

Разработка консольного приложения «список дел» на C#

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается разработка простого приложения для создания списка дел. Приложение разрабатывается на языке программирования C# с помощью IDE Visual Studio. Практическим результатом является разработанное приложение.

Ключевые слова: список дел, C#, список задач, консоль

Developing a to-do list console application in C#

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes how to develop a simple to-do list application. The application is developed in the C# programming language using the Visual Studio IDE. The practical result is the developed application.

Keywords: to-do list, c#, task list, console

To Do — это приложение, которое помогает планировать и упорядочивать задачи, предстоящие события, идеи и т. д.

От учебных заданий до домашних задач — это поможет упорядочить и приоритизировать то, что хотите сделать. Вот лишь несколько направлений которое закрывает данное приложение:

- Указать сроки выполнения домашних работ;
- Составить список покупок;
- Поставить напоминания;
- Список вещей, которые нужно взять с собой на выездные игры;
- Список поручений, которые необходимо выполнить;
- Документы, которые нужно сдать по проекту;
- Заметки об идеях, песнях, людях или фильмах на будущее.

Цель данной статьи создать простое приложения для списка задач в среде разработки Visual Studio на языке программирования C#.

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления

основных видов и классификаций языков программирования[1]. З. С. Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов[2]. В статье Ф.В. Патюченко, И.С. Слащев, А.В. Клименко, Л.А. Трегубенко были рассмотрены два подхода для создания программ на базе windows, обоснование выбора одного из них [3]. S. Torsten, S. Toshiyuki, M. Derek в своем отчете описали интегрированную программную систему контроля качества, разработанную для значительного повышения производительности и качества анализа проб в лабораториях по анализу бериллия на территории комплекса Министерства энергетики США на языке программирования C# [4].

Создаем проект и называем его. см. рисунок 1.

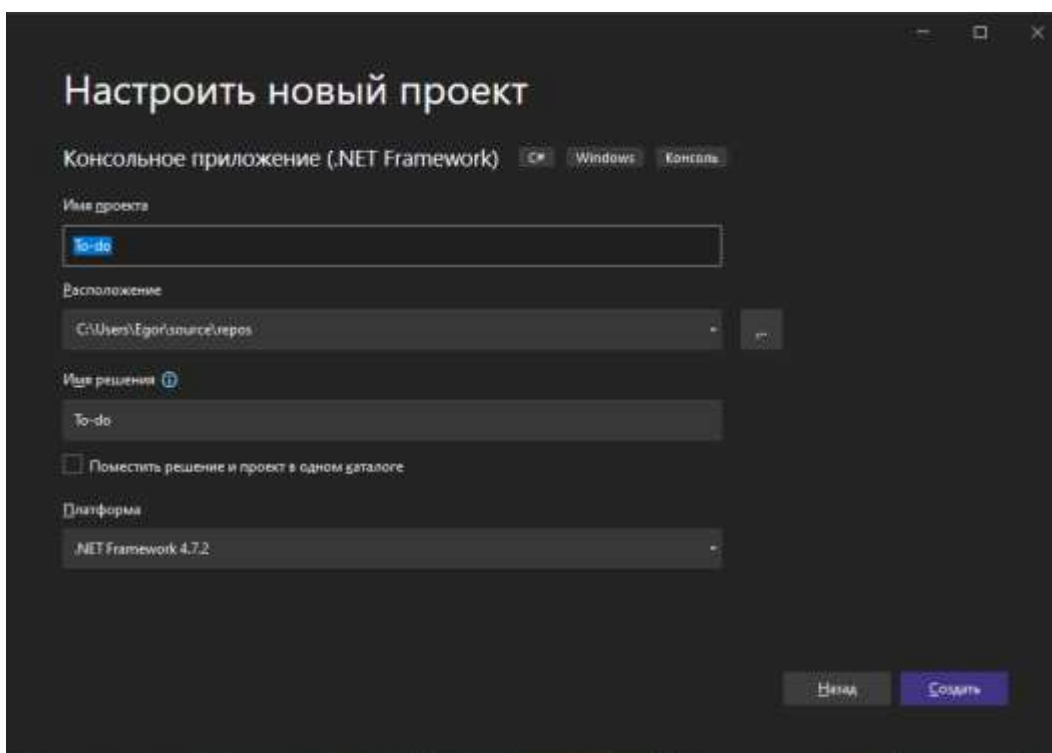


Рис. 1. Создание проекта

Для начала добавим в программу приветственное сообщение см. рисунок 2.

```
Console.WriteLine("Добро пожаловать, это ваш список дел!");
```

Рис. 2. Приветственное сообщение

Далее при помощи класса с типом List создадим список строк для хранения задач в списке. Также создадим переменную с типом string для хранения выбора пользователя см. рисунок 3.

```
List<string> taskList = new List<string>();  
string option = "";
```

Рис. 3. Создания списка для хранения задач

Основная логика программы будет изложена в цикле while, выбор будет осуществлен при помощи условных операторов if, else. Создадим данный цикл и введем основные функциональные кнопки доступные для пользователя см. рисунок 4.

```
while (option != "e")  
{  
    Console.WriteLine("\nЧем бы вы хотели заняться?");  
    Console.WriteLine("Введите 1 чтобы добавить дело в список.");  
    Console.WriteLine("Введите 2 чтобы удалить задачу из списка.");  
    Console.WriteLine("Введите 3 чтобы увидеть список.");  
    Console.WriteLine("Введите e чтобы выйти.\n");  
  
    option = Console.ReadLine();  
  
    if (option == "1")  
    {  
    }  
    else if (option == "2")  
    {  
    }  
    else if (option == "3")  
    {  
    }  
    else if (option == "e")  
    {  
    }  
    else  
    {  
    }  
}
```

Рис. 4. Добавление кнопок выбора функций

Далее приступаем к «оживлению» функционала, в условных операторах прописываем, за что каждая клавиша отвечает см. рисунок 5.

```
if (option == "1")
{
    Console.WriteLine("Пожалуйста введите название для задачи.");
    string task = Console.ReadLine();
    taskList.Add(task);
    Console.WriteLine("Задача добавлена в список.");
}
else if (option == "2")
{
    for (int i = 0; i < taskList.Count; i++)
    {
        Console.WriteLine(i + " : " + taskList[i]);
    }

    Console.WriteLine("Пожалуйста введите номер задачи для удаления.");
    int taskNumber = Convert.ToInt32(Console.ReadLine());
    taskList.RemoveAt(taskNumber);
}
else if (option == "3")
{
    Console.WriteLine("Эти задачи в вашем списке : ");

    for (int i = 0; i < taskList.Count; i++)
    {
        Console.WriteLine(taskList[i]);
    }
}
else if (option == "e")
{
    Console.WriteLine("Выход из программы.");
}
else
{
    Console.WriteLine("Неизвестная опция, попробуйте еще раз.");
}
}
Console.WriteLine("До скорых встреч.");
Console.ReadLine();
```

Рис.5. Добавление логики функциям

Теперь приступаем к проверке функционала программы см. рисунок 6.



Рис.6. Приветственное окно

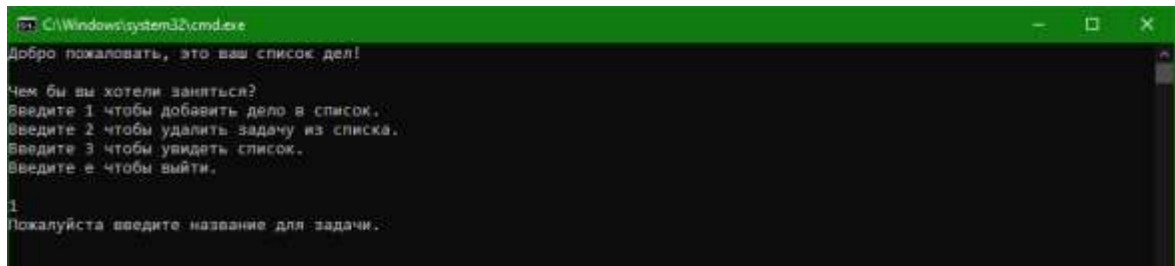


Рис.7. Добавление задачи в список

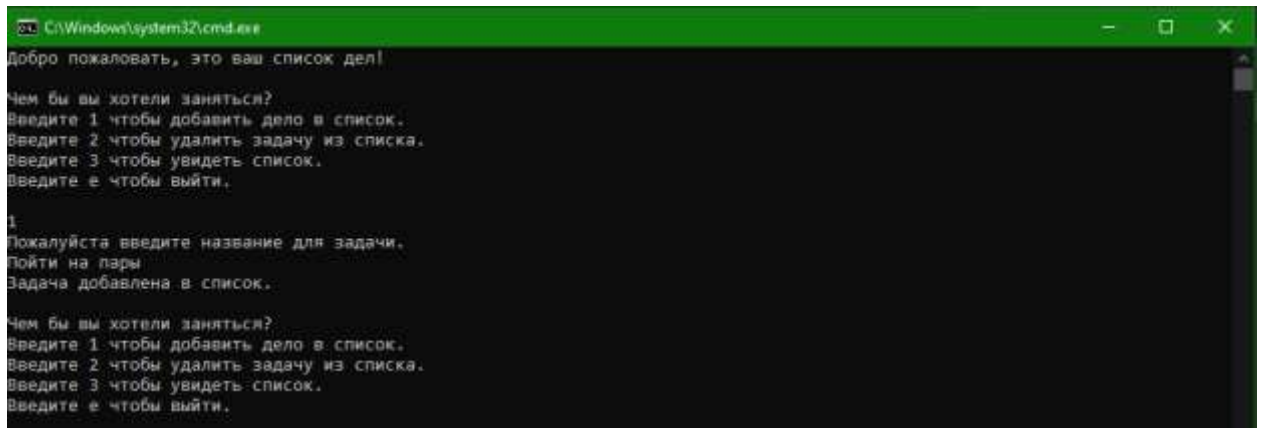


Рис.8. Добавление задачи в список

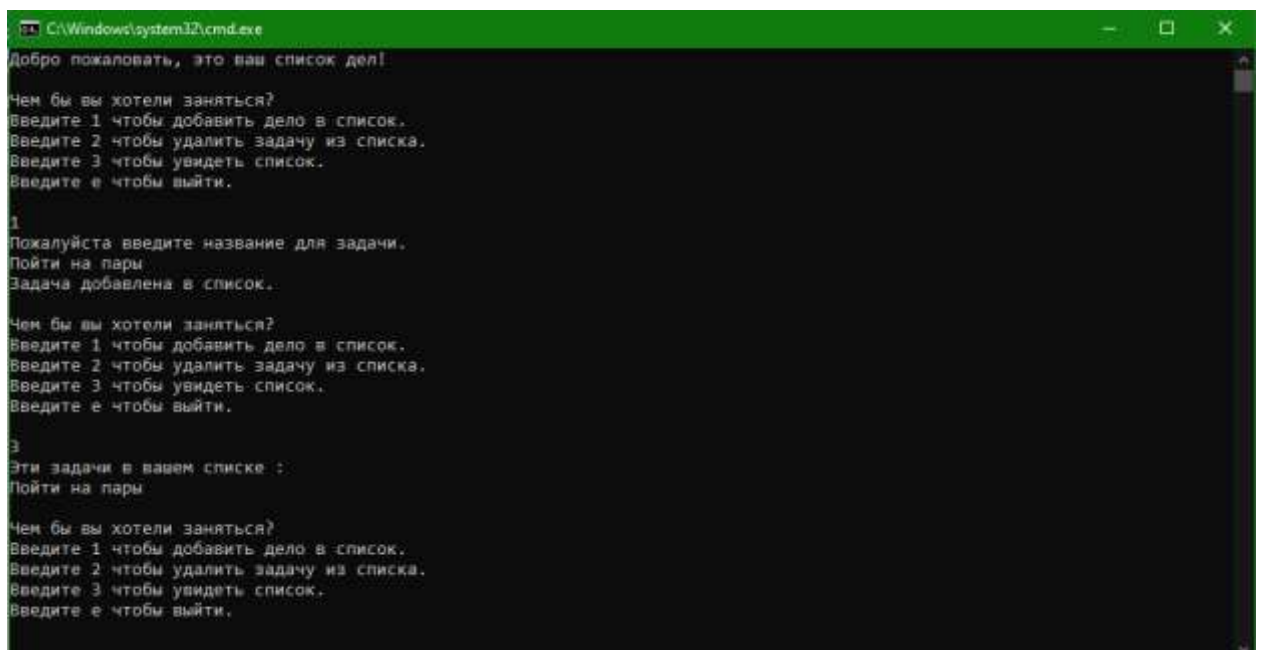
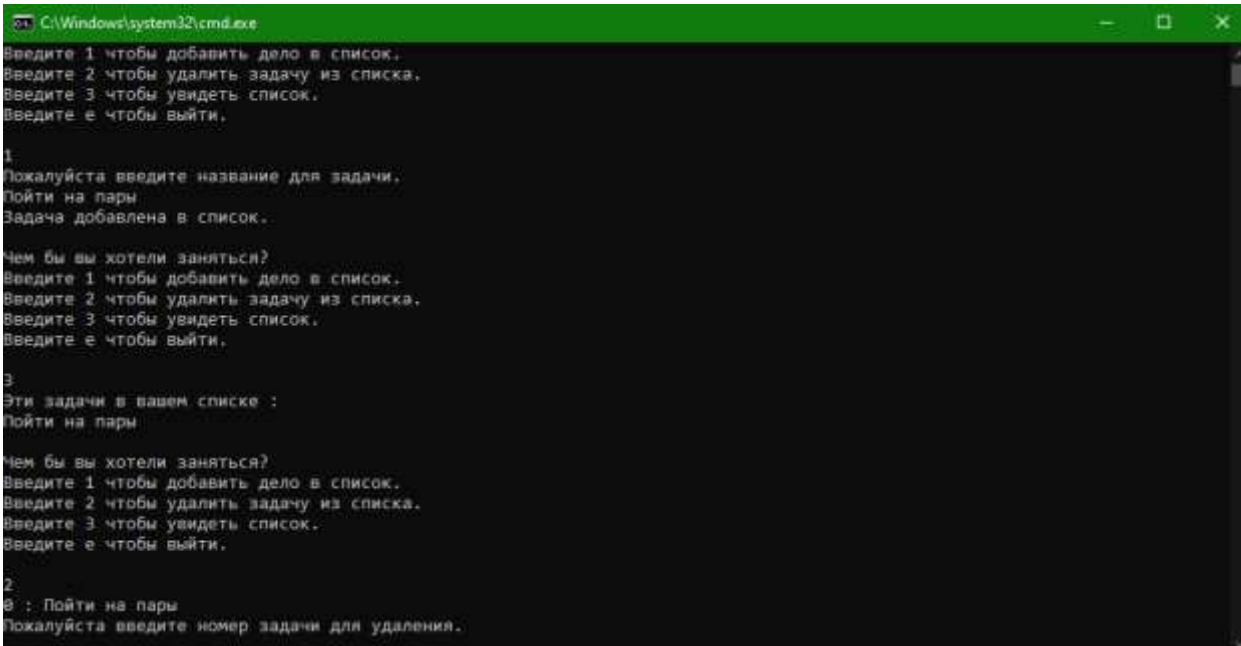


Рис.9. Вывод списка задач



```
C:\Windows\system32\cmd.exe
Введите 1 чтобы добавить дело в список.
Введите 2 чтобы удалить задачу из списка.
Введите 3 чтобы увидеть список.
Введите e чтобы выйти.
1
Пожалуйста введите название для задачи.
Пойти на пары
Задача добавлена в список.

Чем бы вы хотели заняться?
Введите 1 чтобы добавить дело в список.
Введите 2 чтобы удалить задачу из списка.
Введите 3 чтобы увидеть список.
Введите e чтобы выйти.
3
Эти задачи в вашем списке :
Пойти на пары

Чем бы вы хотели заняться?
Введите 1 чтобы добавить дело в список.
Введите 2 чтобы удалить задачу из списка.
Введите 3 чтобы увидеть список.
Введите e чтобы выйти.
2
В : Пойти на пары
Пожалуйста введите номер задачи для удаления.
```

Рис.10. Удаление задачи из списка

В данной статье были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью Visual Studio и языка программирования C#. Такой выбор заметно упростил разработку проекта, так как в интернете имеется достаточное кол-во документации. Во время разработки был полученный ценный опыт работы с этим средством разработки.

В итоге была разработана программа для создания списка дел. Данное консольное приложение имеет потенциал к развитию, например, добавление новых функций.

Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня // Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Патюченко Ф.В., Слащев И.С., Клименко А.В., Трегубенко Л.А. Windows form или windows presentation foundation // Modern science. 2019. №7-2. С. 318-320.
4. Torsten S., Toshiyuki S., Derek M. Improving Sample Analysis Throughput and Quality with a .NET™-based, Real-Time QC Decision Support System // Journal of the Association for Laboratory Automation. 2003. №8. С. 107-112.