Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 1**

**по дисциплине**

**«Программирование компьютерной графики»**

Тема работы: Разработка консольного приложения

Выполнил:

ст. гр. ПРИ-120

М. А. Бочков

Принял:

преп. каф. ИСПИ

Жигалов И.Е.

Владимир, 2023

**Цель работы:**

Освоение принципов программирования в среде Microsoft Visual C# для операционной системы Windows с использованием Microsoft .net Framework. Получение практических навыков построения консольных приложений в Visual C#.

**Ход работы:**

**Задание**

1. Ознакомиться по методическим указаниям и литературе с назначением среды VisualC#, ее компонентами и особенностями работы в ней.

2. Выполнить действия, приведенные в разделах 1.2, 1.3 для построения простой консольной программы, изучить ее структуру. При разработке программы имя проекта, создаваемого в MS Visual Studio, должно содержать фамилию студента и группу (например, Ivanov\_Ivan\_ISG\_105\_lab\_1).

3. Дополнить программу, добавив в реализуемый класс (раздел 1.3) два новых метода / команды, управляющих поведением объекта.

На экран должны выводиться текстовые строки, поясняющие работу программы в данном режиме и содержащие ФИО и группу автора.

**Задание 2**

Листинг C#:

internal class Program

{

private static void Main(string[] args)

{

Console.WriteLine("Бочков Михаил Алексеевич ПРИ-120");

int resault = 777;

if (resault == 777)

{

Console.WriteLine("Congratulations, you win!!!!"); // выведет: «Поздравляем, вы победили»

}

else

{

Console.WriteLine("please, try again \n"); // выведет: «Пожалуйста, попробуйте снова»

Console.WriteLine("we are confident - you will be lucky"); //выведет: «Мы уверены- вам повезет»

}

int value = 2;

switch (value)

{

case 1:

Console.WriteLine("Цифра 1");

break;

case 2:

Console.WriteLine("Цифра 2");

break;

case 3:

Console.WriteLine("Цифра 3");

break;

default:

Console.WriteLine("для этого варианта действие не определено");

break;

}

int a = 100;

while (a > 5)

{

Console.WriteLine(a);

a--;

}

int b = 100;

do

{

Console.WriteLine(a);

b--;

} while (b > 5);

for (int i = 100; i > 5; i--)

{

Console.WriteLine(a);

}

int[] arr = new int[] { 1, 2, 3 }; // наш массив

foreach (int x in arr)

{

System.Console.WriteLine(x);

}

int[] arr\_name = new int[2];

arr\_name[0] = 1;

arr\_name[1] = 2;

//int[] arr\_name = new int[] { 1, 2 };

// int[] arr\_name = { 1, 2 };

int[,] arr\_name\_1 = new int[4, 2];

// int[,] arr\_name\_2 = { { 0, 1, 2, 3 }, { 0, 1, 2 } };

for (int i = 0; i < arr\_name\_1.GetLength(0); i++)

{

for (int j = 0; j < arr\_name\_1.GetLength(1); j++)

{

System.Console.WriteLine(arr\_name\_1[i, j]);

}

}

}

}

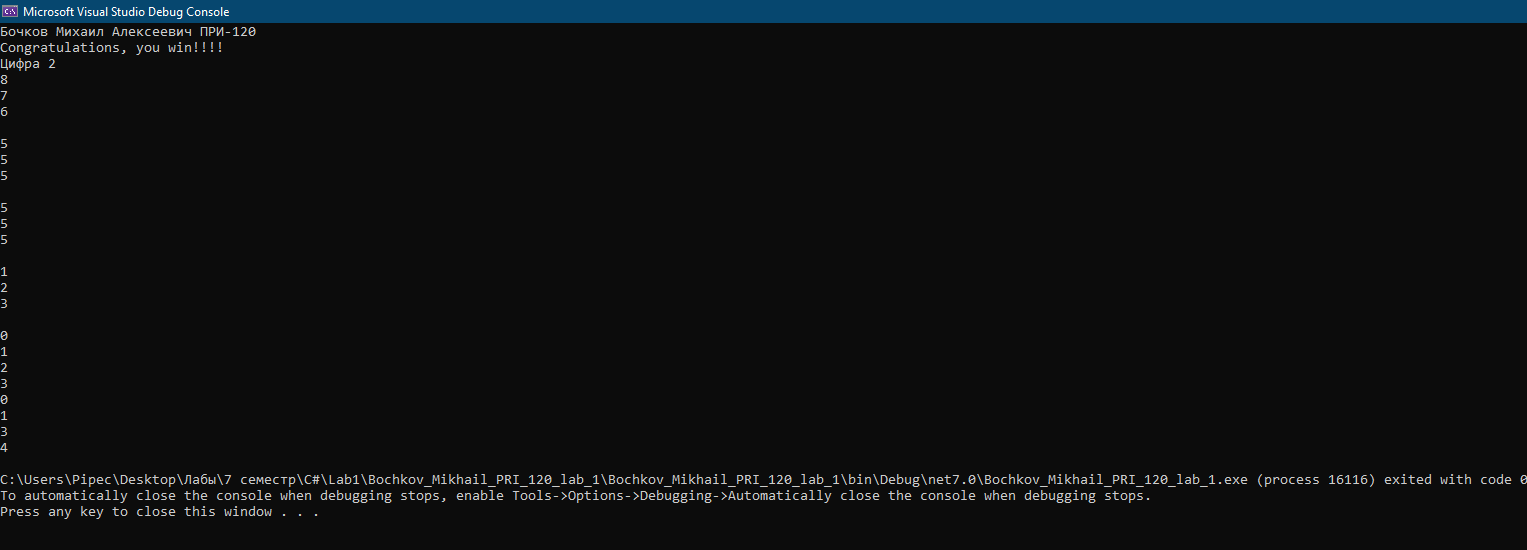


Рисунок - результат выполнения задания №2

**Задания 3**

Листинг С#:

internal class Program

{

private static void Main(string[] args)

{

Console.WriteLine("Бочков Михаил Алексеевич ПРИ-120");

// переменная, которая будет хранить команду пользователя

string user\_command = "";

// бесконечный цикл

bool Infinity = true;

Man SomeMan = null;

while (Infinity) // пока Infinity равно true

{

// приглашение ввести команду

Console.WriteLine("Пожалуйста, введите команду");

// получение строки (команды) от пользователя

user\_command = Console.ReadLine();

switch (user\_command)

{

// если user\_command содержит строку exit

case "exit":

Infinity = false;

// теперь цикл завершиться, и программа завершит свое выполнение

break;

// если user\_command содержит строку help

case "help":

Console.WriteLine("Список команд:");

Console.WriteLine("---");

Console.WriteLine("create\_man : команда создает человечка, (экземпляр класса Man)");

Console.WriteLine("kill\_man : команда убивает человечка");

Console.WriteLine("talk : команда заставляет человечка говорить (если создан экземпляр класса)");

Console.WriteLine("go : команда заставляет человечка идти (если создан экземпляр класса)");

Console.WriteLine("info : команда показывает инмормацию о человеке (если создан экземпляр класса)");

Console.WriteLine("resurrect : команда воскрешает человека, если он мертв (если создан экземпляр класса)");

Console.WriteLine("---");

Console.WriteLine("---");

break;

case "create\_man":

{

// проверяем, создан ли уже какой либо человечек

if (SomeMan != null)

{

// человечек уже существует - "убиваем" его

SomeMan.Kill();

}

// просим ввести имя человечка

Console.WriteLine("Пожалуйста, введите имя создаваемого человечка \n");

// получаем строку введенную пользователем

user\_command = Console.ReadLine();

// создаем новый объект в памяти, в качестве параметра

// передаем имя человечка

SomeMan = new Man(user\_command);

// сообщаем о создании

Console.WriteLine("Человечек успешно создан \n");

SomeMan.ShowInfo();

break;

}

case "kill\_man":

{

// проверяем, что объект существует в памяти

if (SomeMan != null)

{

// вызываем фукнцию "смерти"

SomeMan.Kill();

}

break;

}

case "talk":

{

// проверяем, что объект существует в памяти

if (SomeMan != null)

{

// вызываем фукнцию разговора

string resultTalking = SomeMan.Talk();

Console.WriteLine($"{resultTalking}");

}

else // иначе

{

System.Console.WriteLine("Человечек не создан. Команда не может быть выполнена");

}

break;

}

case "go":

{

// проверяем, что объект существует в памяти

if (SomeMan != null)

{

// вызываем фукнцию передвижения

SomeMan.Go();

}

else

{

System.Console.WriteLine("Человечек не создан. Команда не может быть выполнена");

}

break;

}

case "info":

if (SomeMan != null)

{

SomeMan.ShowInfo();

}

else

{

Console.WriteLine("Человечек не создан. Команда не может быть выполнена");

}

break;

case "resurrect":

if (SomeMan != null)

{

SomeMan.Resurrect();

}

else

{

Console.WriteLine("Человечек не создан. Команда не может быть выполнена");

}

break;

// если команду определить не удалось

default:

System.Console.WriteLine("Ваша команда не определена, пожалуйста повторите снова");

System.Console.WriteLine("Для вывода списка команд введите команду help");

System.Console.WriteLine("Для завершения программы введите команду exit");

break;

}

}

}

}

public class Man

{

private Random rnd = new Random();

// конструктор класса (данная функция вызывается

// при создании нового экземпляра класса

public Man(string name)

{

Name = name;

isLife = true;

Age = (uint)rnd.Next(15, 50);

// и здоровье, от 10 до 100%

Health = (uint)rnd.Next(10, 100);

}

// закрытые члены, которые нельзя именить

// извне класса

// строка, содержащая имя

private string Name;

// беззнаковое целое число, содержащая возраст

private uint Age;

// беззнаковое целое число, отражающее уровень здоровья

private uint Health;

// булево, означающее жив ли данный человечек

private bool isLife;

// заготовка функции "говорить"

public string Talk()

{

// генерируем случайное число от 1 до 3

int random\_talk = rnd.Next(1, 3);

// объявляем переменную, в которой мы будем хранить строку

string tmp\_str = "";

switch (random\_talk)

{

case 1: // если 1 - называем свое имя

tmp\_str = "Привет, меня зовут " + Name + ", рад познакомиться";

break;

case 2: // возраст

// генериуем текст сообщения

tmp\_str = "Мне " + Age + ". А тебе?";

// завершаем оператор выбора

break;

case 3: // говорим о своем здоровье

// в зависимости от параметра здоровья

if (Health > 50)

tmp\_str = "Да я зводоров как бык!";

else

tmp\_str = "Со здоровьем у меня неважно, дожить бы до " + (Age + 10).ToString();

// завершаем оператор выбора

break;

}

return tmp\_str;

}

// заготовка функции "идти"

public void Go()

{

// если объект жив

if (isLife == true)

{

// если показатель более 40

// считаем объект здоровым

if (Health > 40)

{

// генерируем строку текста

string outString = Name + " мирно прогуливается по городу";

// выводим в консоль

System.Console.WriteLine(outString);

}

else

{

// генерируем строку текста

string outString = Name + " болен и не может гулять по городу";

// выводим в консоль

System.Console.WriteLine(outString);

}

}

else

{

// генерируем строку текста

string outString = Name + " не может идти, он умер";

System.Console.WriteLine(outString);

}

}

// функция, возвращающая показатель - жив ли данный человечек.

public bool IsAlive()

{

// возращаем значение, к которому мы не можем

// обратиться на прямую из вне класса,

// так как оно имеет статус private

return isLife;

}

public void Kill()

{

// устанавливаем значение isLife (жив)

// в false...

isLife = false;

Console.WriteLine("Человек убит");

}

public void ShowInfo()

{

Console.WriteLine($"Имя: {Name}");

Console.WriteLine($"Возраст: {Age}");

Console.WriteLine($"Здоровье: {Health}");

if (isLife == true) Console.WriteLine("Человек жив");

else Console.WriteLine("Человек мертв");

}

public void Resurrect()

{

if (isLife == false)

{

isLife = true;

Console.WriteLine("Человек воскрешён");

}

else { Console.WriteLine("Человек и так жив"); }

}

}

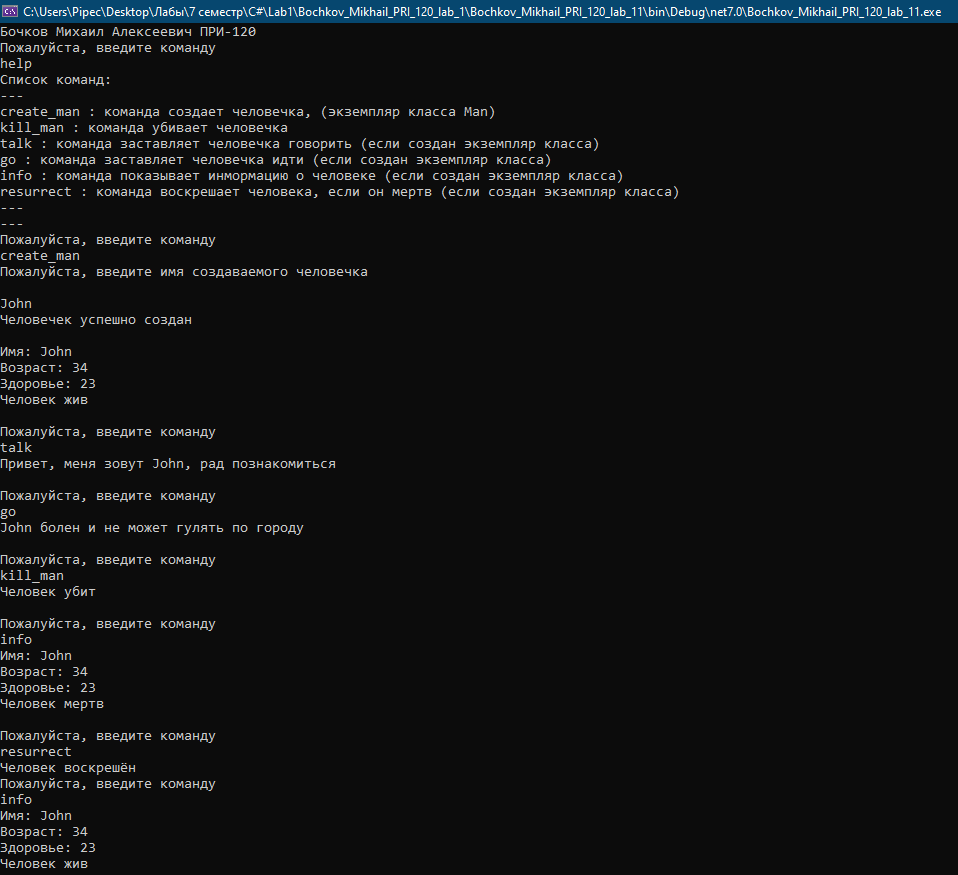


Рисунок 2 - результат выполнения задания 3

**Вывод**

В процессе выполнения работы были освоены принципы программирования в среде Microsoft Visual C# для операционной системы Windows с использованием Microsoft .net Framework, были получены практические навыки построения консольных приложений в Visual C#