

## Лабораторная работа №1. Разработка консольных приложений C#

### Часть I. Подготовка к работе

1. Запустите интегрированную среду разработки **Microsoft Visual Studio 2010**.
2. Создайте новый проект:
  - 1) выберите пункты главного меню **Файл – Создать – Проект...**
  - 2) в колонке Установленные шаблоны выберите категорию **Visual C#**;
  - 3) выберите шаблон **Консольное приложение**;
  - 4) в поле ввода **Имя** укажите название проекта **Example\_1**;
  - 5) в поле ввода **Имя решения** введите **Лабораторная работа\_1**;
  - 6) выберите **Расположение** проекта на вашем диске;
  - 7) нажмите кнопку **ОК**.
3. В **Обозревателе решений** рассмотрите структуру проекта.
4. В **Окне редактора кода** проанализируйте содержимое модуля исходного кода **Program.cs** и добавьте одноименные однострочные комментарии к его частям, содержащим:
  - 1) включение пространств имен;
  - 2) объявление пространства имен **Example\_1**;
  - 3) класс **Program**;
  - 4) главный метод (точку входа в программу).
5. Выведите на экран приветствие. В тело главного метода добавьте следующий программный код:

```
Console.WriteLine("Hello, world!"); //вывод строки
Console.ReadLine(); //ожидание нажатия клавиши Enter
```
6. Выполните отладку проекта, выбрав в главном меню пункт **Отладка – Начать отладку** или нажав клавишу **F5**.
7. Проверьте правильность работы программы.
8. Сохраните изменения, выбрав в Главном меню пункт **Файл – Сохранить Program.cs** или нажав комбинацию клавиш **Ctrl+S**.
9. Теперь вы можете открыть решение Лабораторная работа 1 через команды главного меню, выбрав файл **Лабораторная работа\_1.sln** или запустив его двойным щелчком из каталога сохранения.

## Часть II. Консольный ввод-вывод

**Задание 1:** Выведите на экран приветствие пользователя по имени и определите наличие совершеннолетия.

1. В главном методе объявите строковую переменную **name**:

```
string name;
```

2. Обеспечьте ввод имени с клавиатуры:

```
Console.WriteLine("Введите имя: ");  
name = Console.ReadLine(); //считывание введенной пользователем строки в переменную
```

3. Перед выводом приветствия выполните очистку экрана:

```
Console.Clear();
```

4. Выведите приветствие:

```
Console.WriteLine("Здравствуй, {0}! ", name);
```

5. Выполните отладку и проверьте правильность работы программы.

6. **Самостоятельно** добавьте в программу считывание фамилии и приветствие по имени и фамилии.

7. Объявите строковую переменную **input** и целочисленную (тип **byte**) **age**.

8. Выведите сообщение с запросом возраста пользователя и считайте введенное значение в строковую переменную.

9. Проверьте, можно ли преобразовать введенное значение в число. Если да, то определите признак совершеннолетия:

```
//если строку input можно преобразовать в тип byte,  
//то в переменную age будет записано это значение  
if (byte.TryParse(input, out age))  
{  
    if (age < 18)  
    {  
        Console.WriteLine("Несовершеннолетний");  
    }  
    else  
        Console.WriteLine("Совершеннолетний");  
}  
else  
    Console.WriteLine("Ошибка ввода");
```

10. Выполните отладку и проверьте правильность работы программы. В качестве тестовых данных введите числа большие или равные 0, отрицательные числа, строки.

**Задание 2.** Создайте приложение с консольным меню для отображения текущей даты и времени.

1. Добавьте новый проект **Example\_2**.

2. Назначьте новый проект главным: **Проект – Назначить запускаемым проектом**.

3. Объявите и проинициализируйте переменные:

```
string menu, choice=""; //сообщение, ответ пользователя  
DateTime today;         //переменная типа дата-время
```

```
//строковой переменной присваиваем значение дословного литерала
menu = @" Выберите действие:
1 - Вывести текущую дату;
2 - Вывести текущее время;
3 - выход из программы";

today = DateTime.Now; //присваиваем текущую дату и время соответствующей переменной
```

4. Обеспечьте вывод текстового меню и интерпретацию ответа пользователя, до тех пор, пока не будет выбран пункт «Выход из программы»:

```
while (choise != "3") //условие завершения цикла - пока пользователь не введет «3»
{
    Console.Clear(); //очистка экрана
    Console.WriteLine(menu); //вывод текстового меню
    choise = Console.ReadLine(); //считывание ответа пользователя

    switch (choise) //проверяем варианты ответа пользователя
    {
        //если пользователь выбрал 1й пункт, выводим дату
        case "1": Console.WriteLine("Сегодня: {0}", today.ToShortDateString());
                    break;

        //если пользователь выбрал 2й пункт, выводим время
        case "2": Console.WriteLine("Сейчас: {0}", today.ToShortTimeString());
                    break;

        //если пользователь выбрал 3й пункт, выходим из программы
        case "3": Console.WriteLine("Выход из программы");
                    break;

        //если пользователь ввел другое значение
        default: Console.WriteLine("Ошибка ввода");
                    break;
    }
    Console.ReadLine();
}
```

5. Выполните отладку и проверьте правильность работы программы.

**Задание 3.** Разработайте приложение для формирования и вывода на экран заданного количества случайных целых чисел.

1. В новом проекте **Example\_3** объявите следующие переменные:

```
byte count = 0; //количество генерируемых чисел
string input;
int n; //случайное число
```

2. Создайте и инициализируйте генератор случайных чисел:

```
Random rand = new Random();
```

3. Запросите у пользователя количество генерируемых чисел и считайте ответ в строковую переменную **input**.

4. В блоке **try** опишите инструкции для формирования указанного количества случайных чисел:

```
try
{
    count = System.Convert.ToByte(input); //преобразование строки input в целое count
    if (count>0)
    {
        for (int i=1; i<=count; i++)
        {
            n = rand.Next(0, 11); //генерируем число в диапазоне от 0 до 10
        }
    }
}
```

```

        Console.WriteLine("{0}-е число: {1} ", i, n);
    }
}

```

5. В блоках **catch** выполните обработку исключительных ситуаций, которые могут возникнуть при преобразовании строки методом `Convert.ToByte()`:

```

catch (System.ArgumentNullException) //введена пустая строка (отсутствует аргумент)
{
    Console.WriteLine("Ошибка: пустая строка");
}
catch (System.FormatException) //введено значение неверного типа
{
    Console.WriteLine("Ошибка: неправильный формат");
}
catch (System.OverflowException) //значение выходит за диапазон byte
{
    Console.WriteLine("Ошибка: переполнение");
}

```

6. Выполните отладку и проверьте правильность работы программы при вводе различных значений.

### Часть III. Работа с массивами и создание методов

**Задание 4:** Создайте программу, выполняющую подсчет количества «холодных» дней в январе (дней, когда среднесуточная температура была ниже 0).

1. Добавьте в решение проект **Example\_4**.

2. Объявите переменные:

```
const byte n = 31; //количество дней
short [] tempJan = new short [n]; //массив со значениями температуры по дням
Random rand = new Random(); //создание генератора чисел
byte coldDays = 0; //количество "холодных" дней
```

3. Заполните массив случайными значениями температур в диапазоне от -10 до 15 и выведите на экран по 7 значений в строке:

```
for (byte i = 0; i < n; i++)
{
    //генерируем число от -10 до 10 и заносим в массив
    tempJan[i] = (short)rand.Next(-10, 15);

    //добавляем перенос строки для разделения недель
    if (i % 7 == 0) Console.WriteLine();

    //выводим значение температуры текущего дня
    Console.Write("{0,3}", tempJan[i].ToString());
}
```

4. Проверьте результат работы программы.

5. Подсчитайте и выведите количество дней с отрицательной температурой:

```
foreach (short tempDay in tempJan) //для каждого элемента массива проверяем
{
    if (tempDay < 0) //если температура меньше 0
    {
        coldDays++; //увеличиваем количество «холодных» дней
    }
}
Console.WriteLine("Температура опускалась ниже нуля {0} раз", coldDays);
```

6. Проверьте правильность работы программы.

**Задание 5:** Разработайте приложение для создания и просмотра списка пользователей.

1. Добавьте в решение проект **Example\_5**.

2. В классе **Program** объявите следующие переменные:

```
class Program
{
    const byte n = 30; //максимальное количество пользователей
    static short count = 0; //текущее количество пользователей
    static string[] userList = new string[n]; //список пользователей
}
```

3. Далее опишите программный код метода добавления пользователя в список:

```
static void addUser()
{
    if (count < n)
    {
        Console.WriteLine("Введите имя пользователя:");
    }
}
```

```

        string userName = Console.ReadLine(); //считываем имя
        //если такое имя не найдено в массиве
        if (Array.BinarySearch(userList, userName) < 0)
        {
            userList[count] = userName; //добавляем новое значение в массив
            count++; //увеличиваем текущее количество элементов
            Array.Sort(userList); //сортировка массива
            Console.WriteLine("Пользователь добавлен");
        }
        else
            Console.WriteLine("Пользователь с таким именем существует");
    }
    else
        Console.WriteLine("Превышено допустимое количество пользователей");
    return; //завершаем выполнение метода
}

```

4. Там же опишите программный код метода просмотра списка пользователей:

```

static void viewList()
{
    Console.WriteLine("Список пользователей: ");
    if (count > 0)
    {
        foreach (string userName in userList)
        {
            if (userName != null) //если ячейка массива не содержит пустую строку
            {
                Console.WriteLine(userName);
            }
        }
    }
    else
        Console.WriteLine("Нет записей!");
    return;
}

```

5. В главном методе реализуйте меню консольного приложения:

```

string menu = "Выберите действие: \n1 - добавить пользователя;\n2 - вывести список\nпользователей;\n3 - выход";
string inputString="";
do
{
    Console.Clear();
    Console.WriteLine(menu); //вывод меню
    inputString = Console.ReadLine(); //считывание выбранного пункта меню
    Console.Clear();

    switch (inputString) //проверяем выбор
    {
        case "1": addUser(); break; //вызов функции добавления пользователя
        case "2": viewList(); break; //вызов функции просмотра списка
        case "3": Console.WriteLine("Работа программы завершена"); break;
    }
    Console.ReadLine();
}
while (inputString != "3");

```

6. Выполните отладку и проверьте правильность работы программы.

**Задание 6:** Выполните заполнение и обработку двумерного массива с помощью методов.

1. Добавьте в решение проект **Example\_6**.
2. Объявите следующие константы и переменные в классе **Program**:

```

const byte n = 10; //количество строк
const byte m = 10; //количество столбцов
static short [,] massiv = new short [n, m]; //двумерный массив

```

- Там же добавьте метод заполнения массива случайными значениями, в который диапазон значений (минимальное и максимальное) передается в качестве параметров:

```

static void setArray(short minValue, short maxValue)
{
    Random rand = new Random();
    for (byte i = 0; i < n; i++)
    {
        for (byte j = 0; j < m; j++)
        {
            massiv [i,j] = (short)rand.Next(minValue, maxValue + 1);
        }
    }
    return;
}

```

- Для вывода на экран элементов массива, добавьте функцию, возвращающую в качестве результата строку:

```

static string printArray()
{
    string outputString = "";
    for (byte i = 0; i < n; i++)
    {
        for (byte j = 0; j < m; j++)
        {
            outputString = outputString + String.Format("{0,4}", massiv [i, j]);
        }
        outputString = outputString + '\n';
    }
    return outputString; //возвращаем результат функции
}

```

- В главном методе объявите переменные:

```

short minValue, maxValue; //диапазон значений

```

- Обеспечьте ввод с клавиатуры диапазона значений.

- Заполните массив и выведите его на экран:

```

setArray(minValue, maxValue);
Console.WriteLine(printArray()); //выводим на экран результат выполнения функции

```

- Выполните отладку проекта и проверьте правильность выполнения операций.

- В классе **Program** опишите функцию, которая вычисляет среднее значение элементов массива:

```

static short AVERAGE()
{
    short result; int summ = 0;

    foreach (short element in massiv) //подсчитываем сумму всех элементов
        summ = summ + element;

    result = (short) (summ / (n * m)); //вычисляем среднее значение

    return result; //возвращаем результат метода
}

```

- В методе **Main** вызовите метод **Average** и выведите на экран полученные данные:

```

Console.WriteLine("Среднее значение равно {0}", AVERAGE());

```

11. Проверьте правильность работы программы.

12. **Самостоятельно** дополните программу функцией вывода количества элементов с заданным пользователем значением.



## Часть V. Задания для самостоятельного решения

**Задание:** Разработайте консольное приложение в среде MS Visual Studio на языке C# для решения задачи согласно своему варианту.

**В программе должны быть предусмотрены:**

- 1) меню;
- 2) выход из программы по выбору соответствующего пункта меню;
- 3) проверка корректности ввода данных;
- 4) обработка возможных исключительных ситуаций;
- 5) комментарии к основным блокам и переменным;
- 6) 3 сценария тестирования согласно шаблону.

**Шаблон сценария тестирования:**

Наименование пункта	Описание
Название проекта	Название проекта Visual Studio.
Название тестирования	Описание проверяемого условия. Например, проверка корректности ввода цены товара.
Резюме испытания	Описание результата тестирования. Например, вывод сообщения об ошибке.
Шаги тестирования	Перечисление детальных шагов тестирования в виде нумерованного списка действий пользователя.
Данные тестирования	Тестовые данные, используемые для этого тестирования. Например, значение цены – 5 рублей.
Постусловия	Какое состояние у системы будет после проведения тестирования. Например, работа программы будет завершена.

**Вариант 1:** В двумерном массиве хранятся данные о товарах на мебельном складе: название, тип, производитель. Реализуйте следующий функционал: добавление новой позиции на склад, вывод полного списка товаров, вывод на экран и подсчет количества позиций выбранного пользователем типа мебели, поиск и отображение сведений о товаре по введенному пользователем названию.

Входные данные:

Поле	Значение	Пример значения
Название	Уникальное	Комфорт
Тип	Может повторяться	Диван
Производитель	Может повторяться	ОАО «Диваны и кресла»

**Вариант 2:** В двумерном массиве хранятся данные об абитуриентах колледжа: количество баллов по русскому языку, математике и код специальности. Обеспечьте следующий функционал: добавление новой записи, вывод полного списка с указанием суммы баллов по двум предметам, поиск и отображение списка отличников (имеющий по 100 баллов за каждый предмет), вывод среднего балла абитуриентов по выбранной пользователем специальности.

Входные данные:

Поле	Значение	Пример значения
Русский язык	0 – 100, может повторяться	75
Математика	0 – 100, может повторяться	65
Код специальности	1 – 10, может повторяться	2

**Вариант 3:** В двумерном массиве хранится информация о сотрудниках компании: ФИО, специальность и подразделение. Обеспечьте следующий функционал: добавление новой записи, вывод полного списка сотрудников, поиск и отображение информации о сотруднике по фамилии, вывод и подсчет количества сотрудников выбранного пользователем подразделения.

Входные данные:

Поле	Значение	Пример значения
ФИО	Если повторяется, добавить номер	Иванов Иван Иванович 2
Должность	Может повторяться	Продавец
Подразделение	Может повторяться	Отдел продаж

**Вариант 4:** В двумерном массиве хранится информация о книгах домашней библиотеки: название, автор, жанр. Обеспечьте следующий функционал: добавление новой записи, вывод полного списка книг, поиск по названию и автору (одновременно), вывод на экран и подсчет количества книг выбранного пользователем жанра.

Входные данные:

Поле	Значение	Пример значения
Название	Может повторяться	Портрет Дориана Грея
Автор	Может повторяться	Оскар Уайльд
Жанр	Может повторяться	Роман

**Вариант 5:** В двумерном массиве хранится информация о поставках товаров на овощную базу: номер склада, количество килограммов моркови, количество килограммов картофеля. Обеспечьте следующий функционал: добавление новой записи, вывод полного списка; вывод на экран позиций, находящихся на выбранном пользователем складе; вывод на экран номеров складов, на которых общее количество товара меньше введенного пользователем значения.

Входные данные:

Поле	Значение	Пример значения
Код склада	1-5, может повторяться	3
Количество моркови	Целое число (кг)	100
Количество картофеля	Целое число (кг)	52

**Вариант 6:** В двумерном массиве хранятся результаты олимпиады по информатике: код участника, количество баллов за тест, количество баллов за практическое задание. Обеспечьте следующий функционал: добавление новой записи; вывод полного списка участников; поиск и вывод кодов участника(-ов), набравших наибольшее количество баллов; подсчет и вывод количества участников, не справившихся с тестом, практическим заданием или обоими заданиями.

Входные данные:

Поле	Значение	Пример значения
Код участника	Целое уникальное число	6
Баллы за тест	0 – 100, может повторяться	68
Баллы за практическое задание	0 – 10, может повторяться	8

**Вариант 7:** В двумерном массиве хранятся данные об участниках IT-конференции: ФИО, город, название секции. Обеспечьте следующий функционал: добавление новой записи; вывод полного списка участников; вывод списка участников из выбранного пользователем города; подсчет количества участников в секции, указанной пользователем.

Входные данные:

Поле	Значение	Пример значения
ФИО	Если повторяется, добавить номер	Петров Петр Петрович 3
Город	Может повторяться	Новосибирск
Секция	Может повторяться	Облачные технологии

**Вариант 8:** В двумерном массиве хранятся данные о пользователях информационной системы. Обеспечьте следующий функционал: добавление новой записи, вывод полного списка пользователей, поиск и вывод информации о пользователе по значению логина, вывод на экран общего количества пользователей указанного пользователем типа учетной записи.

Входные данные:

Поле	Значение	Пример значения
ФИО	Может повторяться	Васильев Василий Васильевич
Логин	Уникальное значение	Vasya
Пароль	6 символов, может повторяться	123456
Тип учетной записи	Может повторяться	Администратор