

Министерство образования и науки Российской Федерации

ФГАОУ ВО «КФУ им. В.И. Вернадского»

физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

Шор Константин Александрович

отчет по лабораторной работе №2
по дисциплине «**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ**»

Направление подготовки:
09.03.01 "Информатика и вычислительная техника"

Оценка - "отлично"



Симферополь, 2021

Лабораторная работа №2.

Тема: Использование программных конструкций C#

Цель работы: научиться преобразовывать различные типы данных в C#, познакомиться с типом данных Decimal, научиться грамотно использовать циклы для итерационных вычислений с контролем погрешности, обрабатывать события нажатия клавиш, научиться использовать классы String, StringBuilder, компоненты Grid или DataGridView в Windows Forms или WPF приложениях

Описание ключевых понятий:

Класс – описание объекта, также объект представляет экземпляр класса;

Статический тип – возвращаемое значение функции связывается с типом в момент объявления и тип не может быть отменён позже;

Динамический тип – переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной;

Встроенные типы - фундаментальные типы, которые задаются стандартами языка C# и встроены в компилятор; **IL** - MSIL (Microsoft Intermediate Language, IL), управляемый код, двухэтапная компиляция,

Типы значения – переменная типа значения содержит экземпляр типа.;

Ссылочные типы – переменная типа содержит ссылку на экземпляр типа;

Фундаментальные типы: логический, символьный, целый, с плавающей точкой, void, указатели, ссылки, массивы, перечисления, структуры, классы

Преобразования типов: упаковать; распаковать; неявное преобразование; явное преобразование; класс Convert;

Перед выполнением лабораторной работы изучена следующая литература:

1. Изучил презентацию лектора: «Типы в C#» и «Приведение типов в C#» Прослушана видеолекция сотрудника Microsoft С.Байдачного «Net Framework»
2. Прочитаны 3-4 лекции Биллиг В.А. Основы объектного программирования на языке C#
3. Просмотрен видеоурок #50, #38 на ютуб канале SimpleCode
4. Изучил нужные сведения на сайте MSDN Microsoft и Metanit.com.

Выполнены 4 задания, описанных в методических указания к выполнению лабораторных работ.

Задание 0: Приведение и преобразование типов.

```

static void Main(string[] args)
{
    /*ТАБЛИЦА НЕЯВНЫХ ПРЕОБРАЗОВАНИЙ
    *sbyte -> short,int,long,float,double,decimal,nint;
    *byte -> short,ushort,int,uint,long,ulong,float,double,decimal,nint,nuint;
    *short -> int,long,float,double,decimal,nint;
    *ushort -> int,uint,long,ulong,float,double,decimal,nint,nuint;
    *int -> long,float,double,decimal,nint;
    *uint -> long,ulong,float,double,decimal,nuint;
    *long -> float,double,decimal;
    *ulong -> float,double,decimal;
    *float -> double;
    *nint -> long,float,double,decimal;
    *nuint -> ulong,float,double,decimal;
    */
    // (1) Неявное преобразование простых типов
    byte a = 2;
    double b = a + 0.6;
    Console.WriteLine("Неявное простое: " + b);
    // (1) Неявное преобразование ссылочного типов
    object obj = new Silka { A = 7};
}

```

Рис.1 Неявное преобразование

```

/*ТАБЛИЦА ЯВНЫХ ПРЕОБРАЗОВАНИЙ
*sbyte -> byte,ushort,uint,ulong,nuint;
*byte -> sbyte;
*short -> sbyte,byte,ushort,uint,ulong,nuint;
*ushort -> sbyte,byte,short;
*int -> sbyte,byte,short,ushort,uint,ulong,nuint;
*uint -> sbyte,byte,short,ushort,int;
*long -> sbyte,byte,short,ushort,int,uint,ulong,nint,nuint;
*ulong -> sbyte,byte,short,ushort,int,uint,long,nint,nuint;
*float -> sbyte,byte,short,ushort,int,uint,long,ulong,decimal,nint,nuint;
*double -> sbyte,byte,short,ushort,int,uint,long,ulong,float,decimal,nint,nuint;
*decimal -> sbyte,byte,short,ushort,int,uint,long,ulong,float,double,nint,nuint;
*nint -> sbyte,byte,short,ushort,int,uint,long,ulong,nuint;
*nuint -> sbyte,byte,short,ushort,int,uint,long,ulong,nint;
*
*/
// (2) Явное преобразование простых типов
double t = 3.8;
int d = (int)t;
Console.WriteLine("Явное простое:" + d);
// (2) Явное преобразование ссылочного типов
Silka silka = (Silka)obj;
silka.Print();

```

Рис.2 Явные преобразования

```

// (3) Вызвал и обработал исключение преобразования типов
try
{
    t = Convert.ToDouble("s");
}
catch
{
    Console.WriteLine("Исключение");
}

```

Рис.3 Обработка исключений

```

// (4) Безопасное приведение с помощью as
ссылка: 1
static void SafelyAS(object obj)
{
    Silka silka = obj as Silka;

    if (silka != null)
    {
        silka.Print();
    }
}

// (4) Безопасное приведение с помощью is
ссылка: 1
static void SafelyIS(object obj)
{
    if (obj is Silka)
    {
        Silka silka = (Silka)obj;
        silka.Print();
    }
}

```

Рис.4 Безопасное преобразование с помощью as и is

```

/* (6) Преобразование с помощью
Parse, TryParse класса System
string str = "7";
int c = Convert.ToInt32(str);
int C = c + 1;
Console.WriteLine(C);

int i = int.Parse(str);
int I = i + 2;
Console.WriteLine(I);

int g;
int.TryParse(str, out g);
int G = g + 3;
Console.WriteLine(g);

//
SafelyAS(obj);
SafelyIS(obj);

```

Рис.5 Convert, Parse, TryParse

Задание 1: Вычисление квадратного корня с контролем точности.

```
private void inputTextBox_KeyDown(object sender, KeyEventArgs e)
{
    string TextBox = inputTextBox.Text;
    decimal.TryParse(TextBox, out decimal numberDecimal);

    if (e.Key == Key.Enter)
    {
        i++;
        iteration.Content = i;
        quess = numberDecimal / 2;
        result = ((numberDecimal / quess) + quess) / 2;

        for (int j = 0 ; i > j; j++)
        {
            result = quess;
            quess = ((numberDecimal / quess) + quess) / 2;
            sqrt.Content = result;
        }

        delta = result - Convert.ToDecimal( newtonLabel.Content);
        Delta.Content = delta;
    }
}
```

Рис.6 Вычисление корня

Задание 2: Конвертация целочисленных данных в бинарные.

```
private void convertButton_Click(object sender, RoutedEventArgs e)
{
    string content = inputTextBox.Text;
    bool error = int.TryParse(content, out int i);

    if (error)
    {
        if (i >= 0)
        {
            int remainder = 0;

            do
            {
                remainder = i % 2;
                i = i / 2;

                binary.Insert(0, remainder);
                binaryLabel.Content = binary.ToString();
            } while (i > 0);
            return;
        }
        else
        {
            MessageBox.Show("Please enter a positive number or zero");
        }
    }
}
```

Рис.7 Конвертация в бинарный код

Задание 3: Умножение матриц.

```
// Рандомное заполнение
ссылка: 1
private void But1_Click(object sender, RoutedEventArgs e)
{
    int Col1 = Convert.ToInt32(Col_1.Text);
    int Row12 = Convert.ToInt32(ColRow.Text);
    double[,] matrix1 = new double[Row12, Col1];

    var random = new Random();

    for(int i = 0; i < Row12; i++)
    {
        for (int j = 0; j < Col1; j++)
        {
            matrix1[i, j] = random.Next(100);
        }
    }
    Matrix.initializeGrid(ref Matrix1, matrix1);
}
```

Рис.8 Рандомное заполнение матриц

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    int Col1 = Convert.ToInt32(Col_1.Text);
    int Col2 = Convert.ToInt32(Col_2.Text);
    int Row12 = Convert.ToInt32(ColRow.Text);

    double[,] matrix1 = Matrix.getValuesFromGrid(Matrix1);
    double[,] matrix2 = Matrix.getValuesFromGrid(Matrix2);
    double[,] matrix3 = new double[Row12, Col2];

    for (int i = 0; i < Row12; i++)
    {
        for (int j = 0; j < Col2; j++)
        {
            matrix3[i,j] = 0;
            for (int k = 0; k < Col1; k++)
            {
                matrix3[i, j] += matrix1[i, k] * matrix2[i, k];
                Matrix.initializeGrid(ref Result, matrix3);
            }...
        }
    }
}
```

Рис.9 Перемножение матриц

Представлены 4 проекта, реализованных в Visual Studio Common Eddition 2019.

Проекты представлены преподавателю в электронной форме, продемонстрирована их работоспособность, разъяснены детали программного кода.