

Практическая работа 33 Взаимодействие с MSWord

Цель занятия: Получить практический опыт разработки модулей для взаимодействия с MicrosoftOfficeWord

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio)
Microsoft Visual Studio 2010

Краткие теоретические сведения

Пример кода показывает, как сделать следующее:

- Вставка абзацев с текстом и форматированием.
- Просмотр и изменение диапазонов в рамках документа.
- Вставка таблиц, форматирование таблицы и заполнения таблицы данными.

Добавление диаграммы.

Чтобы создать новый документ Word с помощью VisualC# .NET, выполните следующие действия.

Запустите MicrosoftVisualStudio 2010. В меню файл выберите команду Создать и выберите проект. В области Типы проектов выберите Проекты VisualC#и в списке Шаблоны выберите Приложение Windows . По умолчанию будет создана форма Form1.

Добавьте ссылку на Библиотеку объектов MicrosoftWord. Чтобы сделать это, выполните следующие действия.

В меню проект выберите команду Добавить ссылку. На вкладке «СОМ» найдите Библиотеку объектов MicrosoftWord и нажмите кнопку выбрать.

Нажмите кнопку ОК в диалоговом окне AddReferences для подтверждения сделанного.

В меню Вид выберите панели инструментов, чтобы отобразить панель инструментов и добавьте кнопку в форму Form1.

Дважды щелкните кнопку Button1. Откроется окно кода для формы.

Добавьте следующую строку в конец списка директив using :

```
using Word = Microsoft.Office.Interop.Word;  
using System.Reflection;
```

Замените следующий код в окне кода

```
private void button1_Click(object sender, System.EventArgs e)  
{
```

```

}
Ha
private void button1_Click(object sender, System.EventArgs e)
{
    object oMissing = System.Reflection.Missing.Value;
    object oEndOfDoc = "\\endofdoc"; /* \endofdoc is a predefined
bookmark */

    //Start Word and create a new document.
    Word._Application oWord;
    Word._Document oDoc;
    oWord = new Word.Application();
    oWord.Visible = true;
    oDoc = oWord.Documents.Add(ref oMissing, ref oMissing,
        ref oMissing, ref oMissing);
    //Insert a paragraph at the beginning of the document.
    Word.Paragraph oPara1;
    oPara1 = oDoc.Content.Paragraphs.Add(ref oMissing);
    oPara1.Range.Text = "Heading 1";
    oPara1.Range.Font.Bold = 1;
    oPara1.Format.SpaceAfter = 24; //24 pt spacing after paragraph.
    oPara1.Range.InsertParagraphAfter();

    //Insert a paragraph at the end of the document.
    Word.Paragraph oPara2;
    object oRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
    oPara2 = oDoc.Content.Paragraphs.Add(ref oRng);
    oPara2.Range.Text = "Heading 2";
    oPara2.Format.SpaceAfter = 6;
    oPara2.Range.InsertParagraphAfter();

    //Insert another paragraph.
    Word.Paragraph oPara3;
    oRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
    oPara3 = oDoc.Content.Paragraphs.Add(ref oRng);
    oPara3.Range.Text = "This is a sentence of normal text. Now here is a
table:";

    oPara3.Range.Font.Bold = 0;
    oPara3.Format.SpaceAfter = 24;
    oPara3.Range.InsertParagraphAfter();

    //Insert a 3 x 5 table, fill it with data, and make the first row
//bold and italic.
    Word.Table oTable;

```

```

        Word.Range wrdRng = oDoc.Bookmarks.get_Item(ref
oEndOfDoc).Range;
        oTable = oDoc.Tables.Add(wrdRng, 3, 5, ref oMissing, ref oMissing);
        oTable.Range.ParagraphFormat.SpaceAfter = 6;
        int r, c;
        string strText;
        for(r = 1; r <= 3; r++)
            for(c = 1; c <= 5; c++)
                {
                    strText = "r" + r + "c" + c;
                    oTable.Cell(r, c).Range.Text = strText;
                }
        oTable.Rows[1].Range.Font.Bold = 1;
        oTable.Rows[1].Range.Font.Italic = 1;

//Add some text after the table.
Word.Paragraph oPara4;
oRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
oPara4 = oDoc.Content.Paragraphs.Add(ref oRng);
oPara4.Range.InsertParagraphBefore();
oPara4.Range.Text = "And here's another table:";
oPara4.Format.SpaceAfter = 24;
oPara4.Range.InsertParagraphAfter();

//Insert a 5 x 2 table, fill it with data, and change the column widths.
wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
oTable = oDoc.Tables.Add(wrdRng, 5, 2, ref oMissing, ref oMissing);
oTable.Range.ParagraphFormat.SpaceAfter = 6;
for(r = 1; r <= 5; r++)
    for(c = 1; c <= 2; c++)
        {
            strText = "r" + r + "c" + c;
            oTable.Cell(r, c).Range.Text = strText;
        }
oTable.Columns[1].Width = oWord.InchesToPoints(2); //Change
width of columns 1 & 2
oTable.Columns[2].Width = oWord.InchesToPoints(3);

//Keep inserting text. When you get to 7 inches from top of the
//document, insert a hard page break.
object oPos;
double dPos = oWord.InchesToPoints(7);
oDoc.Bookmarks.get_Item(ref
oEndOfDoc).Range.InsertParagraphAfter();
do

```

```

    {
        wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
        wrdRng.ParagraphFormat.SpaceAfter = 6;
        wrdRng.InsertAfter("A line of text");
        wrdRng.InsertParagraphAfter();
        oPos = wrdRng.get_Information
            (Word.WdInformation.wdVerticalPositionRelativeToPage);
    }
    while(dPos >= Convert.ToDouble(oPos));
    object oCollapseEnd = Word.WdCollapseDirection.wdCollapseEnd;
    object oPageBreak = Word.WdBreakType.wdPageBreak;
    wrdRng.Collapse(ref oCollapseEnd);
    wrdRng.InsertBreak(ref oPageBreak);
    wrdRng.Collapse(ref oCollapseEnd);
    wrdRng.InsertAfter("We're now on page 2. Here's my chart:");
    wrdRng.InsertParagraphAfter();

    //Insert a chart.
    Word.InlineShape oShape;
    object oClassType = "MSGraph.Chart.8";
    wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
    oShape = wrdRng.InlineShapes.AddOLEObject(ref oClassType, ref
oMissing,
        ref oMissing, ref oMissing, ref oMissing,
        ref oMissing, ref oMissing, ref oMissing);

    //Demonstrate use of late bound oChart and oChartApp objects to
    //manipulate the chart object with MSGraph.
    object oChart;
    object oChartApp;
    oChart = oShape.OLEFormat.Object;
    oChartApp = oChart.GetType().InvokeMember("Application",
        BindingFlags.GetProperty, null, oChart, null);

    //Change the chart type to Line.
    object[] Parameters = new Object[1];
    Parameters[0] = 4; //xlLine = 4
    oChart.GetType().InvokeMember("ChartType",
BindingFlags.SetProperty,
        null, oChart, Parameters);

    //Update the chart image and quit MSGraph.
    oChartApp.GetType().InvokeMember("Update",
        BindingFlags.InvokeMethod, null, oChartApp, null);
    oChartApp.GetType().InvokeMember("Quit",

```

```

        BindingFlags.InvokeMethod, null, oChartApp, null);
//... If desired, you can proceed from here using the Microsoft Graph
//Object model on the oChart and oChartApp objects to make
additional
//changes to the chart.

//Set the width of the chart.
oShape.Width = oWord.InchesToPoints(6.25f);
oShape.Height = oWord.InchesToPoints(3.57f);

//Add text after the chart.
wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
wrdRng.InsertParagraphAfter();
wrdRng.InsertAfter("THE END.");

//Close this form.
this.Close();
}

```

Нажмите клавишу F5 для построения и запуска программы.

Нажмите кнопку Button1 для запуска автоматизации Word и создать документ.

После завершения выполнения, изучите документ, который был создан. Документ содержит две страницы отформатированные абзацы, таблицы и диаграммы.

Использование шаблона

При использовании автоматизации для создания документов, которые находятся в общем формате, можно воспользоваться преимуществом начиная с нового документа, основанного на предварительно отформатированного шаблона процесса.

Можно иметь больший контроль над форматированием и размещением объектов в документе.

Для создания документа с меньшим объемом кода.

С помощью шаблона, то можно точная настройка положения таблиц, абзацев и других объектов в документе, а также включать форматирование для этих объектов.

```

object oTemplate = "c:\\MyTemplate.dot";
oDoc = oWord.Documents.Add(ref oTemplate, ref oMissing,
    refoMissing, refoMissing);

```

В шаблоне можно определить закладки, чтобы клиент автоматизации мог заполнить текст в определенном месте в документе, как показано ниже:

```

object oBookMark = "MyBookmark";
oDoc.Bookmarks.Item(ref oBookMark).Range.Text = "Some Text Here";

```

Другое преимущество использования шаблона в том, что можно создавать и сохранять стили форматирования, которые вы хотите применить во время выполнения, как показано ниже:

```
object oStyleName = "MyStyle";  
oDoc.Bookmarks.Item(ref oBookMark).Range.set_Style(ref oStyleName);  
- или -  
object oStyleName = "MyStyle";  
oWord.Selection.set_Style(ref oStyleName);
```

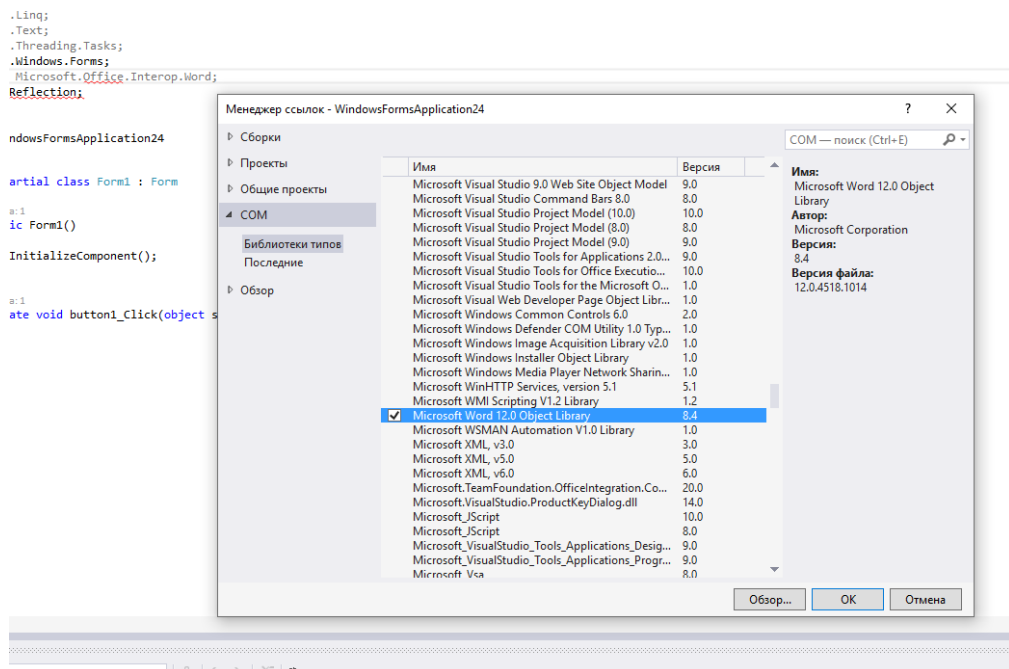
Задания

- 1 Изучить теоретические сведения и задание к работе
- 2 В соответствии с вариантом задания разработать отлаженный модуль.

Порядок выполнения работы

Для работы с Ms Word необходимо подключить библиотеки
using Word = Microsoft.Office.Interop.Word;
using System.Reflection;

Кроме того, подключить ссылку на библиотеку (добавить ссылку, перейти в раздел COM, там найти нужную библиотеку:



Чтобы работать с текстовым файлом, сначала открывается сам Word, а затем файл, аналогично в программе сначала объявляем приложение:

```
Word._Application appw;  
appw = new Word.Application();  
appw.Visible = true;
```

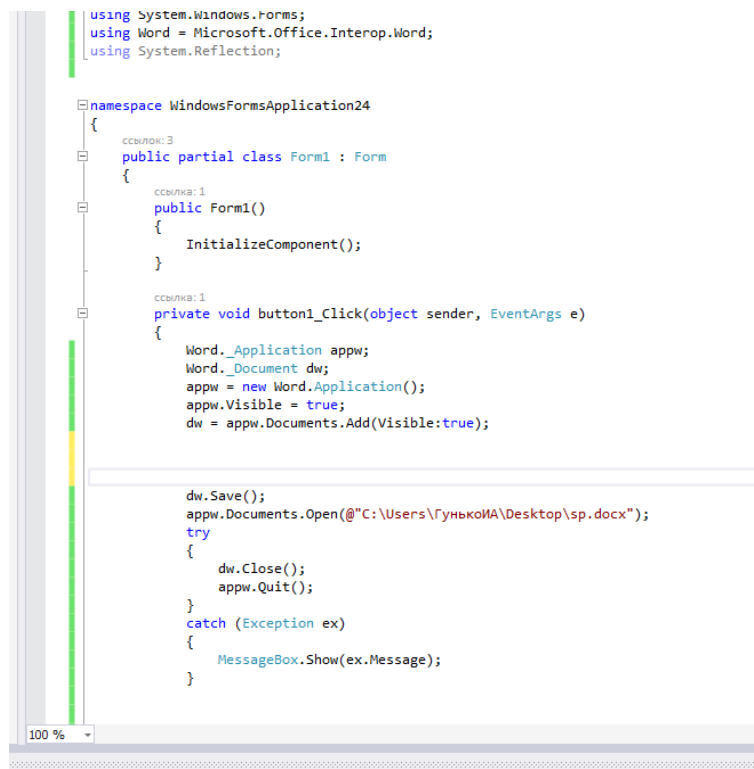
а затем документ

```
Word._Document dw;  
dw = appw.Documents.Add(Visible:true);
```

После выполнения действий с текстовым файлом **ОБЯЗАТЕЛЬНО** необходимо закрыть документ и приложение, иначе они остаются запущенными в фоновом режиме и при попытке работать с ними снова, являются недоступными:

```
try  
{  
    dw.Close();  
    appw.Quit();  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
}
```

Создание и открытие пустого текстового файла:



```
using System.Windows.Forms;  
using Word = Microsoft.Office.Interop.Word;  
using System.Reflection;  
  
namespace WindowsFormsApplication24  
{  
    ссылка:3  
    public partial class Form1 : Form  
    {  
        ссылка:1  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        ссылка:1  
        private void button1_Click(object sender, EventArgs e)  
        {  
            Word._Application appw;  
            Word._Document dw;  
            appw = new Word.Application();  
            appw.Visible = true;  
            dw = appw.Documents.Add(Visible:true);  
  
            dw.Save();  
            appw.Documents.Open(@"C:\Users\ГуньяковИА\Desktop\sp.docx");  
            try  
            {  
                dw.Close();  
                appw.Quit();  
            }  
            catch (Exception ex)  
            {  
                MessageBox.Show(ex.Message);  
            }  
        }  
    }  
}
```

Перед сохранением файла (`dw.Save();`) вставим код создания таблицы в документ, например, таблицу 2×2 :

```
Обозреватель объектов Form1.cs Form1.cs [Конструктор]
WindowsFormsApplication24 WindowsFormsApplication24.Form1

Word._Application appw;
Word._Document dw;
appw = new Word.Application();
appw.Visible = true;
dw = appw.Documents.Add(Visible:true);

// таблица
var Paragraph = appw.ActiveDocument.Paragraphs.Add();
//Получаем диапазон
var tableRange = Paragraph.Range;
//Добавляем таблицу в указанный диапазон
appw.ActiveDocument.Tables.Add(tableRange, 2, 2);

//Границы таблицы
var table = appw.ActiveDocument.Tables[appw.ActiveDocument.Tables.Count];
table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
table.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

//Шанка таблицы

dw.Save();
appw.Documents.Open(@"C:\Users\ГуныкойА\Desktop\sp.docx");
try
{
    dw.Close();
    appw.Quit();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
```

Создадим таблицу для примера библиотеки, для этого изменим ориентацию листа, создадим 8 колонок, определим их ширину, добавим шапку таблицы:

```
Form1.cs [Конструктор] Обозреватель объектов
wsFormsApplication24 WindowsFormsApplication24.Form1

dw = appw.Documents.Add(Visible:true);

// Ориентация страницы
dw.PageSetup.Orientation = Word.WdOrientation.wdOrientLandscape;

// таблица
var Paragraph = appw.ActiveDocument.Paragraphs.Add();
var tableRange = Paragraph.Range;
appw.ActiveDocument.Tables.Add(tableRange, 4, 8);

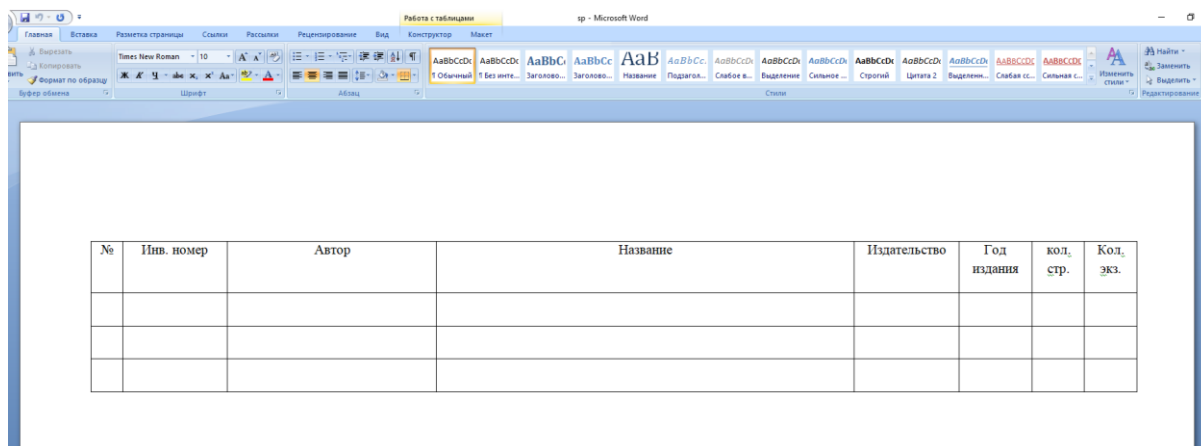
//Границы таблицы
var table = appw.ActiveDocument.Tables[appw.ActiveDocument.Tables.Count];
table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
table.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

//Свойства таблицы
table.Range.Font.Size = 10;
table.Columns[1].Width = appw.PixelsToPoints(30f);
table.Columns[2].Width = appw.PixelsToPoints(100f);
table.Columns[3].Width = appw.PixelsToPoints(200f);
table.Columns[4].Width = appw.PixelsToPoints(400f);
table.Columns[5].Width = appw.PixelsToPoints(100f);
table.Columns[6].Width = appw.PixelsToPoints(70f);
table.Columns[7].Width = appw.PixelsToPoints(50f);
table.Columns[8].Width = appw.PixelsToPoints(50f);
table.Range.Font.Name = "Times New Roman";
table.Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Шанка таблицы
table.Cell(1, 1).Range.Text = "№";
table.Cell(1, 2).Range.Text = "Инв. номер";
table.Cell(1, 3).Range.Text = "Автор";
table.Cell(1, 4).Range.Text = "Название";
table.Cell(1, 5).Range.Text = "Издательство";
table.Cell(1, 6).Range.Text = "Год издания";
table.Cell(1, 7).Range.Text = "кол. стр.";
table.Cell(1, 8).Range.Text = "Кол. экз.";

dw.Save();
appw.Documents.Open(@"C:\Users\ГуныкойА\Desktop\sp.docx");
```


Получилась таблица:



№	Инв. номер	Автор	Название	Издательство	Год издания	кол. стр.	Кол. экз.

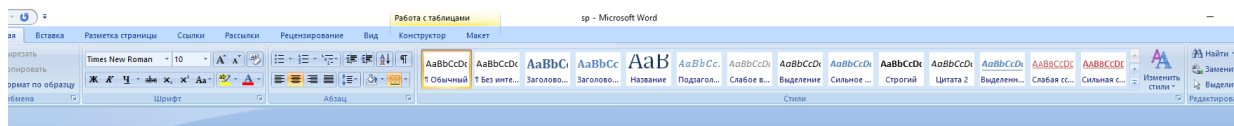
Осталось внести в нее данные. Информация из текстового файла собрана в коллекцию (практическая 32), осталось только перенести элементы этой коллекции в соответствующие ячейки таблицы, причем для текстовых данных изменить выравнивание ячеек по левому краю.

```
formsApplication24 | windowsFormsApplication24.Form1 | button1_Click(01
table.Cell(1, 6).Range.Text = "год издания";
table.Cell(1, 7).Range.Text = "кол. стр.";
table.Cell(1, 8).Range.Text = "Кол. экз.";

int i = 0;
foreach (Book bb in books)
{
    table.Cell(i + 2, 1).Range.Text = (i+1).ToString();
    table.Cell(i + 2, 1).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;
    table.Cell(i + 2, 2).Range.Text = bb.innum;
    table.Cell(i + 2, 2).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;
    table.Cell(i + 2, 3).Range.Text = bb.author;
    table.Cell(i + 2, 3).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;
    table.Cell(i + 2, 4).Range.Text = bb.name;
    table.Cell(i + 2, 4).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;
    table.Cell(i + 2, 5).Range.Text = bb.publishing;
    table.Cell(i + 2, 5).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;
    table.Cell(i + 2, 6).Range.Text = bb.year.ToString();
    table.Cell(i + 2, 7).Range.Text = bb.quantity.ToString();
    table.Cell(i + 2, 8).Range.Text = bb.nal.ToString();
    i++;
}

dw.Save();
appw.Documents.Open(@"C:\Users\ГуныкoиA\Desktop\sp.docx");
try
{
    dw.Close();
    appw.Quit();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Получим



№	Инв. номер	Автор	Название	Издательство	Год издания	кол. стр.	Кол. экз.
1	0000000001	Фленов М.	Библия С#	БХВ-Петербург	2009	547	3
2	0000000002	Васильев А.Н.	Программирование на С# для начинающих_Особенности языка	Эксмо	2019	526	5
3	0000000003	Семакин И.Г., Шестаков А.П.	Основы программирования	Академия	2008	431	15
4	0000000004	Малов В.	Тайны знаменитых пиратов	Оннкс	2005	255	1

Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Вывод

Варианты заданий

Варианты заданий представлены в практической работе 31. Осуществите сохранение содержимого таблицы в документе Word.

Используемая литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.
2. Шарп Джон Ш26 Microsoft Visual С#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.
3. Васильев А.Н. Программирование на С# для начинающих. Основные сведения. – Москва: Эксмо, 2018.
4. Васильев А.Н. Программирование на С# для начинающих. Особенности языка. – Москва: Эксмо, 2019.
5. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.