Лабораторная работа №1

Создание простейшего консольного приложения на языке C#

Цель работы: получить навыки написания простейших приложений на языке C#.

Теоретическая часть

При разработке приложения в среде Visual Studio рекомендуется руководствоваться следующими правилами:

- каждая практическая работа выполняется в отдельном решении (Solution) с именем ***VP_LabNN.sln***, где NN – номер работы (***VP_PractNN.sln*** для практических работ);
- разрабатываемые классы нужно создавать в отдельном проекте (Project) типа «библиотека классов» (Class library). Имя проекту задается, исходя из назначения классов. В одном решении может быть несколько библиотек классов;
- для работы с пользовательским интерфейсом (консоль, форма и т.д.) создаются отдельные проекты (Console application для консоли). Имя проекту задается, исходя из задачи, для которой этот проект будет использоваться. В конце обязательно указывается UI. Например, RegularExpressionsUI.

Работа с проектом в Visual Studio

Процесс создания решения и проекта представлен ниже:

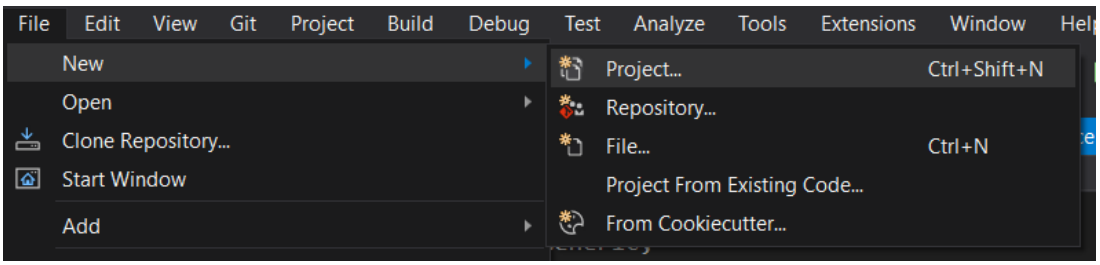

Рисунок 1 – Создание проекта

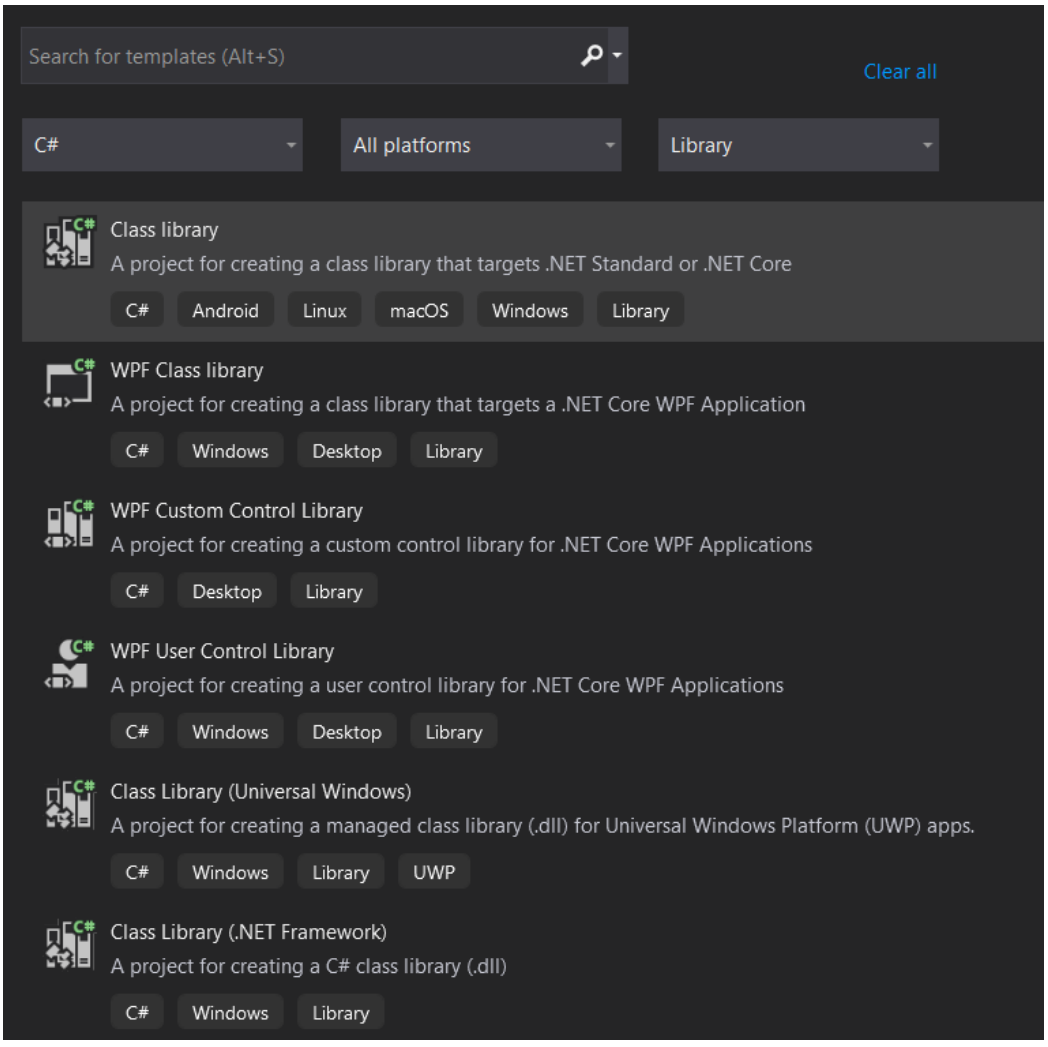

Рисунок 2 – Выбор нужного шаблона

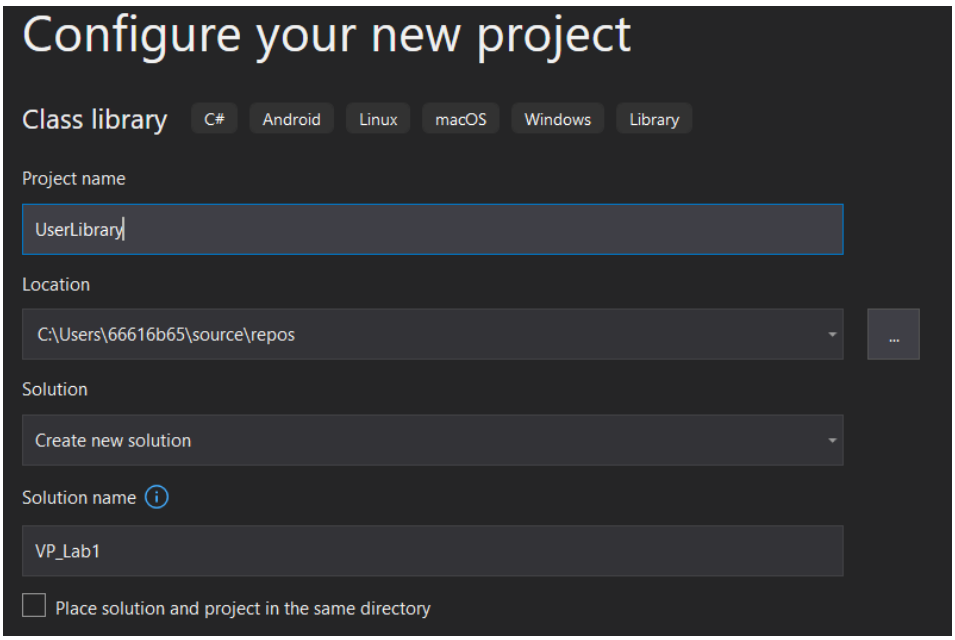

Рисунок 3 – Задание имен проекта и решения

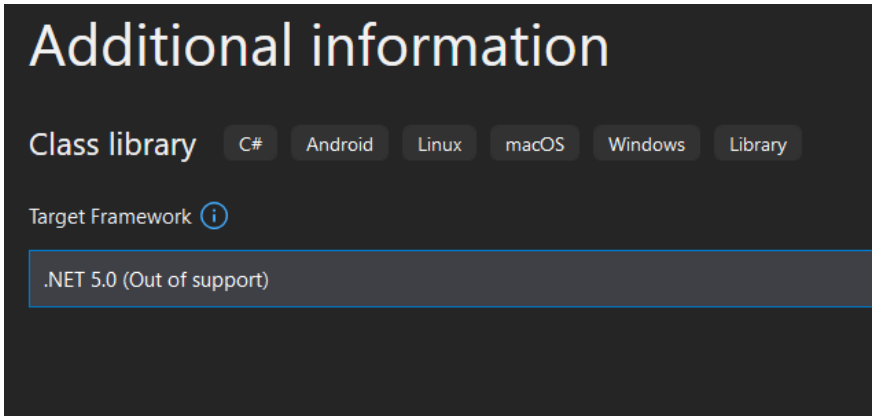

Рисунок 4 – Выбор версии фреймворка

Решение и все его проекты можно посмотреть в окне «Обозреватель решений» (Solution Explorer).

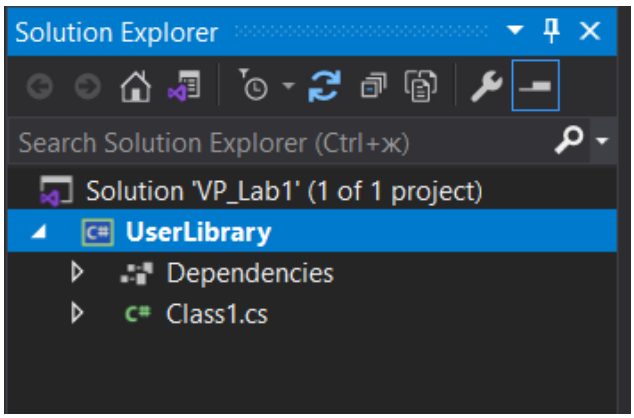

Рисунок 5 – Обозреватель решений

По умолчанию класс имеет имя Class1. Если переименовать его в коде, то поменяется только имя класса, но не имя файла.

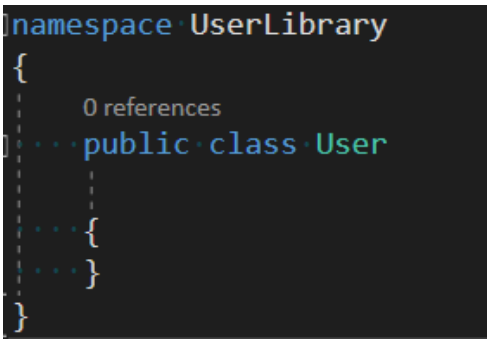

Рисунок 6 – Переименование класса

Для переименования файла нужно выполнить пункт контекстного меню файла Rename (Переименовать). Имя самого класса поменяется автоматически.

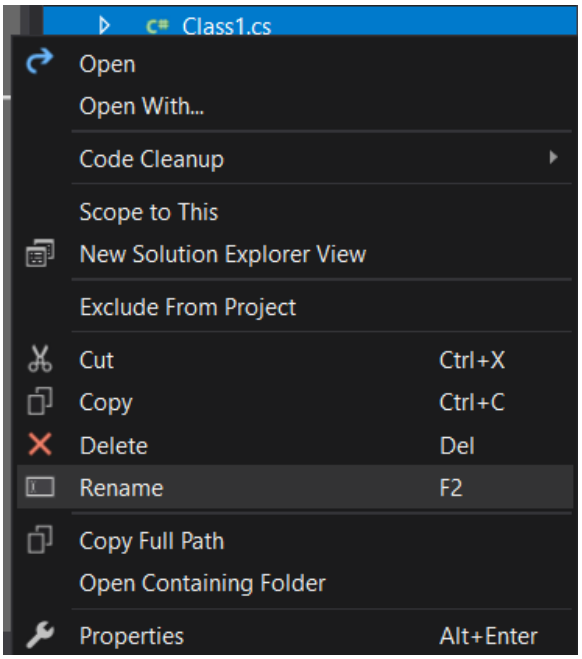

Рисунок 7 – Переименование файла класса

В контекстном меню проекта можно посмотреть доступные действия.

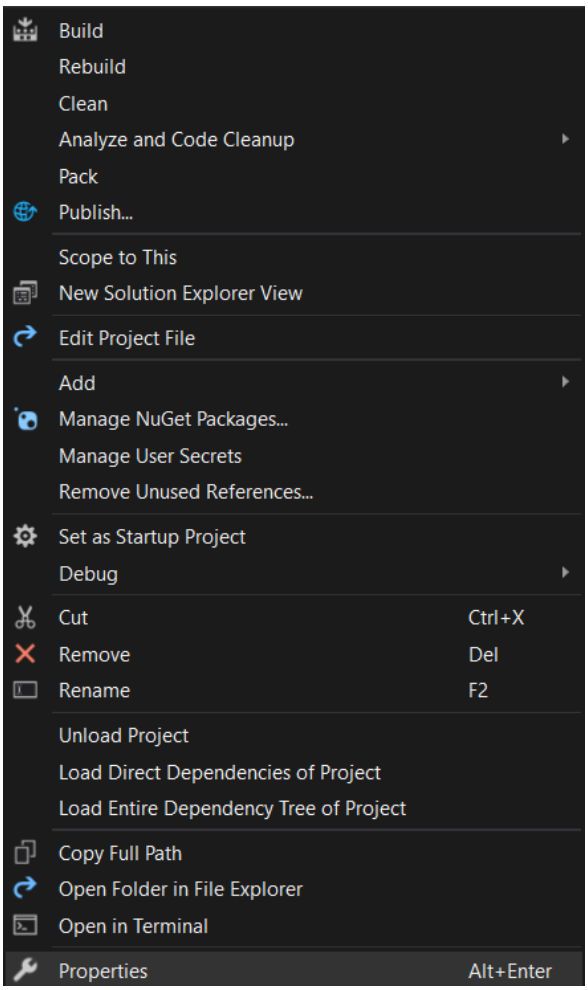

Рисунок 8 – Контекстное меню проекта

Можно выделить следующие действия:

- Build (Собрать);
- Rebuild (Пересобрать);
- Add (Добавить). Например, таким образом можно добавить новый класс или папку;
- Manage NuGet Packages (Управление пакетами NuGet);
- Set as Startup Project (Задать автозапускаемым проектом). Такой проект запускается при отладке (F5);
- Remove (Удалить);
- Rename (Переименовать);
- Open Folder in File Explorer (Открыть в проводнике);
- Properties (Свойства).

В окне Свойства есть вся информация о проекте. Здесь можно изменить версию фреймворка и тип проекта.

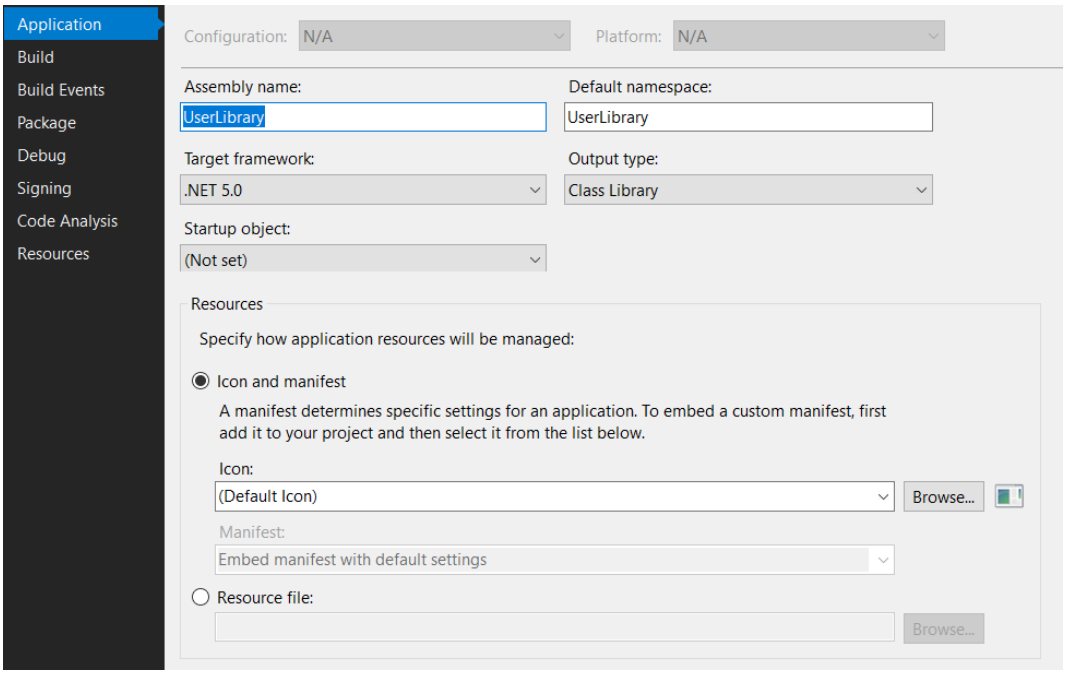

Рисунок 7 – Свойства проекта

Добавить новый проект в решение можно через контекстное меню решения.

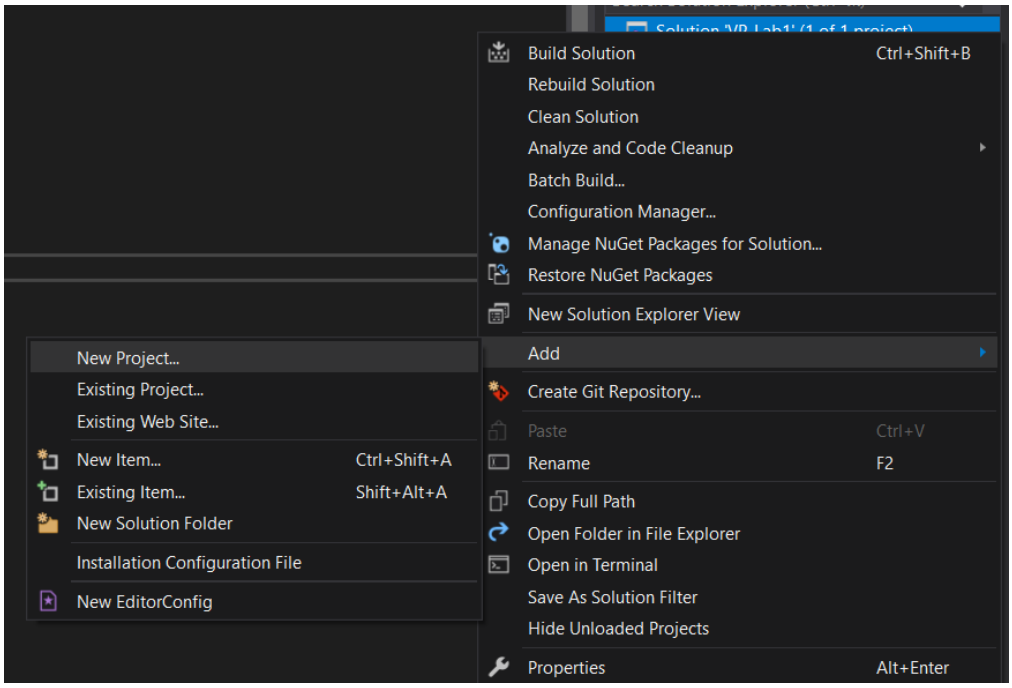

Рисунок 8 – Добавление нового проекта

Добавленный проект можно назначить автозапускаемым. В обозревателе решений его название указывается полужирным шрифтом.

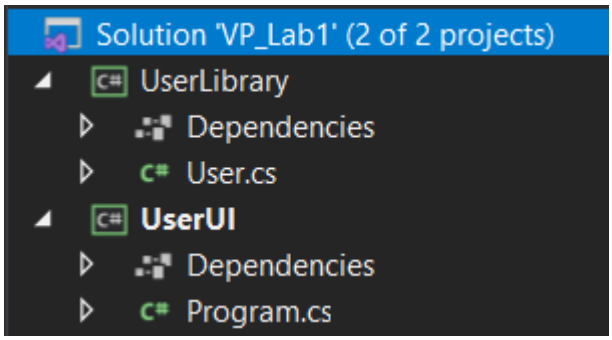

Рисунок 9 – Автозапускаемый проект

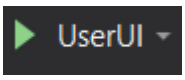

Рисунок 10 – Кнопка отладки

Чтобы в одном из проектов использовать класс, описанный в другом проекте, нужно добавить ссылку на него. Далее показано, как в проекте UserUI использовать класс из проекта UserLibrary.

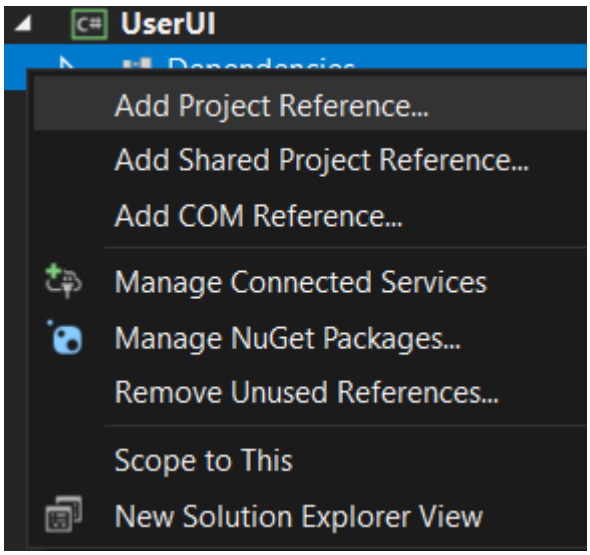

Рисунок 11 – Контекстное меню Dependencies проекта UserUI

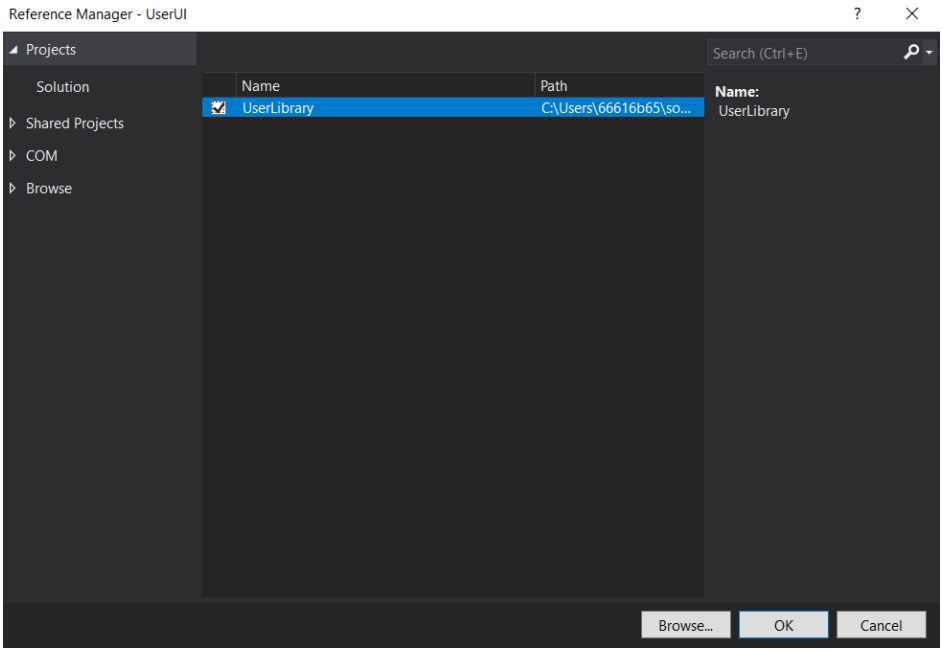

Рисунок 12 – Окно добавления ссылки

Если нужно добавить проект из другого решения – можно воспользоваться кнопкой Browse.

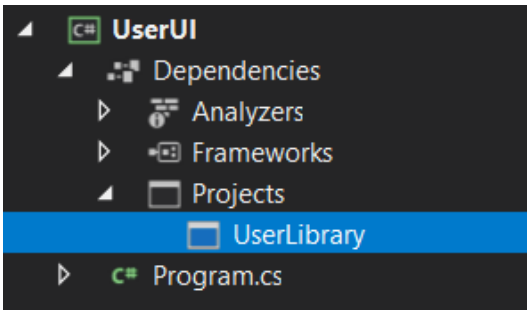

Рисунок 13 – Добавленная ссылка

Попробуем обратиться к добавленному классу.

```
using System;

namespace UserUI
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            User user = new User();
        }
    }
}
```

Рисунок 14 – Обращение к классу

При попытке создать экземпляр класса возникает ошибка. Её можно исправить путем указания пространства имен (namespace):

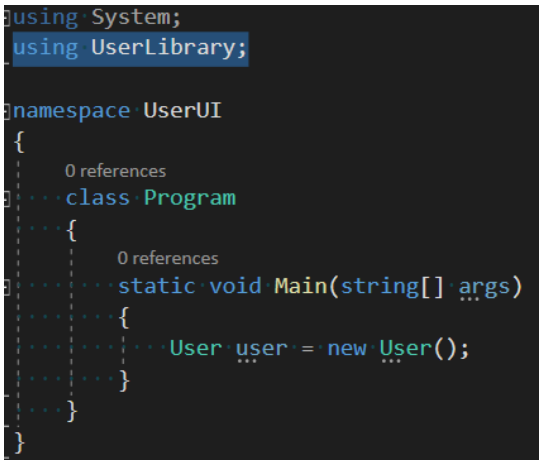

Рисунок 15 – Способ 1

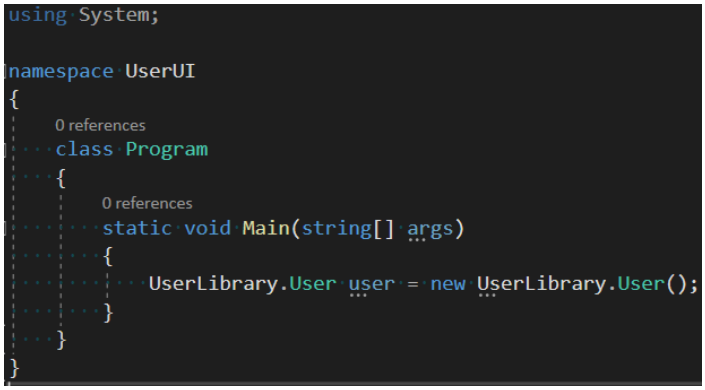

Рисунок 16 – Способ 2

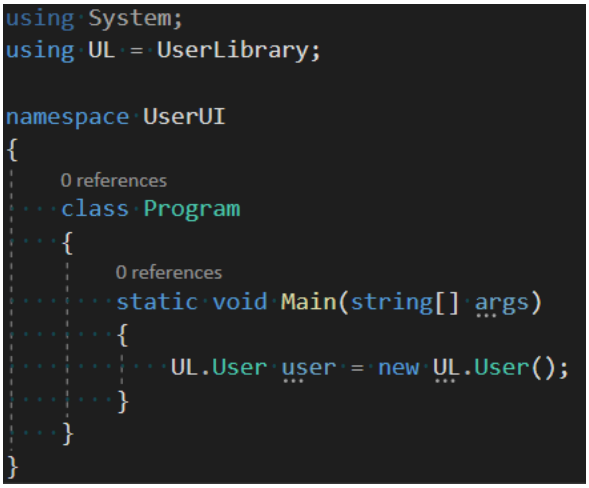

Рисунок 17 – Способ 3 с использованием псевдонима для пространства имен

В пункте меню Debug можно посмотреть доступные режимы отладки.

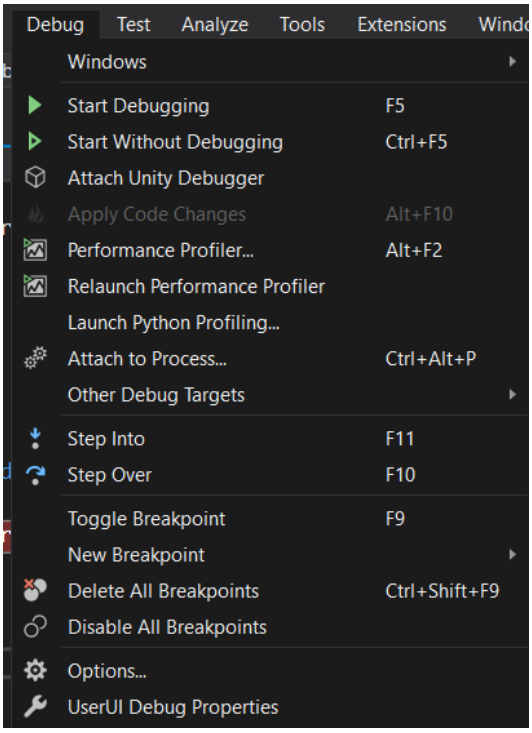

Рисунок 18 – Пункт Debug

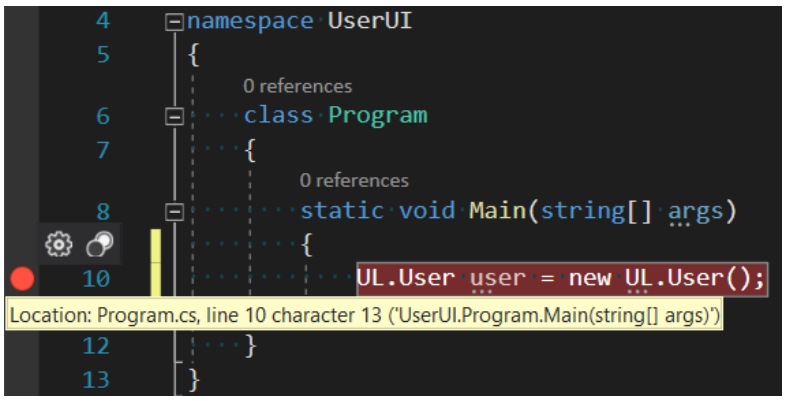

Рисунок 18 – Добавление точки останова

Для удобства отладки полезно добавить окно Watch.

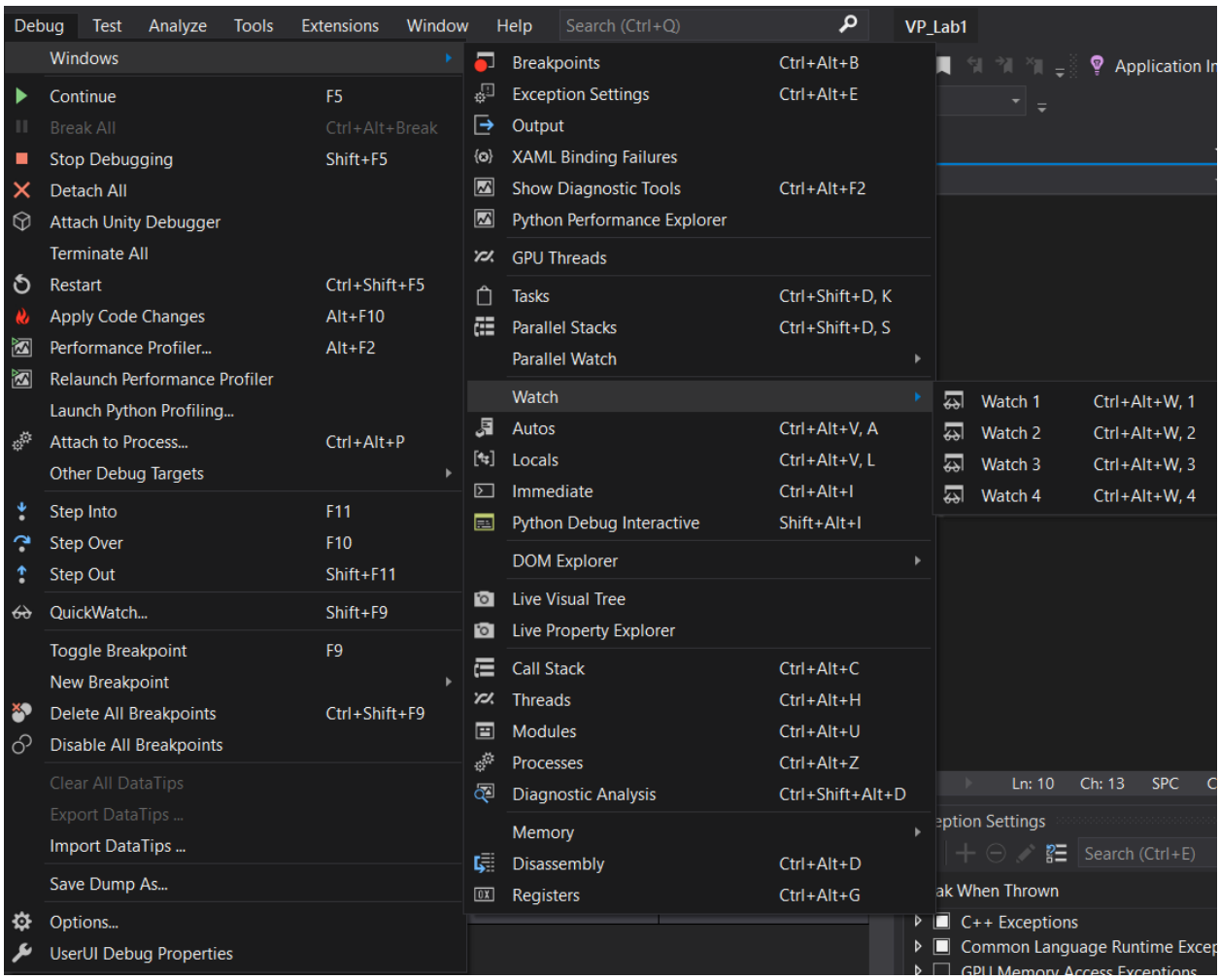

Рисунок 19 – Добавление окна Watch

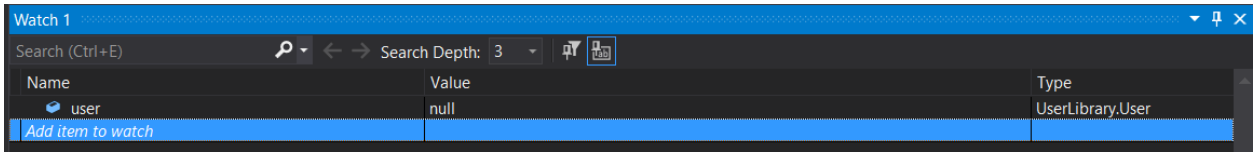

Рисунок 20 – Окно Watch

Качество кода

В процессе написания кода на языке C# разработчику рекомендуется использовать соглашения о написании кода (Coding Convention).

Соглашения о написании кода предназначены для реализации следующих целей:

– Создание согласованного вида кода, позволяющего читателям сосредоточиться на содержимом, а не на структуре.

- Предоставление читателям возможности делать предположения, основанные на опыте, и поэтому быстрее понимать код.
- Упрощение процессов копирования, изменения и обслуживания кода.
- Предоставление лучших методик C#.

Соглашения могут отличаться в зависимости от компании или проекта. Для начинающих разработчиков целесообразно ознакомиться со следующими соглашениями:

- https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/coding-conventions
- https://learn.microsoft.com/ru-ru/dotnet/standard/design-guidelines/general-naming-conventions
- https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md

Также рекомендуется ознакомиться с антипаттернами в программировании:

https://bool.dev/blog/detail/antipatterny-v-programmirovanii-i-proektirovanii-arkhitektury

Практическая часть

Требования:

1. Все программы выполняются в консоли.
2. Все решения оформляются в виде статических методов в отдельном статическом классе (или классах). В классе Program методы **только вызываются**.

3. Метод должен принимать входные данные и возвращать результат. Например, если нужно посчитать количество символов в строке, то метод должен принять строку и вернуть целое число. Вывод результатов в консоль осуществляется в классе Program.

№ бригады	Номера задач
1	1, 2
2	3, 4
3	5, 6
4	7, 8
5	9, 10
6	11, 12
7	13, 14
8	15, 16
9	17, 18
10	19, 20
11	1, 11
12	2, 12
13	3, 13

Задача 1. Введите с клавиатуры строку произвольной длины и подсчитайте процент вхождения заданного символа в строку.

Задача 2. Задан массив действительных чисел размерности 10×10. Найти суммы элементов каждой строки, произведения элементов каждого столбца, и максимальный элемент главной диагонали (подсказка: все элементы, для которых номер строки совпадает с номером столбца).

Задача 3. В заданной строке текста определите количество слов. Каждое слово отделено друг от друга пробелом.

Задача 4. Задан одномерный массив целых чисел. Образуйте из него два отсортированных по возрастанию массива, содержащих четные и нечетные числа. Подсказка: четное число делится на 2 без остатка.

Задача 5. В заданном массиве действительных чисел найдите разность между максимальным и минимальным числом.

Задача 6. В одномерном массиве из 100 чисел M[] подсчитайте количество элементов, удовлетворяющих условию 0 < M[i] <125.

Задача 7. Определите, является ли исходная строка символов палиндромом (читается одинаково с начала и с конца). Регистры символов и пробелы игнорируйте.

Задача 8. Задана квадратная матрица целых чисел. Подсчитайте количество отрицательных и положительных элементов, а также выведите на печать координаты нулевых элементов (номер строки и номер столбца).

Задача 9. Введите с клавиатуры строку произвольной длины и подсчитайте процент вхождения гласных букв латинского алфавита в строку (не различая регистры).

Задача 10. Задан массив действительных чисел из N элементов (используйте генератор случайных чисел). Определить количество элементов, значения которых находятся в диапазоне от -100 до +100.

Задача 11. Задано пять произвольных целых чисел (элементы массива). Определить, является ли их расположение в массиве упорядоченным (т.е. по возрастанию или по убыванию) или неупорядоченным.

Задача 12. Задан массив действительных чисел из N элементов (используйте генератор случайных чисел). Определить количество элементов, значения которых находятся вне диапазона от -10 до +10.

Задача 13. В двухмерном массиве переставьте попарно соседние строки, т.е. 1-ю со 2-ой, 3-ю с 4-й и т.д. Результат выведите на экран.

Задача 14. Определите, присутствует ли в тексте, заданном в виде строки, некоторое слово (различие регистра игнорируйте).

Задача 15. Задан генератором случайных чисел одномерный массив из действительных чисел. Найдите максимальное и минимальное число этого массива.

Задача 16. Напишите методы для ввода одномерного массива с клавиатуры и для вывода его содержимого на экран.

Задача 17. В массиве строк (список фамилий) определите самую длинную фамилию.

Задача 18. Найдите сумму и произведение элементов квадратной матрицы размерности 10×10.

Задача 19. Задано три треугольника со своими сторонами (a, b, c). Найти треугольник с наибольшим периметром или наибольшей площадью.

Задача 20. Сформируйте массив целых чисел по алгоритму Фибоначчи: 1-й и 2-й элемент равны 1, а каждый последующий равен сумме двух предыдущих, т.е.: 1, 1, 2, 3, 5, 8, … . Найдите сумму и произведение его N членов.

Содержание отчета

1. Титульный лист
2. Цель работы
3. Задание
4. Код программы
5. Результат выполнения