

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**

**Лабораторная работа №5**

**по дисциплине** «**Методы программирования**»

«Автоматизация Word»

**Выполнил:** ст. гр. ТКИ-342

Речкалов Р.Ю.

**Проверил:** к.т.н. Сафронов А.И.

**Москва – 2024 г**

# **1. Цель работы**

Освоить навыки подключения внешних модулей (библиотек классов) и их использования в разрабатываемых программах; освоить навыки программируемой настройки параметров документов *Microsoft Office Word*.

# **2. Формулировка задачи**

1. Создать и настроить графический пользовательский интерфейс, взаимодействующий с текстовым файлом (*\*.txt / \*.csv*), а также с модулем (библиотекой классов) текстового редактора *Microsoft Office Word*.

2. Создать текстовый файл. Текстовый файл исходно пуст, но нацелен на хранение *M* строк с разделителями вида:

|  |  |
| --- | --- |
| *param1;param2;param3;…;paramN* | (1) |

Каждая строка – набор значимых параметров, записанных в порядке, предусмотренных автором-разработчиком графического пользовательского интерфейса. Количество параметров *N* так же определяется автором-разработчиком.

Количество строк *M* определяется количеством абзацев на титульном листе отчёта по выполненной работе в формате *Microsoft Office Word*. Например:

а. Наименование ведомства (Министерство транспорта Российской Федерации);

б. Регалии вуза (Федеральное государственное автономное образовательное учреждение высшего образования);

в. Наименование вуза («Российский университет транспорта» (РУТ (МИИТ));

г. Наименование института (Институт транспортной техники и систем управления);

д. Наименование кафедры (Кафедра «Управление и защита информации»);

е. Вид документа (Отчёт / Реферат / Курсовая работа / Курсовой проект);

ж. Вариация связки (по практике / по);

и. Вид учебного занятия (Учебной практике / Производственной практике / Лабораторной работе / Практическому занятию / Индивидуальному заданию);

к. Вариация связки, если необходима (по дисциплине);

л. Наименование дисциплины, если необходимо («Алгоритмизация и технологии программирования» / «Методы программирования»);

м. Вариация связки, если необходима (на тему / представлено решение задачи);

н. Строка «Выполнил: ст. гр.» с указанием номера учебной группы;

о. Ф.И.О. автора-составителя документа;

п. Номер варианта;

р. Строка «Проверил: » с указанием регалий и Ф.И.О. проверяющего лица.

с. Строка формата «Город – год».

Например, *param1* – строка текста, выводимая в абзац документа *Microsoft Office Word; param2* – семейство/наименование шрифта (*Times New Roman*); *param3* – размер шрифта (15 пт) и так далее.

3. При загрузке графического пользовательского интерфейса все параметры из файла считываются в строковый массив, если файл не пуст, и изменяют состояние интерфейсных элементов управления, размещённых на графическом пользовательском интерфейсе. Если файл пуст, то все интерфейсные элементы управления графического пользовательского интерфейса остаются в исходном, нулевом состоянии.

4. Должна быть предусмотрена экранная кнопка «Сохранить», записывающая в файл выставленные или изменённые на графическом пользовательском интерфейсе значения параметров.

5. Должна быть предусмотрена кнопка «Создать», инициирующая генерацию титульного листа в формате *Microsoft Office Word.*

6. Должен быть предусмотрен предварительный просмотр документа, планируемого к выгрузке (рассмотреть вариант использования интерфейсного элемента *PictureBox* для этих целей).

7. Документ, представленный иллюстрацией, должен быть первично набран в *Microsoft Office Word* вручную. Он запланирован в качестве тестового примера в данной работе.

8. Все заполнители в работе выполняются при помощи «кареток»  [, , ] (*Tab Stops*) и табуляции [https://prosto-kod.ru/17_01_2023/4.jpg] (*Tab*).

9. Все неявные колоночные выравнивания допустимо выполнять необрамлёнными таблицами (таблицами без визуальных границ).

11. Между названием вуза и наименованием кафедры должен иметь место интерфейсный элементов управления заполнитель типа «нижнее подчёркивание», выполненный посредством табуляции.

12. Предоставить пользователю дополнительный комбинированный список, задающий семейство шрифтов (*Calibri*, *Times New Roman*, *Courier New* и др.), влияющий абсолютно на все элементы, размещённые на титульном листе. При необходимости ввести поправки на позиционирование элементов для различных шрифтов.

13. Сгенерировать файл, описываемый изображением



Рисунок 1 - Изображение индивидуального задания

# **3. Составление диаграммы классов, входящих в состав решения.**

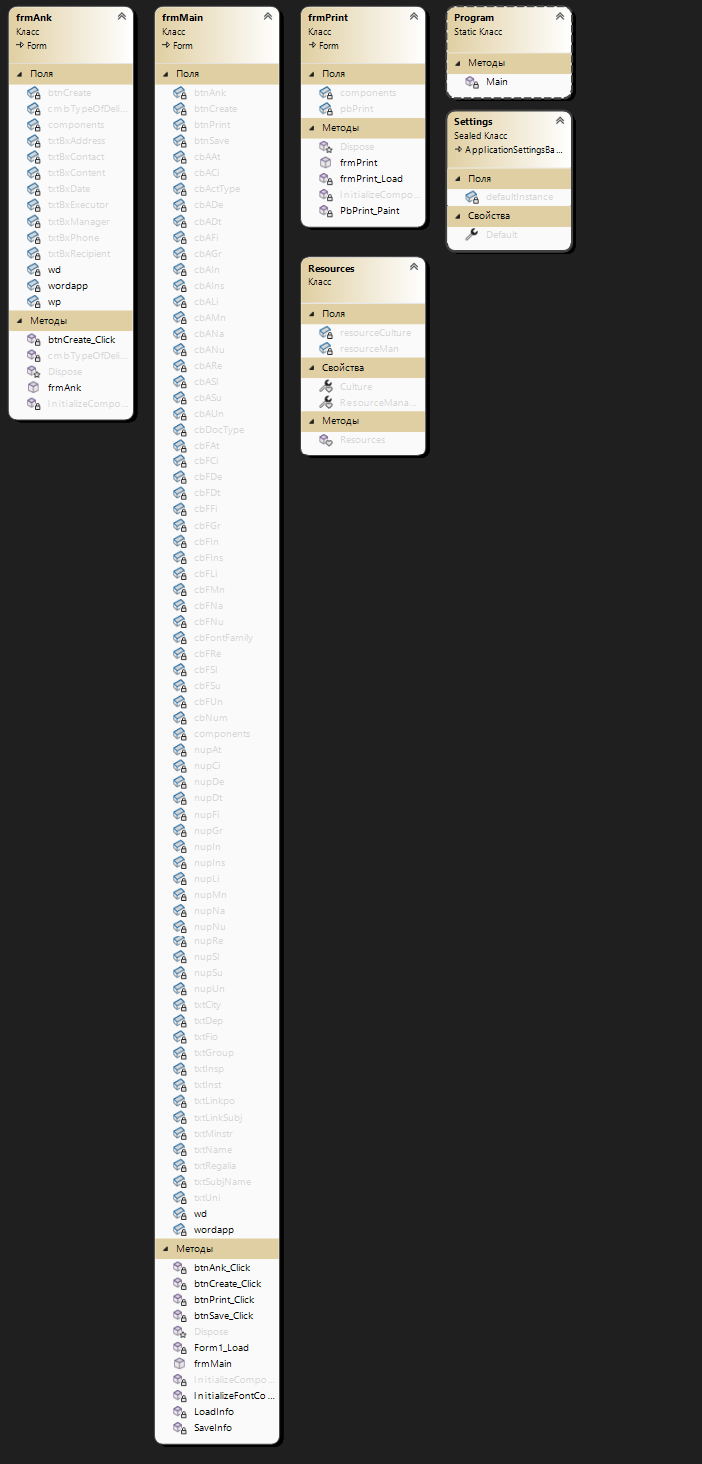


Рисунок 2 – Диаграмма классов

# **4. Составление сети Петри запрограммированного технологического процесса.**

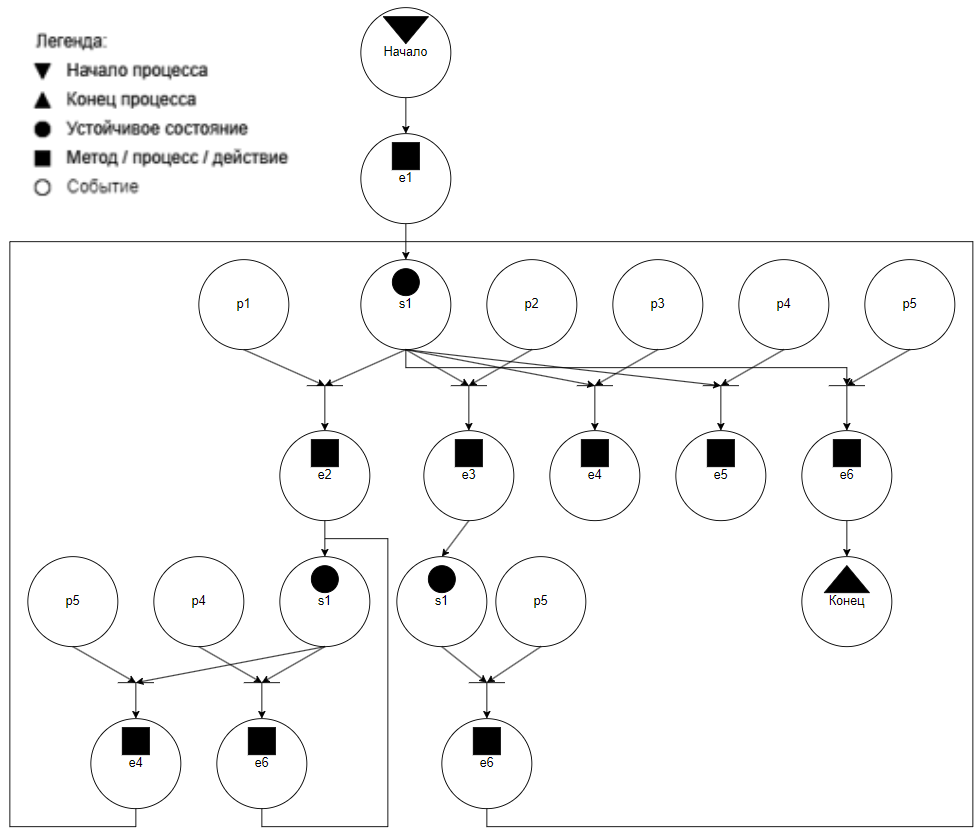


Рисунок 3 - Сеть Петри основной формы

**Описание сети Петри**

- состояния (states)

s1 – форма ожидает действий пользователя

- действия (effects)

e1 – информация из текстового файла загружается в поля формы

e2 – открытие формы анкеты

e3 – открытие формы предпросмотра

e4 – создание файла Word

e5 – сохранение полей формы в текстовый файл

e6 – закрытие формы

- события (prompts)

p1 – нажата кнопка Анкета

p2 – нажата кнопка Предпросмотр

p3 – нажата кнопка Сохранить

p4 – нажата кнопка Создать

p5 – пользователь закрывает форму

# **5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ).**

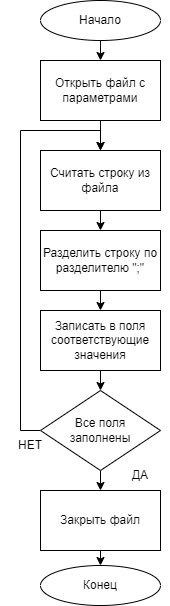


Рисунок 4 – Алгоритм загрузки параметров из файла в форму

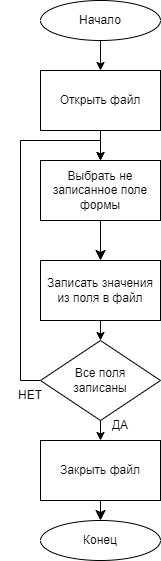


Рисунок 5 - Алгоритм записи информации в файл

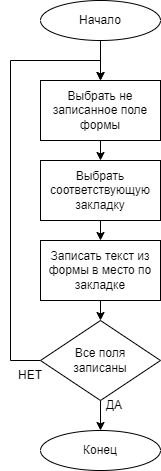
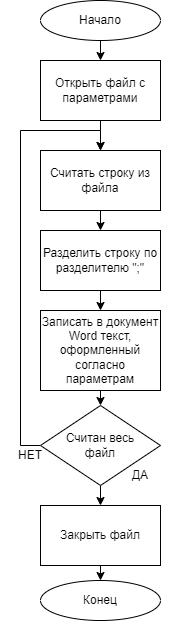


Рисунок 6 – Создание файла Word (титульный лист – слева; анкета – справа)

# **6. Подбор тестовых примеров.**

Тестовые примеры:

* Проверить генерацию титульного листа
* Проверить генерацию анкеты
* Проверить выбор разреженности текста
* Проверить предпросмотр
* Изменить оформление титультного листа

**7. Листинг (код) составленного программного обеспечения.**

Основная форма:

using System;

using System.Windows.Forms;

using System.IO;

using Word = Microsoft.Office.Interop.Word;

using System.Drawing;

namespace Task5

{

public partial class frmMain : Form

{

private Word.Application wordapp;

private Word.Document wd;

public frmMain()

{

InitializeComponent();

InitializeFontComboBox();

}

private void InitializeFontComboBox()

{

cbFontFamily.Items.AddRange(new string[] { "Calibri", "Times New Roman", "Courier New", "Arial", "Verdana" });

cbFontFamily.SelectedIndex = 1; // Устанавливаем "Times New Roman" как значение по умолчанию

}

private void Form1\_Load(object sender, EventArgs e)

{

this.cbActType.SelectedIndex = 0;

this.cbDocType.SelectedIndex = 0;

this.cbNum.SelectedIndex = 0;

this.LoadInfo();

}

private void btnCreate\_Click(object sender, EventArgs e)

{

wordapp = new Word.Application();

wd = wordapp.Documents.Add(Visible: true);

wd = wordapp.Documents.Add();

StreamReader sr = new StreamReader(Application.StartupPath + "\\TextInfo.txt");

string temp;

string[] atemp;

string selectedFont = cbFontFamily.SelectedItem.ToString(); // Получаем выбранный шрифт

wd.Content.Font.Size = 14;

wd.Content.Font.Color = 0;

wd.Content.Font.Name = selectedFont; // Устанавливаем выбранный шрифт для всего документа

while ((temp = sr.ReadLine()) != null)

{

atemp = temp.Split(';');

wordapp.Selection.Font.Name = selectedFont; // Устанавливаем выбранный шрифт для каждого параграфа

wordapp.Selection.Font.Size = int.Parse(atemp[3]);

wordapp.Selection.ParagraphFormat.Alignment = (Word.WdParagraphAlignment)int.Parse(atemp[4]);

wordapp.Selection.TypeText(atemp[0]);

if (atemp[1] == "1")

{

wordapp.Selection.TypeParagraph();

}

}

wordapp.Selection.HomeKey(Word.WdUnits.wdStory, Word.WdMovementType.wdMove);

wordapp.Selection.MoveDown(Word.WdUnits.wdParagraph, 4, Word.WdMovementType.wdMove);

wordapp.Selection.ParagraphFormat.TabStops.Add(wordapp.CentimetersToPoints(16.5f), Word.WdAlignmentTabAlignment.wdRight, Word.WdTabLeader.wdTabLeaderLines);

wordapp.Selection.TypeText("\t");

wordapp.Selection.TypeParagraph();

sr.Close();

wd.Save();

wordapp.Quit();

}

private void btnSave\_Click(object sender, EventArgs e)

{

this.SaveInfo();

}

private void btnPrint\_Click(object sender, EventArgs e)

{

frmPrint fp = new frmPrint();

fp.Show();

}

private void LoadInfo()

{

StreamReader sr = new StreamReader(Application.StartupPath + "\\TextInfo.txt");

string[] args = sr.ReadLine().Split(';');

this.txtMinstr.Text = args[0].Trim();

this.cbFMn.SelectedItem = args[2];

this.nupMn.Value = decimal.Parse(args[3]);

this.cbAMn.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtRegalia.Text = args[0].Trim();

this.cbFRe.SelectedItem = args[2];

this.nupRe.Value = decimal.Parse(args[3]);

this.cbARe.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtUni.Text = args[0].Trim();

this.cbFUn.SelectedItem = args[2];

this.nupUn.Value = decimal.Parse(args[3]);

this.cbAUn.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtInst.Text = args[0].Trim();

this.cbFIn.SelectedItem = args[2];

this.nupIn.Value = decimal.Parse(args[3]);

this.cbAIn.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtDep.Text = args[0].Trim();

this.cbFDe.SelectedItem = args[2];

this.nupDe.Value = decimal.Parse(args[3]);

this.cbADe.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.cbDocType.Text = args[0].Trim();

this.cbFDt.SelectedItem = args[2];

this.nupDt.Value = decimal.Parse(args[3]);

this.cbADt.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtLinkpo.Text = args[0].Trim();

this.cbFLi.SelectedItem = args[2];

this.nupLi.Value = decimal.Parse(args[3]);

this.cbALi.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.cbActType.Text = args[0].Trim();

this.cbFAt.SelectedItem = args[2];

this.nupAt.Value = decimal.Parse(args[3]);

this.cbAAt.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.cbNum.Text = args[0].Trim();

this.cbFNu.SelectedItem = args[2];

this.nupNu.Value = decimal.Parse(args[3]);

this.cbANu.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtName.Text = args[0].Trim();

this.cbFNa.SelectedItem = args[2];

this.nupNa.Value = decimal.Parse(args[3]);

this.cbANa.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtLinkSubj.Text = args[0].Trim();

this.cbFSl.SelectedItem = args[2];

this.nupSl.Value = decimal.Parse(args[3]);

this.cbASl.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtSubjName.Text = args[0].Trim();

this.cbFSu.SelectedItem = args[2];

this.nupSu.Value = decimal.Parse(args[3]);

this.cbASu.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtGroup.Text = "Выполнил: ст. гр. ТКИ-342";

this.cbFGr.SelectedItem = args[2];

this.nupGr.Value = decimal.Parse(args[3]);

this.cbAGr.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtFio.Text = args[0].Trim();

this.cbFFi.SelectedItem = args[2];

this.nupFi.Value = decimal.Parse(args[3]);

this.cbAFi.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtInsp.Text = args[0].Trim();

this.cbFIns.SelectedItem = args[2];

this.nupIns.Value = decimal.Parse(args[3]);

this.cbAIns.SelectedIndex = int.Parse(args[4]);

args = sr.ReadLine().Split(';');

this.txtCity.Text = args[0].Trim();

this.cbFCi.SelectedItem = args[2];

this.nupCi.Value = decimal.Parse(args[3]);

this.cbACi.SelectedIndex = int.Parse(args[4]);

sr.Close();

}

private void SaveInfo()

{

StreamWriter sw = new StreamWriter(Application.StartupPath + "\\TextInfo.txt");

sw.WriteLine(this.txtMinstr.Text+";1;"+this.cbFMn.SelectedItem+";"+this.nupMn.Value+";"+this.cbAMn.SelectedIndex);

sw.WriteLine(this.txtRegalia.Text + ";1;" + this.cbFRe.SelectedItem + ";" + this.nupRe.Value + ";" + this.cbARe.SelectedIndex);

sw.WriteLine(this.txtUni.Text + ";1;" + this.cbFUn.SelectedItem + ";" + this.nupUn.Value + ";" + this.cbAUn.SelectedIndex);

sw.WriteLine(this.txtInst.Text + ";1;" + this.cbFIn.SelectedItem + ";" + this.nupIn.Value + ";" + this.cbAIn.SelectedIndex);

sw.WriteLine(this.txtDep.Text + ";1;" + this.cbFDe.SelectedItem + ";" + this.nupDe.Value + ";" + this.cbADe.SelectedIndex);

sw.WriteLine(this.cbDocType.Text + ";0;" + this.cbFDt.SelectedItem + ";" + this.nupDt.Value + ";" + this.cbADt.SelectedIndex);

sw.WriteLine(" " + this.txtLinkpo.Text + ";0;" + this.cbFLi.SelectedItem + ";" + this.nupLi.Value + ";" + this.cbALi.SelectedIndex);

sw.WriteLine(" " + this.cbActType.Text + ";0;" + this.cbFAt.SelectedItem + ";" + this.nupAt.Value + ";" + this.cbAAt.SelectedIndex);

sw.WriteLine(" " + this.cbNum.Text + ";1;" + this.cbFNu.SelectedItem + ";" + this.nupNu.Value + ";" + this.cbANu.SelectedIndex);

sw.WriteLine(this.txtName.Text + ";1;" + this.cbFNa.SelectedItem + ";" + this.nupNa.Value + ";" + this.cbANa.SelectedIndex);

sw.WriteLine(this.txtLinkSubj.Text + ";0;" + this.cbFSl.SelectedItem + ";" + this.nupSl.Value + ";" + this.cbASl.SelectedIndex);

sw.WriteLine(" " + this.txtSubjName.Text + ";1;" + this.cbFSu.SelectedItem + ";" + this.nupSu.Value + ";" + this.cbASu.SelectedIndex);

sw.WriteLine(this.txtGroup.Text + ";1;" + this.cbFGr.SelectedItem + ";" + this.nupGr.Value + ";" + this.cbAGr.SelectedIndex);

sw.WriteLine(this.txtFio.Text + ";1;" + this.cbFFi.SelectedItem + ";" + this.nupFi.Value + ";" + this.cbAFi.SelectedIndex);

sw.WriteLine(this.txtInsp.Text + ";1;" + this.cbFIns.SelectedItem + ";" + this.nupIns.Value + ";" + this.cbAIns.SelectedIndex);

sw.Write(this.txtCity.Text + ";1;" + this.cbFCi.SelectedItem + ";" + this.nupCi.Value + ";" + this.cbACi.SelectedIndex);

sw.Close();

}

private void btnAnk\_Click(object sender, EventArgs e)

{

frmAnk fa = new frmAnk();

fa.Show();

}

}

}

Форма предпросмотра:

using System;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

namespace Task5

{

public partial class frmPrint : Form

{

public frmPrint()

{

InitializeComponent();

}

private void frmPrint\_Load(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Maximized;

this.pbPrint.Paint += PbPrint\_Paint;

}

private void PbPrint\_Paint(object sender, PaintEventArgs e)

{

Brush br = Brushes.Black;

StringFormat sf = new StringFormat();

StreamReader sr = new StreamReader(Application.StartupPath + "\\TextInfo.txt");

string temp;

string[] atemp;

string buffer = "";

int vert\_marg = 0;

int hor\_marg = 0;

while ((temp = sr.ReadLine()) != null)

{

atemp = temp.Split(';');

buffer += atemp[0];

if (atemp[1] == "1")

{

sf.Alignment = (StringAlignment)int.Parse(atemp[4]);

e.Graphics.DrawString(buffer, new Font(atemp[2], int.Parse(atemp[3])), br, new RectangleF(hor\_marg, vert\_marg, 1920, 20), sf);

buffer = "";

vert\_marg +=20;

}

}

sr.Close();

}

}

}

Форма анкеты:

using System;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace Task5

{

public partial class frmAnk : Form

{

private Word.Application wordapp;

private Word.Document wd;

private Word.Paragraph wp;

public frmAnk()

{

InitializeComponent();

}

private void btnCreate\_Click(object sender, EventArgs e)

{

wordapp = new Word.Application();

wd = wordapp.Documents.Add(Visible: true);

wp = wd.Paragraphs.Add();

// Center-aligned title

wp.Range.Font.Bold = 1;

wp.Range.Font.Size = 14;

wp.Range.Font.Name = "Times New Roman";

wp.Range.Text = "ЗАЯВКА\nна отправку почтовой корреспонденции";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

wp.Range.InsertParagraphAfter();

// Вид отправки

wp.Range.Font.Bold = 0;

wp.Range.Font.Size = 14;

wp.Range.Text = "Вид отправки: " + cmbTypeOfDelivery.SelectedItem.ToString();

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Font.Size = 8;

wp.Range.Text = "(экспресс-почта, простое, заказное/с уведомлением, ценное/с уведомлением, 1 класс/с уведомлением)";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

wp.Range.InsertParagraphAfter();

// 1. Получатель

wp.Range.Font.Size = 14;

wp.Range.Text = "1. Получатель: " + txtBxRecipient.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Font.Size = 8;

wp.Range.Text = "(наименование организации или Фамилия, имя, отчество физического лица)";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

wp.Range.InsertParagraphAfter();

// 2. Адрес получателя

wp.Range.Font.Size = 14;

wp.Range.Text = "2. Адрес получателя: " + txtBxAddress.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Font.Size = 8;

wp.Range.Text = "(индекс, город, область, дом, индекс)";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

wp.Range.InsertParagraphAfter();

// 3. Контактное лицо, телефон

wp.Range.Font.Size = 14;

wp.Range.Text = "3. Контактное лицо, телефон: " + txtBxContact.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Font.Size = 8;

wp.Range.Text = "(в случае отправки экспресс почтой)";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

wp.Range.InsertParagraphAfter();

// 4. Отправляемая корреспонденция

wp.Range.Font.Size = 14;

wp.Range.Text = "4. Отправляемая корреспонденция: " + txtBxContent.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

// 5. Дата подачи заявки исполнителем

wp.Range.Text = "5. Дата подачи заявки исполнителем: " + txtBxDate.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

// 6. Руководитель структурного подразделения

wp.Range.Text = "6. Руководитель структурного подразделения: " + txtBxManager.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Font.Size = 8;

wp.Range.Text = "Ф.И.О.";

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphRight;

wp.Range.InsertParagraphAfter();

// Footer

wp.Range.Font.Size = 14;

wp.Range.Text = "ИСП: " + txtBxExecutor.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wp.Range.InsertParagraphAfter();

wp.Range.Text = "тел.: " + txtBxPhone.Text;

wp.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

wd.Save();

}

}

}

# **8. Графический пользовательский интерфейс программного обеспечения и его описание.**

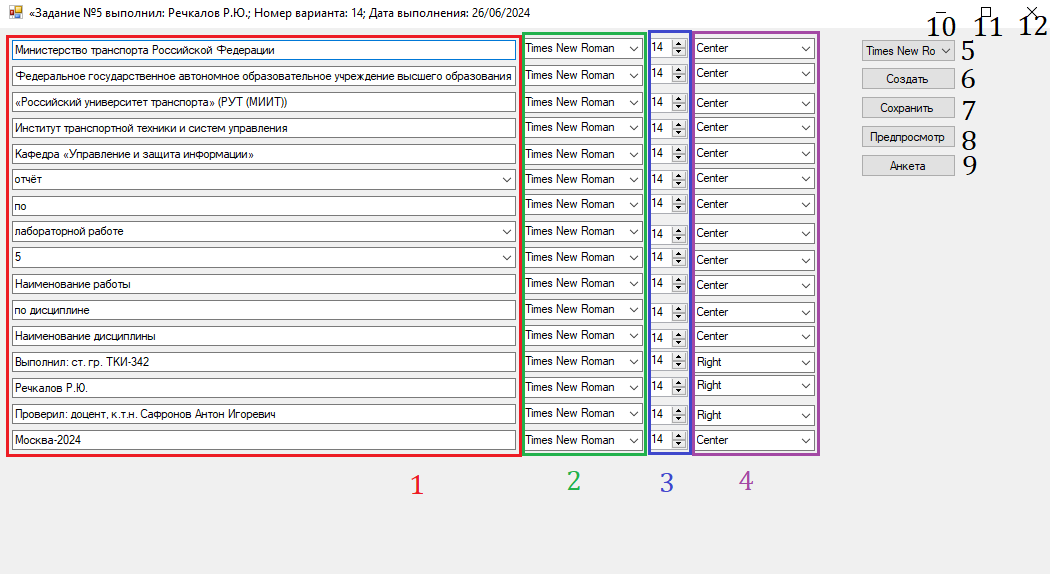


Рисунок 7 - Интерфейс основной формы

1. Текстовые поля/комбинированные списки для ввода/выбора текста
2. Комбинированные списки для выбора шрифта
3. Элементы NumericUpDown для выбора размера шрифта
4. Комбинированные списки для выбора ориентации текста
5. Комбинированный список, задающий семейство шрифтов
6. Кнопка Создать
7. Кнопка Сохранить
8. Кнопка Предпросмотр
9. Кнопка Анкета
10. Кнопка «Свернуть»
11. Кнопка «Максимизировать/минимизировать»
12. Кнопка «Закрыть»

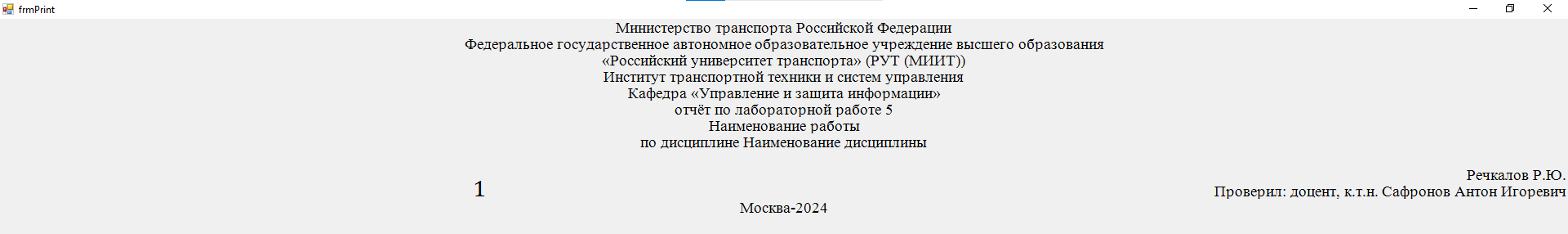


Рисунок 8 - Интерфейс формы предпросмотра

1. PictureBox для вывода текста

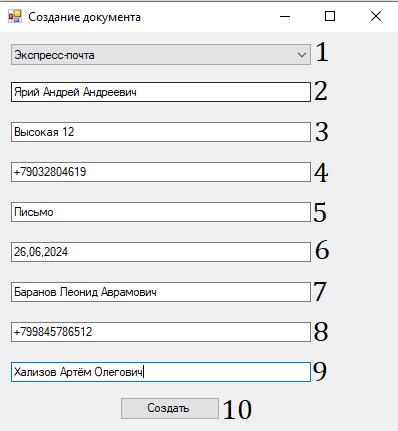


Рисунок 9 - Интерфейс формы анкеты

1. Комбинированный список для выбора вида отправки
2. Ввод получателя
3. Ввод адреса получателя
4. Ввод номера телефона контактного лица
5. Ввод отправляемой корреспонденции
6. Ввод даты подачи заявки исполнителем
7. Ввод руководителя структурного подразделения
8. Ввод номера телефона исполнителя
9. Ввод исполнителя
10. Кнопка «Закрыть»

# **9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.**

1. Заголовок главной формы должен содержать надпись вида: «Задание №5 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».

см. Рисунок 7

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы. Этот момент должен быть согласован с преподавателем, ведущим учебную дисциплину.



Рисунок 10 - Время на момент выполнения работы

3. Имя автора-разработчика графического пользовательского интерфейса вводится через свойства (*Properties*) экранной формы и не должно быть подвержено изменению/редактированию в коде. Это же имя должно быть использовано для вставки в строку Ф.И.О. автора-составителя на титульном листе.

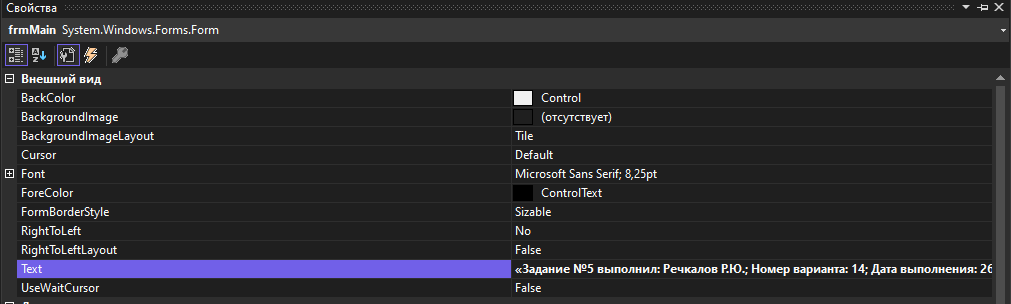


Рисунок 11 - Свойства основной формы

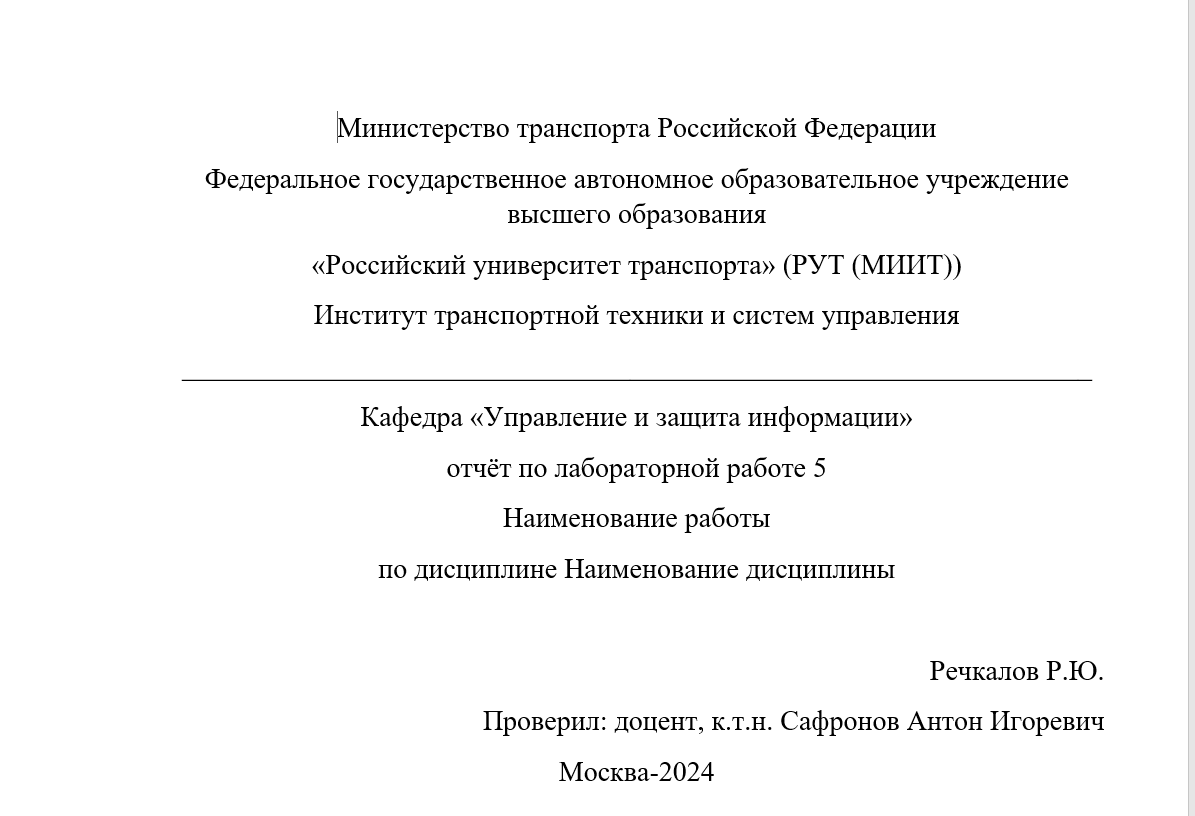


Рисунок 12 - Генерируемый программой титульный лист

4. Имя и должность лица, проверяющего работу, вводится либо через текстовое поле (*Text Box*), либо через комбинированный список (*Combo Box*), если условием индивидуального задания по варианту не предусмотрено иное.

см. Рисунок 7

5. Предусмотреть на графическом пользовательском интерфейсе комбинированный список (*Combo Box*), в котором выбирается вид отчётного документа:

- отчёт;

- реферат;

- эссе;

- курсовой проект;

- курсовая работа;

- доклад;

- домашнее задание.

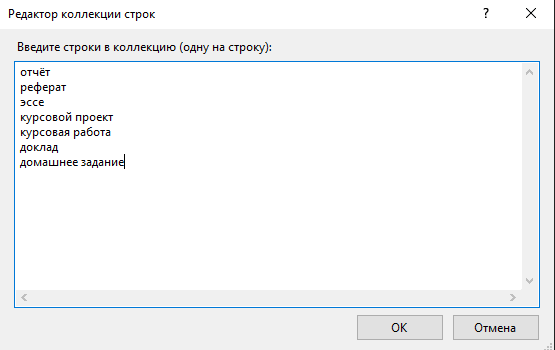


Рисунок 13 – Элементы списка с видами отчётных документов

6. Предусмотреть на графическом пользовательском интерфейсе комбинированный список (*Combo Box*), в котором выбирается вид работы:

- лабораторная работа;

- практическая работа;

- индивидуальное задание;

- учебная практика;

- производственная практика;

- преддипломная практика.

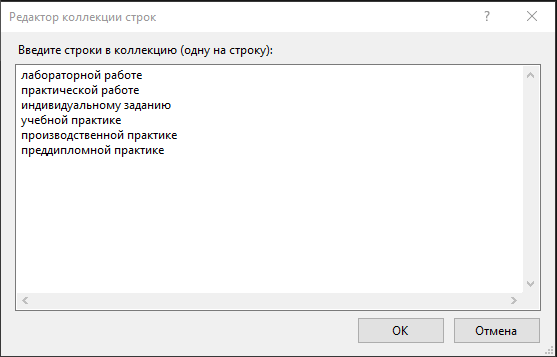


Рисунок 14 - Элементы списка с видами работ

7. Предусмотреть комбинированный список (*Combo Box*), позволяющий выбрать номер работы (диапазон от 1 до 10).

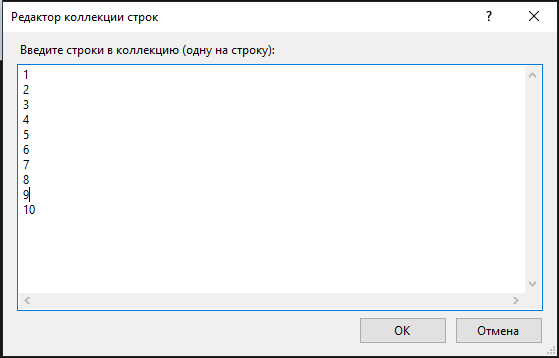


Рисунок 15 - Элементы списка с номерами

8. Предусмотреть текстовое поле для ввода (*Text Box*) наименования работы.

см. Рисунок 7

9. Предусмотреть текстовое поле для ввода (*Text Box*) темы работы.

см. Рисунок 7

10. Предусмотреть текстовое поле для ввода (*Text Box*) наименования дисциплины, в рамках которой выполняется работа.

см. Рисунок 7

11. Между названием вуза и наименованием кафедры должен иметь место один из интерфейсных элементов управления:

- графический разделитель типа «горизонтальная линия» (номер варианта делится на 3 без остатка);

- нижняя граница таблицы, состоящей из одной строки и одной колонки, растянутой по ширине листа (номер варианта делится на 3 и даёт в остатке единицу);

- заполнитель типа «нижнее подчёркивание», выполненный посредством табуляции (номер варианта делится на 3 и даёт в остатке двойку).

см. Рисунок 12

1. Предоставить пользователю дополнительный комбинированный список, задающий семейство шрифтов (*Calibri*, *Times New Roman*, *Courier New* и др.), влияющий абсолютно на все элементы, размещённые на титульном листе. При необходимости ввести поправки на позиционирование элементов для различных шрифтов.

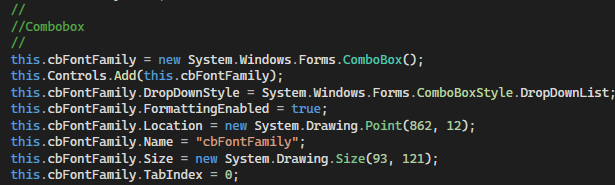
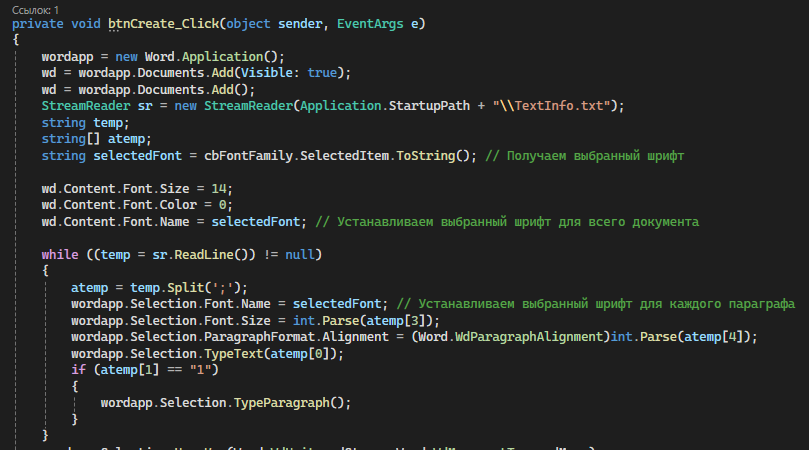
 

Рисунок 16,17 – Фрагмент кода, в котором рассчитывается назначение курса в номере учебной группы студента

# **10. Расчёт тестовых примеров с использованием составленного программного обеспечения.**

* Проверить генерацию титульного листа

см. Рисунок 12

* Проверить генерацию анкеты

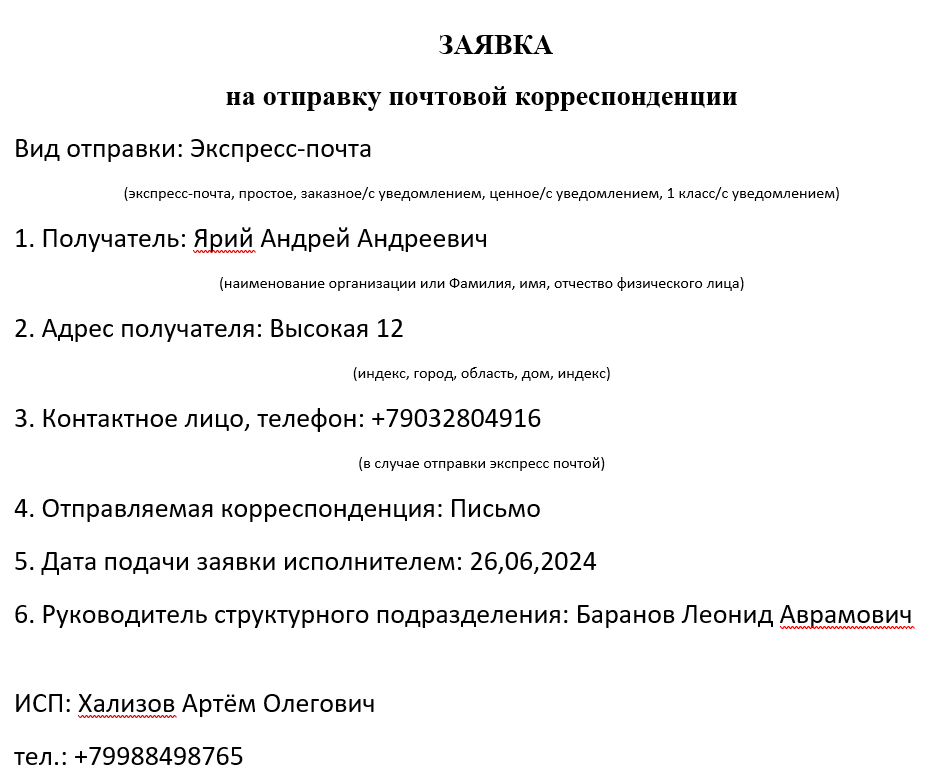


Рисунок 18 - Генерируемая программой анкета

**11. Формулировка вывода о проделанной работе**

В рамках данной работы были закреплены навыки подключения внешних модулей (библиотек классов) и их использования в разрабатываемых программах; навыки программируемой настройки параметров документов Microsoft Office Word. В процессе выполнения проекта были реализованы следующие ключевые аспекты:

Интеграция внешних библиотек:

Подключение библиотеки Microsoft Office Interop для работы с документами Word.

Настройка среды разработки для корректного взаимодействия с внешними модулями.

Освоение работы с объектами и методами библиотеки Interop, что позволило значительно расширить функциональные возможности приложения.

Создание и форматирование документов Word:

Программирование создания новых документов Word с заданными параметрами.

Автоматизация форматирования текста, включая настройку шрифта, размера и стиля текста, а также выравнивания абзацев.

Вставка текстовых данных в документ на основе пользовательского ввода, что обеспечивает персонализацию создаваемых документов.

Пользовательский интерфейс:

Разработка удобного и интуитивно понятного интерфейса для взаимодействия пользователя с программой.

Внедрение комбинированных списков (ComboBox) для выбора параметров форматирования, таких как шрифт и его размер.

Обработка событий, связанных с элементами управления интерфейсом, для динамического обновления содержимого документов на основе выбора пользователя.

Чтение и запись данных:

Реализация функций для загрузки и сохранения информации из внешних файлов, что позволяет сохранять состояние и настройки программы.

Работа с текстовыми файлами для хранения и извлечения пользовательских данных, необходимых для генерации документов.

Оптимизация и надежность кода:

Обеспечение правильного закрытия и освобождения ресурсов, используемых приложением, таких как объекты Word.

Обработка возможных исключений и ошибок для предотвращения сбоев и повышения стабильности программы.

Практическое применение знаний:

Консолидация знаний о программировании на языке C# в реальном проекте.

Применение принципов объектно-ориентированного программирования для создания модульного и расширяемого кода.

Развитие навыков командной работы и управление версиями кода с использованием систем контроля версий.

Эти навыки и знания, полученные в ходе выполнения данной работы, создают прочную основу для дальнейшего профессионального роста в области программной разработки и автоматизации офисных процессов.