

# Лабораторная работа № 1

## Обзор языка С#

### В лабораторной работе используется Microsoft Visual Studio 2008

#### Упражнение 1 – Создание простой программы на С#

Напишите программу «Приветствие». Программа должна запрашивать имя пользователя, сохранять его в переменной и выводить на экран приветствие «Hello, *имя, которое ввел пользователь*».

1. Откройте Visual Studio 2008.
2. В меню File выберите New, затем выберите Project.
3. Из списка шаблонов выберите Console Application.
4. Впишите название проекта, например, Greeting.
5. Укажите путь к проекту C:\Labs\Lab1
6. Измените имя класса на Greeter.
7. Сохраните проект, выбрав из меню File опцию Save All.
8. Далее переходим к написанию программы. В методе **Main** введите `string myName;`
9. Напишите команду, приглашающую пользователя ввести его (ее) имя.
10. Напишите команду, которая выполняет чтение введенного пользователем значения.
11. Добавьте ее одну команду, которая обеспечивает вывод на экран приветствия «Hello, *имя, которое ввел пользователь*».
12. В результате Вы должны получить следующий программный код

```
static void Main(string[] args)
{
    string myName;
    Console.WriteLine("Please enter your name");
    myName = Console.ReadLine();
    Console.WriteLine("Hello {0}", myName);
}
```

13. Сохраните проект, выбрав из меню File опцию Save All.
14. В меню Debug выберите Start Without Debugging (или CTRL+F5).
15. В окне консоли должна появиться подсказка «Please enter your name» (или аналогичная).
16. Впишите имя.
17. После этого появится приветствие и отобразится подсказка “Press any key to continue”
18. Нажмите клавишу и закончите выполнение программы.

#### Упражнение 2 – Компиляция и запуск программы С# из командной строки

Необходимо откомпилировать из командной строки программу, созданную в упражнении 1.

1. Откройте окно командной строки.
2. Перейдите в каталог, где находится проект Greeting: C:\Labs\Lab1\Greeting.
3. Откомпилируйте программу, используя следующую команду:  
`csc /out:Greet.exe Program.cs`

4. Запустите программу, введя  
Greet
5. Закройте окно командной строки.

### Упражнение 3 – Отладка программы с использованием Visual Studio Debugger

В этом упражнении мы будем использовать отладчик программ.

1. В Visual Studio 2008 откройте проект, который был создан в упражнении 1. Для этого в меню File выберите Open, затем выберите Project и укажите в папке C:\Labs\Lab1\Greeting Greetings.sln.
2. Установите курсор на строку, в которой оператор Console.WriteLine встречается первый раз.
3. В меню Debug запустите Start или F5.
4. Выполните пошаговую отладку. Выберите опцию Step Over (F10). Пройдите по шагам до конца программы.
5. В меню Debug выберите Continue или F5.

### Упражнение 4 – Добавление обработки исключений в программу на C#

В этом упражнении необходимо написать программу, которая требует обработки исключений.

1. Откройте Visual Studio 2008, если она не открыта.
2. В меню File выберите New, затем выберите Project.
3. Из списка шаблонов выберите Console Application.
4. Впишите название проекта, например, Divider.
5. Укажите путь к проекту C:\Labs\Lab1
6. Измените имя класса на DivideIt.
7. Сохраните проект, выбрав из меню File опцию Save All.
8. Далее переходим к написанию программы. В методе *Main* введите  
int i, j  
string temp;
9. Напишите команду, приглашающую пользователя ввести первое целое число.
10. Напишите команду, которая выполняет чтение введенного пользователем значения.
11. Напишите выражение, которое конвертирует введенное пользователем строковое значение, соответствующее переменной *temp*, в целочисленное значение и сохраните результат в переменной *i*.
12. Напишите команду, приглашающую пользователя ввести второе целое число.
13. Прочитайте ответ и поместите в переменную *temp*.
14. Конвертируйте значение переменной *temp* в *j*.
15. Сохраните проект, выбрав из меню File опцию Save All.
16. Напишите код, который создает новую целочисленную переменную *k*, которой присваивается результат деления *i* на *j*.
17. Добавьте оператор, отображающий значение *k*.

18. Протестируйте работу программы. Проверьте ее работу, указав корректные данные, напр. 10 и 5, и неправильные данные (деление на 0). Зафиксируйте результат.
19. Добавьте исключения. Оформите блок *Try*. Напишите блок *Catch*.
20. Снова протестируйте работу программы. Проверьте ее работу, указав корректные данные, напр. 10 и 5, и неправильные данные (деление на 0). Зафиксируйте результат.

Natalie E. Novakova