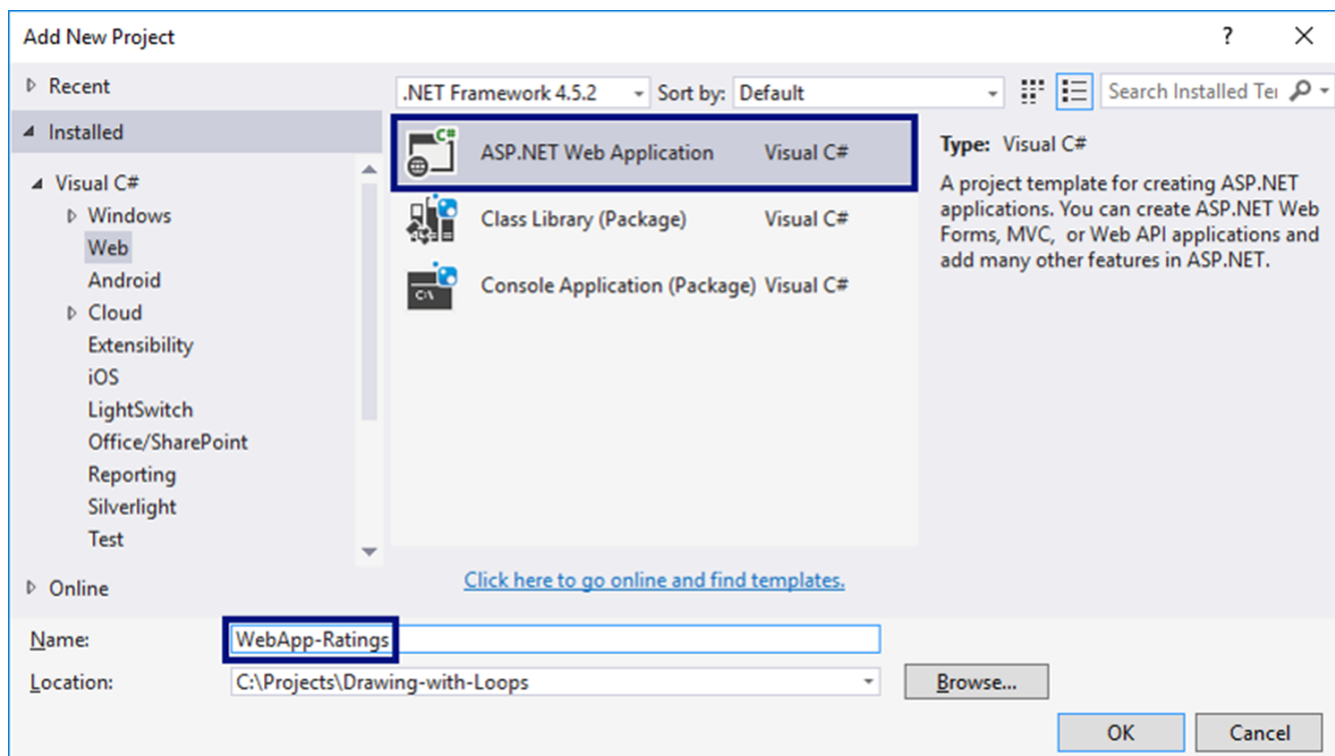
- Всички редове съдържат точно по  $n$  символа.
- Първият ред съдържа **отляво и отдясно** точно  $\text{leftRight} = (n - 1) / 2$  тирета.
- Всеки следващ ред **до средния** съдържа отляво и отдясно с **1 тире по-малко** от предходния.
- Всеки следващ ред **след средния** съдържа отляво и отдясно с **1 тире повече** от предходния.
- Всеки ред съдържа **в средата** си (във вътрешността на диаманта)  $\text{mid} = n - 2 * \text{leftRight} - 2$  тирета.
- Всеки ред съдържа **2 звездички**, освен когато  $\text{mid}$  е отрицателно (тогава има само 1 звездичка).
- За всеки ред може да се изчислят и отпечатаат неговите 5 съставни части:
  - $\text{leftRight}$  тиренца отляво
  - 1 звездичка
  - $\text{mid}$  тиренца в средата (когато  $\text{mid} \geq 0$ )
  - 1 звездичка (когато  $\text{mid} \geq 0$ )
  - 1 звездичка
  - $\text{leftRight}$  тиренца отляво

## 11. Рейтинги – визуализация в уеб среда

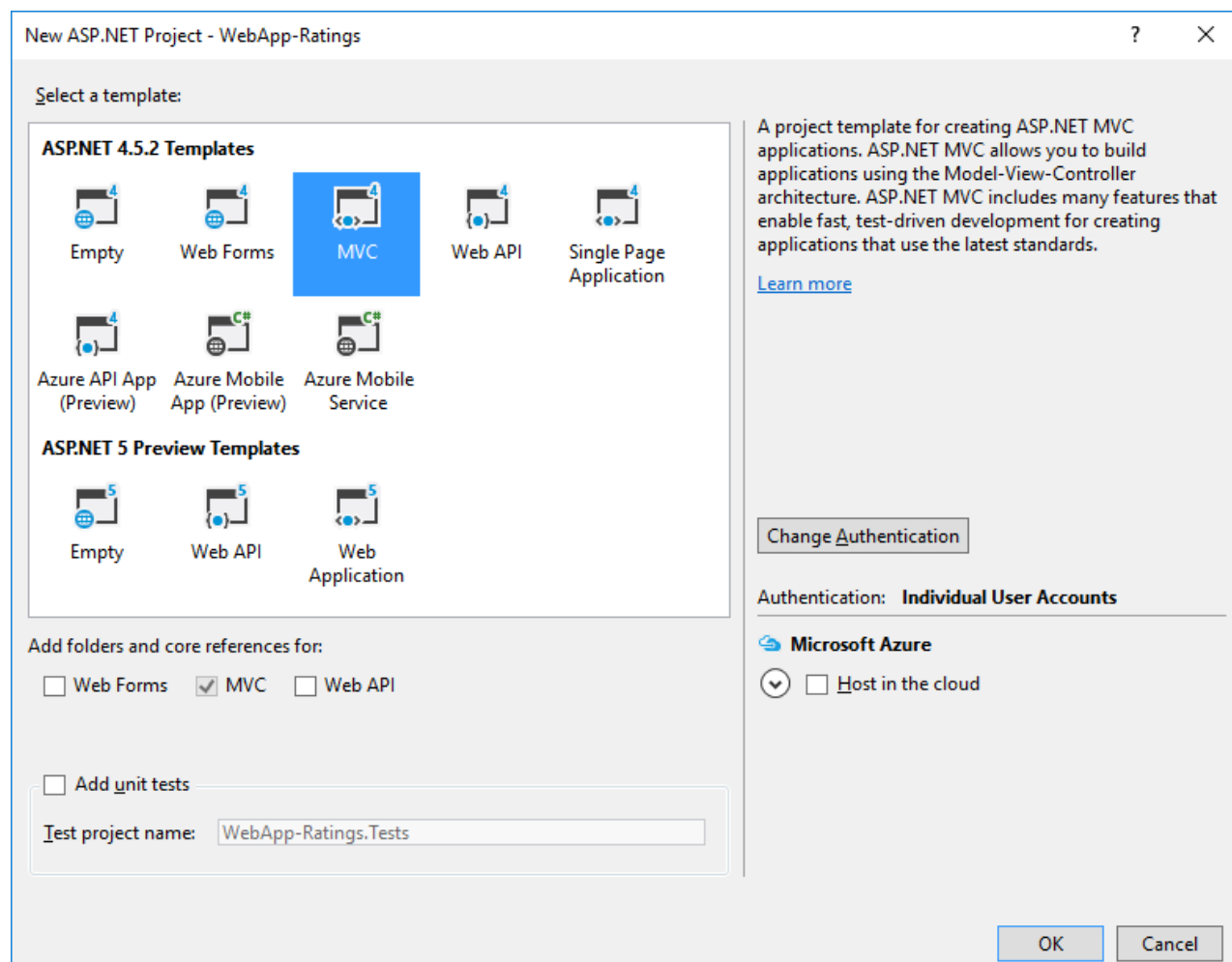
Да се разработи ASP.NET MVC уеб приложение за визуализация на рейтинг (число от 0 до 100). Чертаят се от 1 до 10 звездички (с половинки). Звездичките да се генерират с **for**-цикъл:



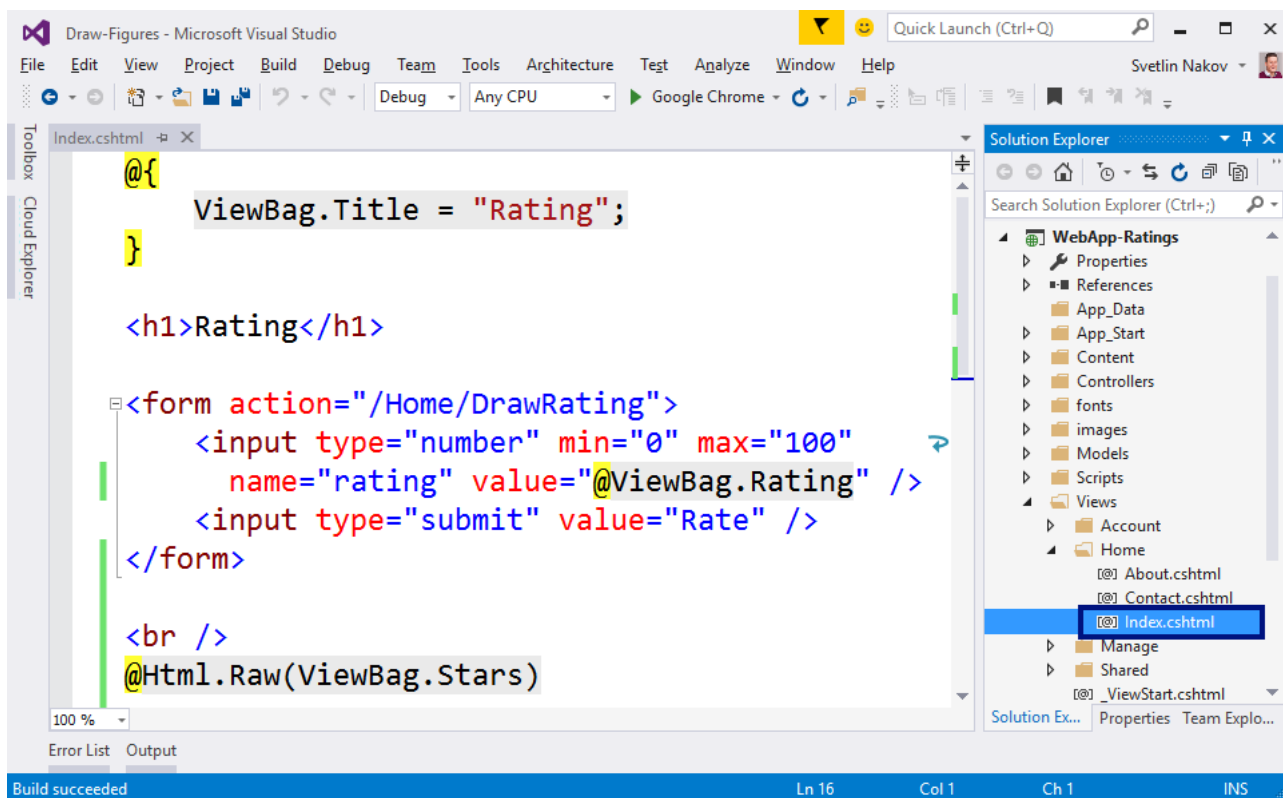
1. Във Visual Studio създайте **ново ASP.NET MVC уеб приложение** с език C#. Добавете нов проект от [Solution Explorer] → [Add] → [New Project...]. Дайте смислено име, например **“WebApp-Ratings”**:



Изберете тип на уеб приложението **“MVC”**:



2. Редактирайте файла **Views/Home/Index.cshtml**. Изтрийте всичко и въведете кода от картинката:



Този код създава уеб форма **<form>** с едно поле **"rating"** за въвеждане на число в интервала [0...100] и бутон **[Draw]** за изпращане на данните от формата към сървъра. Действието, което ще обработи данните, се казва **"/Home/DrawRatings"**, което означава метод **"DrawRatings"** в контролер **"Home"**, който се намира във файла **"HomeController.cs"**. След формата се отпечатва съдържанието на **ViewBag.Stars**. Кодът, който ще се съдържа в него, ще бъде динамично генериран от контролера **HTML** с поредица от звездички.

3. Добавете метод **"DrawRatings"** в контролера **"HomeController"**. Отворете файла **Controllers/HomeController.cs** и добавете следния код в тялото на класа **HomeController**:

```
public ActionResult DrawRating(int rating)
{
    var fullStars = rating * 10 / 100;
    var emptyStars = (100 - rating) * 10 / 100;
    var halfStars = 10 - fullStars - emptyStars;

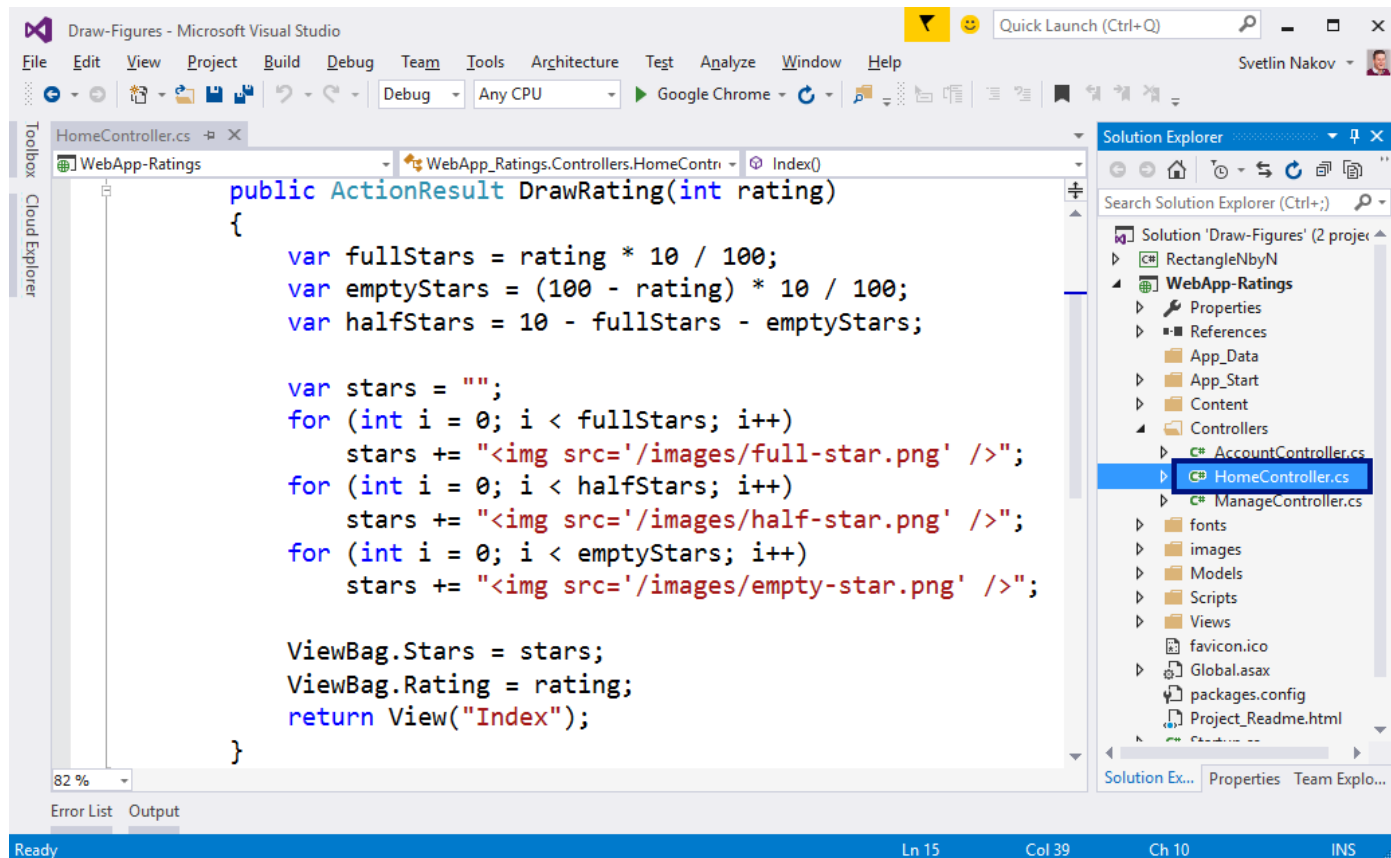
    var stars = "";
    for (int i = 0; i < fullStars; i++)
        stars += "<img src='/images/full-star.png' />";
    for (int i = 0; i < halfStars; i++)
        stars += "<img src='/images/half-star.png' />";
    for (int i = 0; i < emptyStars; i++)
        stars += "<img src='/images/empty-star.png' />";

    ViewBag.Stars = stars;
    ViewBag.Rating = rating;
    return View("Index");
}
```

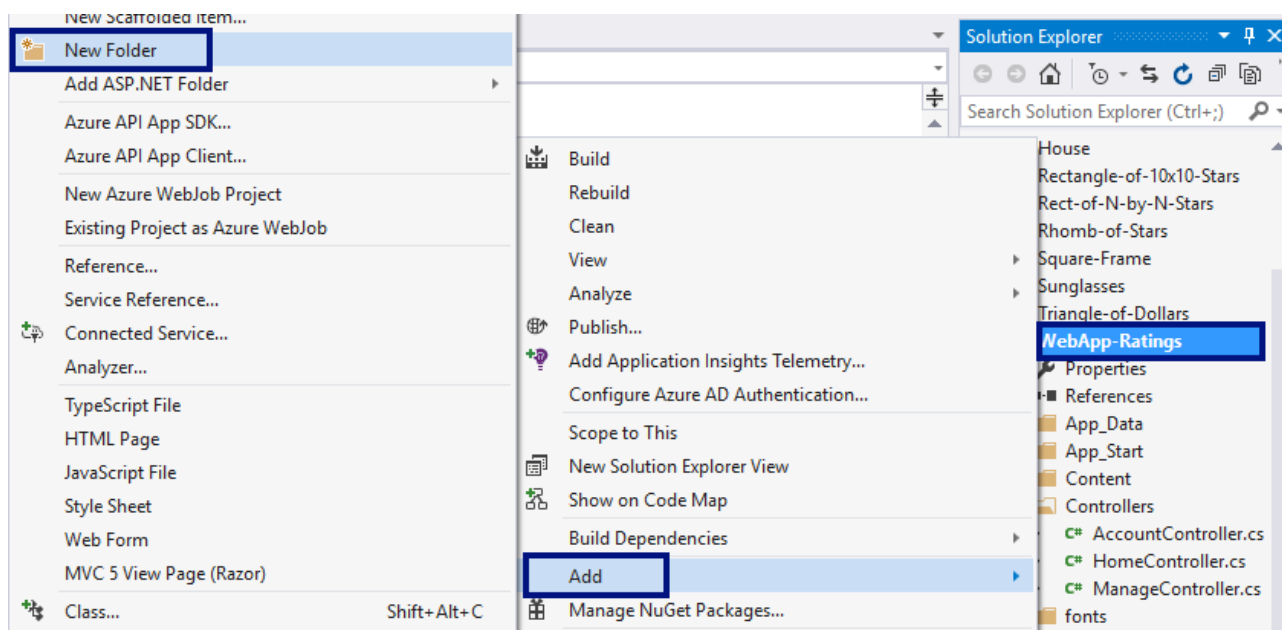


Горният код взема въведеното число **rating**, прави малко пресмятания и изчислява броя **пълни звездички**, броя **празни звездички** и броя **половинки звездички**, след което генерира **HTML код**, който нарежда няколко картинки със звездички една след друга, за да сглоби от тях картинката с рейтинга. Подготвеният HTML код се записва във **ViewBag.Stars** за визуализация от изгледа **Index.cshtml**. Допълнително се запазва и изпратеният рейтинг (като число) във **ViewBag.Rating**, за да се зададе в полето за рейтинг в изгледа.

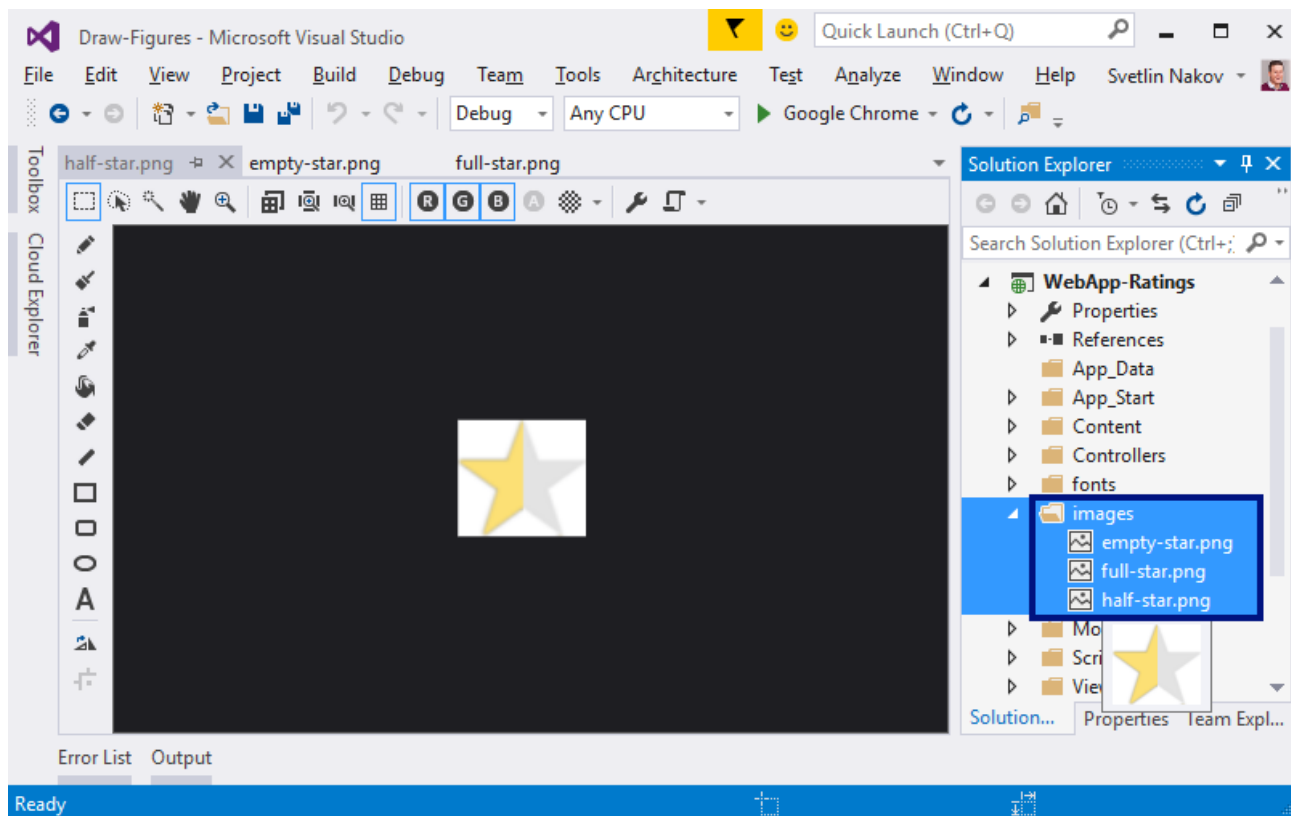
За да се ориентирате по-добре в проекта, може да си помогнете с картинката от Visual Studio по-долу:



4. От [Solution Explorer] направете папка **“images”** в проекта:



Сега добавете **картинките** със звездичките (те са част от заданието за домашно). Копирайте ги от Windows Explorer и ги поставете в папката **“images”** в [Solution Explorer] във Visual Studio с **copy / paste**.



5. Стартирайте проекта с **[Ctrl+F5]** и му се порадвайте:

