

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Рязанский государственный радиотехнический  
университет имени В.Ф. Уткина»  
Рязанский станкостроительный колледж

Отчёт о практической работе №2  
Создание проекта и загрузка в репозиторий  
МДК  
«02. 02»

Выполнила:  
Студент группы ИСП-22  
Шерстнёва Ю.С.  
Проверил:  
Родин Е.Н.

Рязань 2025

## Основная часть

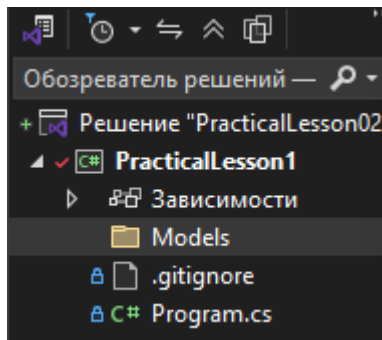
**Цель работы:** Инициализировать новый консольный проект на C# в Visual Studio, настроить его для работы с GitHub и загрузить первоначальную версию в репозиторий.

### Ход выполнения работы:

Задачи:

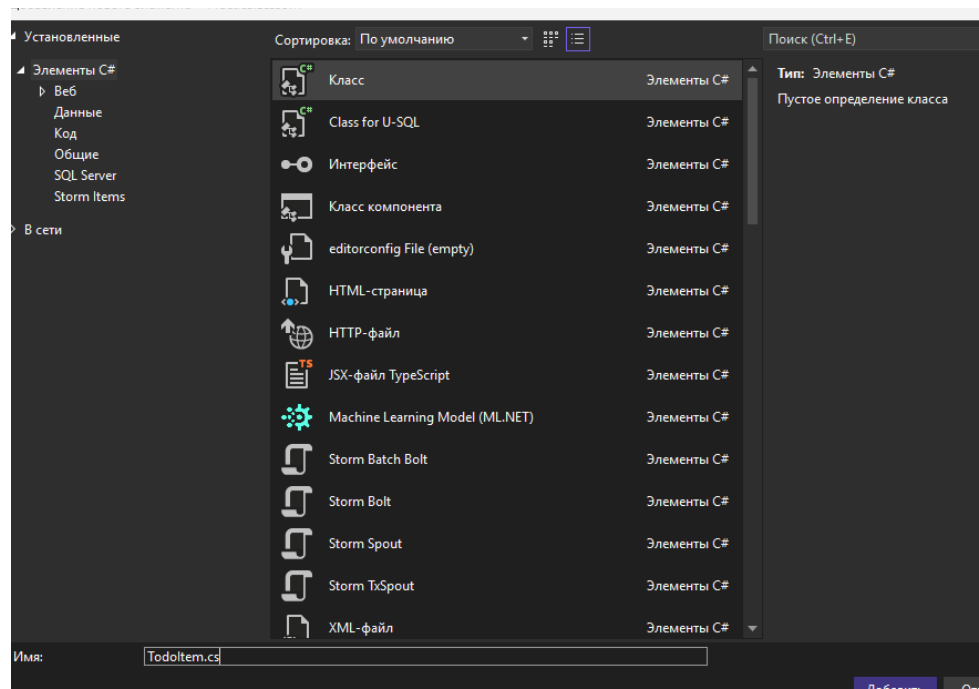
#### 1. Добавление модели задачи:

В вашем консольном проекте создайте папку Models



В папке Models создайте класс TodoItem.cs со следующими свойствами:

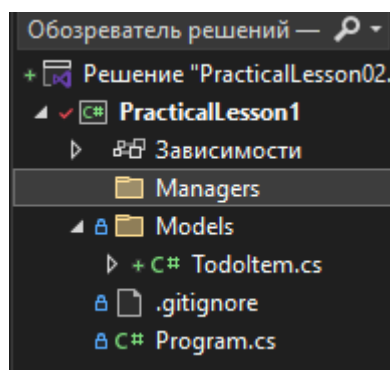
- Id (int),
- Description (string),
- IsCompleted (bool).



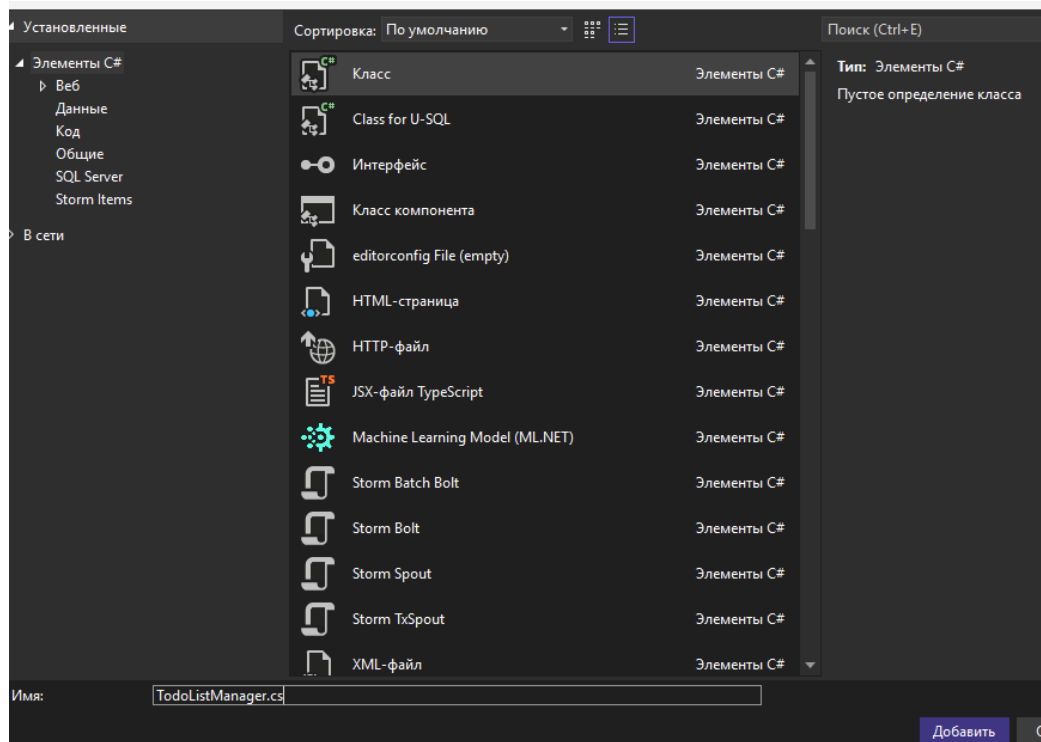
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace PracticalLesson1.Models
8 {
9     // Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
10    internal class TodoItem
11    {
12        // Свойства класса
13        // Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
14        public int Id { get; set; }
15        // Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
16        public string Description { get; set; }
17        // Ссылка: 2 | 0 изменений | 0 авторов, 0 изменений
18        public bool IsCompleted { get; set; }
19
20        // Конструктор для удобного создания задач
21        // Ссылка: 0 | 0 изменений | 0 авторов, 0 изменений
22        public TodoItem(int id, string description)
23        {
24            Id = id;
25            Description = description;
26            IsCompleted = false;
27        }
28
29        // Метод для отображения статуса задачи
30        // Ссылка: 0 | 0 изменений | 0 авторов, 0 изменений
31        public string GetStatusDisplay()
32        {
33            return IsCompleted ? "[x]" : "[ ]";
34        }
35    }
36 }
```

## 2. Создание списка и функции его отображения:

В вашем консольном проекте создайте папку Managers (или Services, Logic – на ваш вкус).



В этой папке создайте новый класс TodoListManager.cs.



Создайте список List todoList.

Добавьте несколько тестовых задач в список.

Реализуйте простую функцию, которая будет выводить содержимое списка дел в консоль, указывая номер задачи, ее описание и статус (например, [ ] для невыполненных и [x] для выполненных).

```

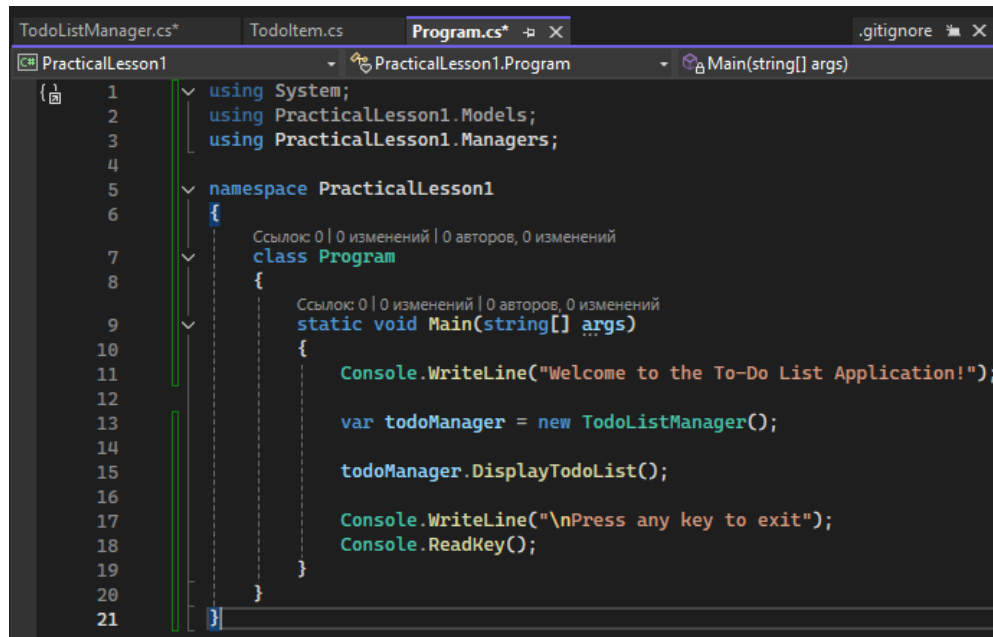
doListManager.cs*  x  Todoltem.cs  PracticalLesson1.csproj*  Program.cs
PracticalLesson1  PracticalLesson1.Managers.ToDoListManager  _practicalLesson1

1  using PracticalLesson1.Models;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace PracticalLesson1.Managers
9  {
10     internal class ToDoListManager
11     {
12         private List<ToDoItem> _practicalLesson1 = new List<ToDoItem>();
13
14         public ToDoListManager()
15         {
16             _practicalLesson1.Add(new ToDoItem(1, "Buy groceries"));
17             _practicalLesson1.Add(new ToDoItem(2, "Read a book"));
18             _practicalLesson1.Add(new ToDoItem(3, "Go for a walk"));
19         }
20         public void DisplayToDoList()
21         {
22             Console.WriteLine("\n--- Your To-Do List ---");
23             if (_practicalLesson1.Count == 0)
24             {
25                 Console.WriteLine("Your list is empty!");
26             }
27             else
28             {
29                 foreach (var item in _practicalLesson1)
30                 {
31                     Console.WriteLine($"{item.Id}. {item.GetStatusDisplay()} {item.Description}");
32                 }
33             }
34             Console.WriteLine("-----");
35         }
36     }
37 }

```

### 3. Внесение изменений в код:

Добавьте файлы `TodoItem.cs` и код для управления списком в `Program.cs`.

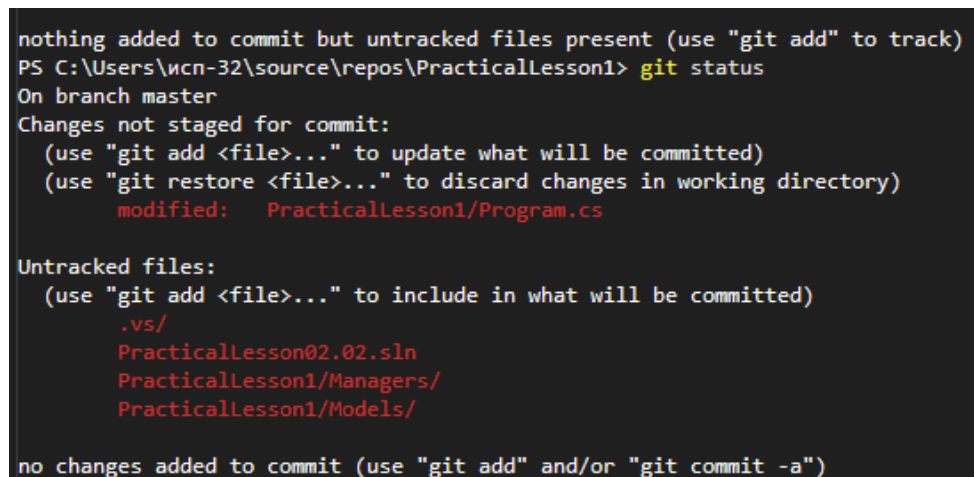


```
1 using System;
2 using PracticalLesson1.Models;
3 using PracticalLesson1.Managers;
4
5 namespace PracticalLesson1
6 {
7     class Program
8     {
9         static void Main(string[] args)
10        {
11            Console.WriteLine("Welcome to the To-Do List Application!");
12
13            var todoManager = new TodoListManager();
14
15            todoManager.DisplayTodoList();
16
17            Console.WriteLine("\nPress any key to exit");
18            Console.ReadKey();
19        }
20    }
21 }
```

### 4. Проверка статуса изменений:

В терминале Visual Studio выполните: `git status`

Проанализируйте, какие файлы были изменены. Вы должны увидеть новые файлы и измененный файл (`Program.cs`).



```
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\исп-32\source\repos\PracticalLesson1> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   PracticalLesson1/Program.cs

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .vs/
        PracticalLesson02.02.sln
        PracticalLesson1/Managers/
        PracticalLesson1/Models/

no changes added to commit (use "git add" and/or "git commit -a")
```

### 5. Индексация изменений:

Добавьте измененные файлы в индекс.

Переместитесь в папку с проектом с помощью команды: `cd PracticalLesson1`

Затем добавьте измененные файлы: `git add .`

```

PS C:\Users\icn-32\source\repos\PracticalLesson1> cd PracticalLesson1
PS C:\Users\icn-32\source\repos\PracticalLesson1\PracticalLesson1> git add .
PS C:\Users\icn-32\source\repos\PracticalLesson1\PracticalLesson1> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Managers/ToDoListManager.cs
        new file:   Models/ToDoItem.cs
        modified:   Program.cs

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../.vs/
        ../PracticalLesson02.02.sln

```

## 6. Фиксация изменений:

Сделайте коммит с описательным сообщением:

`git commit -m "Add ToDoItem model, ToDoListManager and basic display functionality"`

```

PS C:\Users\icn-32\source\repos\PracticalLesson1\PracticalLesson1> git commit -m "Add ToDoItem model,
1, ToDoListManager and basic display functionality"
[master 9849559] Add ToDoItem model, ToDoListManager and basic display functionality
3 files changed, 89 insertions(+), 1 deletion(-)
create mode 100644 PracticalLesson1/Managers/ToDoListManager.cs
create mode 100644 PracticalLesson1/Models/ToDoItem.cs

```

## 7. Загрузка изменений:

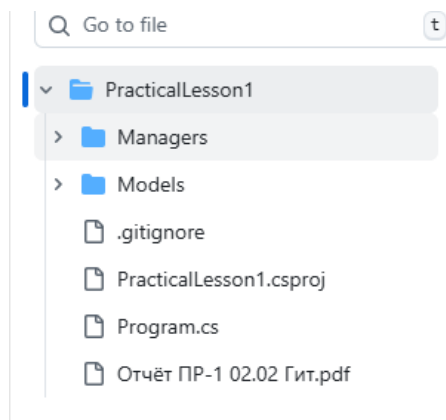
Отправьте коммит в удаленный репозиторий: `git push origin master`

```

PS C:\Users\icn-32\source\repos\PracticalLesson1> git push origin master
Enumerating objects: 17, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 1.94 KiB | 497.00 KiB/s, done.
Total 11 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/JuliaSherstneva/PracticalLesson1.git
06c0fd3..596fdd7 master -> master

```

Убедитесь, что все изменения появились в вашем репозитории



Результат: Добавлен начальный функционал для представления списка дел, а изменения в коде зафиксированы и загружены в GitHub.

## **Заключение**

Таким образом, в ходе выполнения работы были получены навыки инициализации нового консольного проекта на C# в Visual Studio, настроили его для работы с GitHub и загрузили первоначальную версию в репозиторий.