Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 17

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Автоматизация Word»

Выполнил: ст. гр. ТУУ-211

Баранов А. А.

Вариант №7

\_\_\_\_\_\_\_\_\_\_\_

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2024 г.

**Оглавление**

[1. Цель работы 3](#_Toc182695869)

[2. Формулировка задачи 3](#_Toc182695870)

[3. Сеть Петри 4](#_Toc182695871)

[4. Составление схем алгоритмов методов в составе решения 5](#_Toc182695872)

[5. Подбор тестовых примеров 11](#_Toc182695873)

[6. Листинг (код) составленного программного обеспечения 15](#_Toc182695874)

[7. Графический пользовательский интерфейс программного обеспечения и его описание 15](#_Toc182695875)

[8. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению 28](#_Toc182695876)

[9. Расчёт тестовых примеров с использованием составленного программного обеспечения 29](#_Toc182695877)

[10. Формулировку вывода о проделанной работе 32](#_Toc182695878)

# **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**

Создать и настроить графический интерфейс, который взаимодействует с модулем (библиотекой классов) текстового редактора *Microsoft Office Word.* В графическом интерфейсе должны быть предусмотрены кнопки «Создать титульный лист» и «Создать документ». Кнопка «Создать титульный лист» должна инициализировать генерацию титульного листа. Кнопка «Создать документ» должна инициализировать генерацию индивидуального документа, выданного преподавателем, по индивидуальному варианту.

# **3. Сеть Петри**

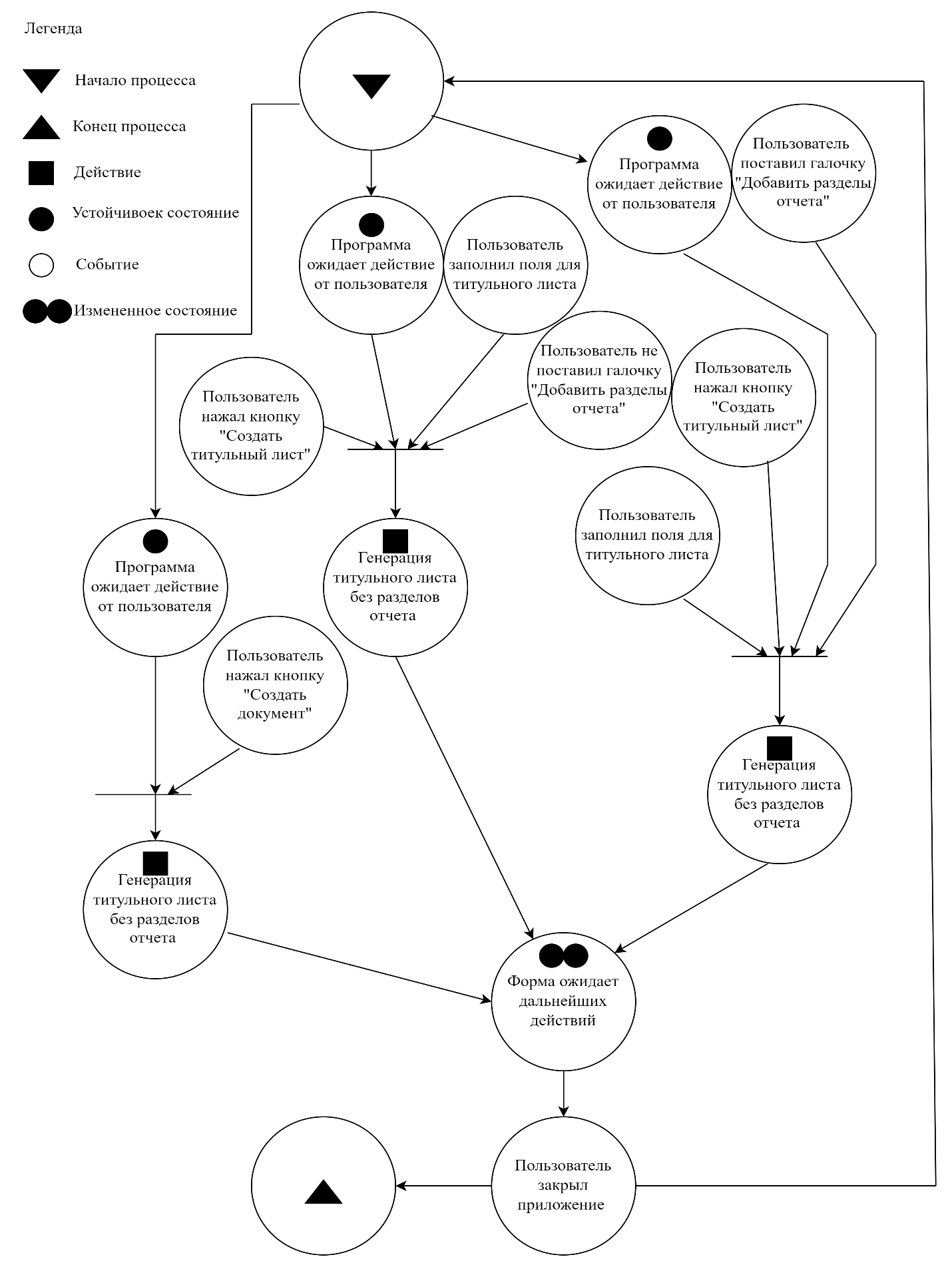


Рисунок 1 – Сеть Петри

# **4. Составление схем алгоритмов методов в составе решения**



Рисунок 2 – Блок-схема метода createATitlePage\_Click

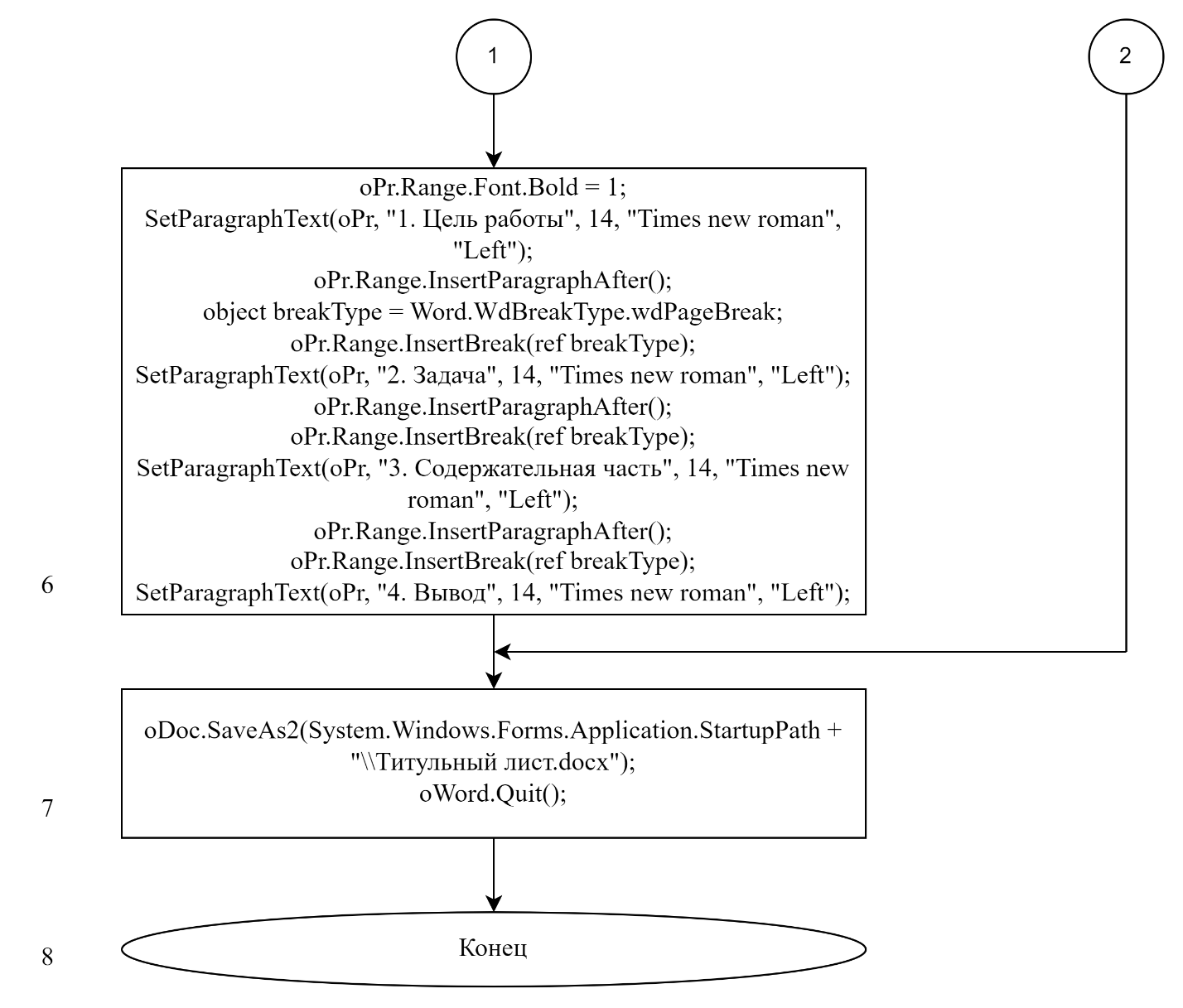


Рисунок 3 – Блок-схема метода createATitlePage\_Click

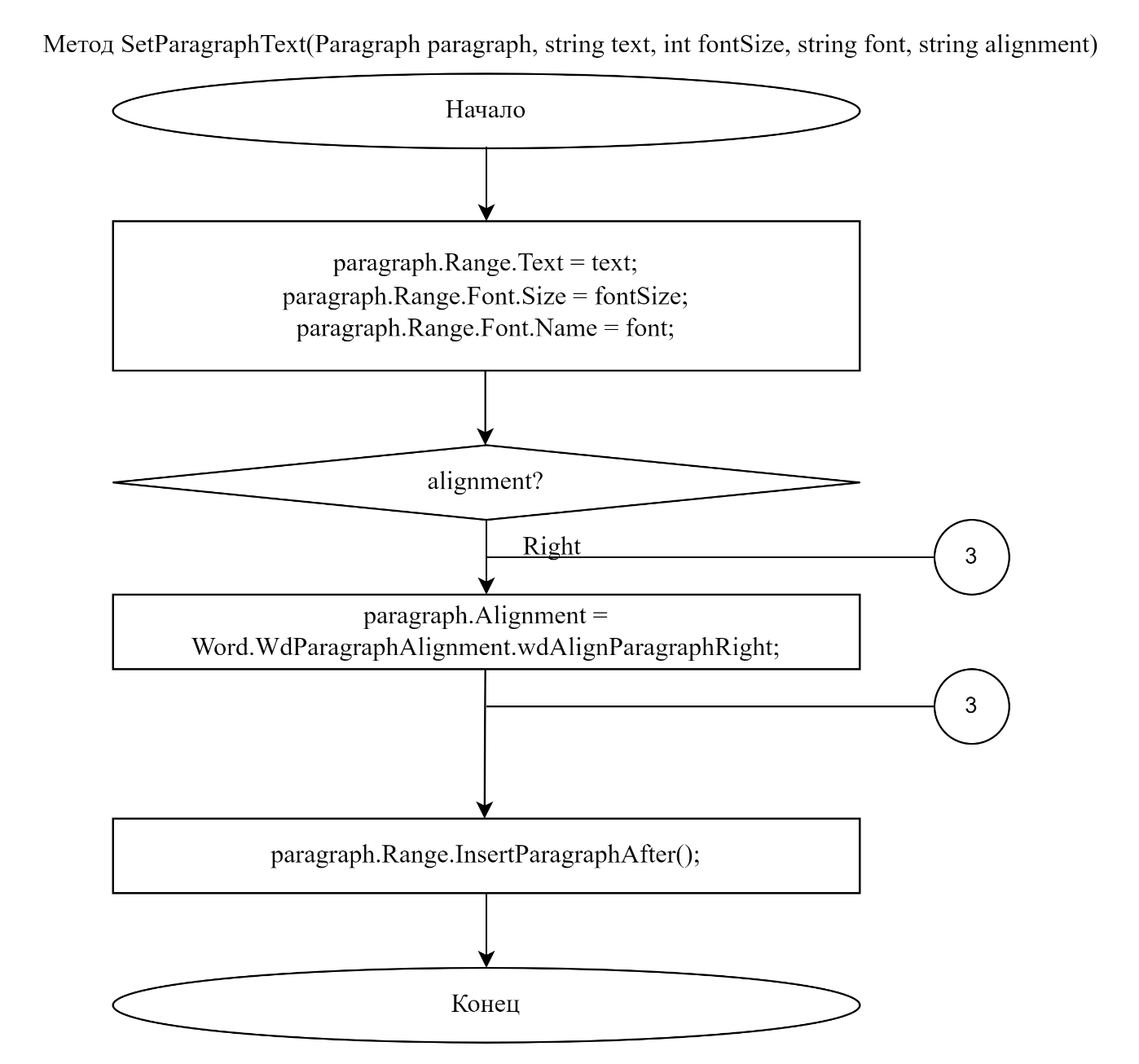


Рисунок 4 – Блок-схема метода SetParagraphText

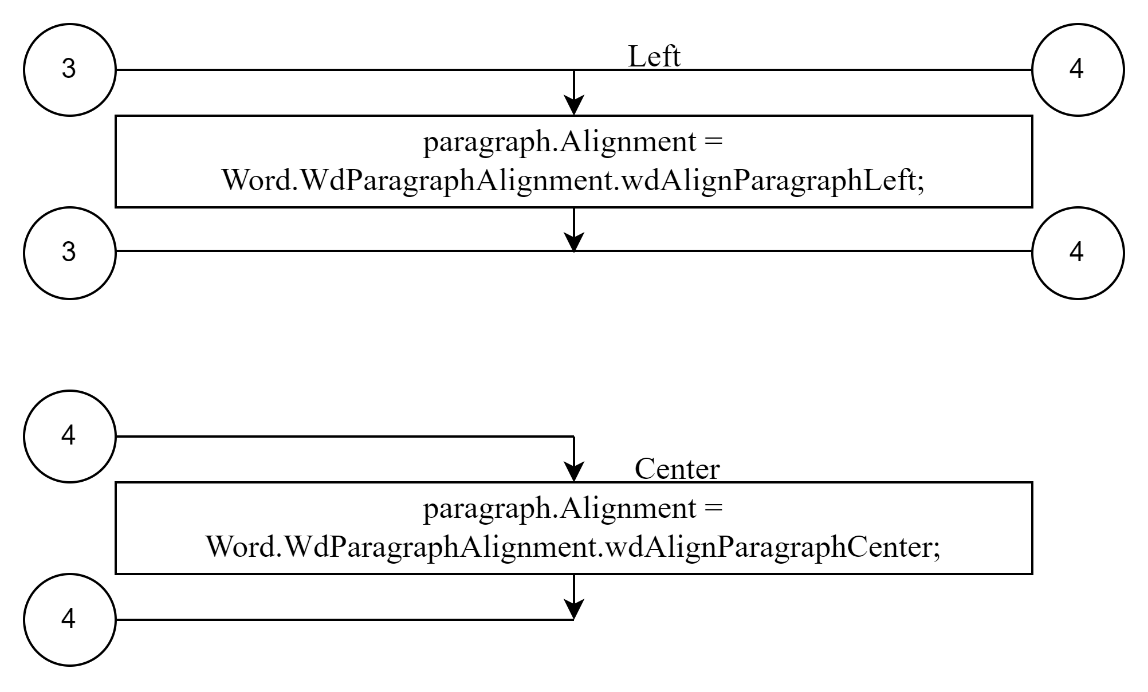


Рисунок 5 – Блок-схема метода SetParagraphText

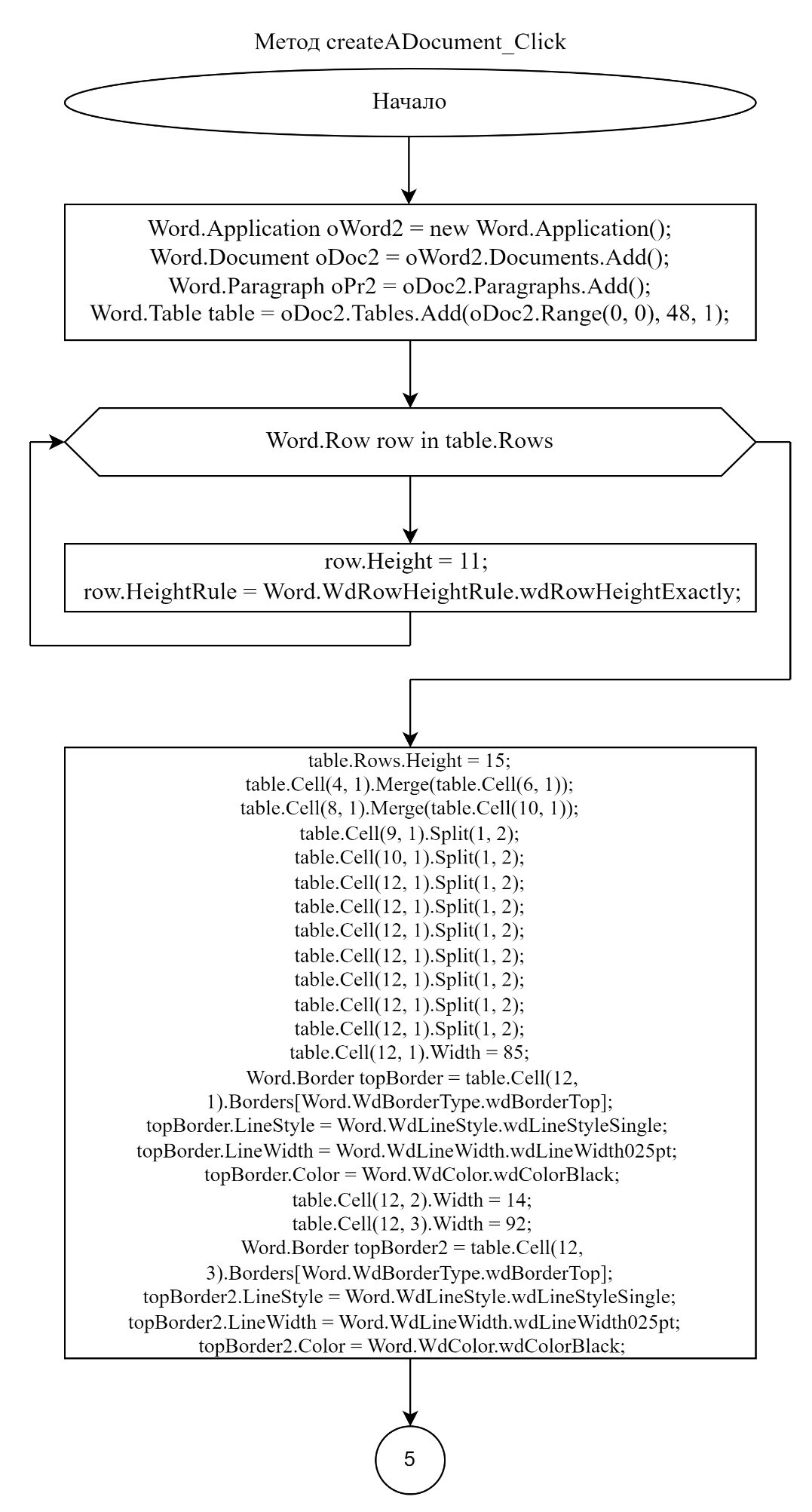


Рисунок 6 – Блок-схема метода createADocument\_Click



Рисунок 7 – Блок-схема метода createATitlePage\_Click

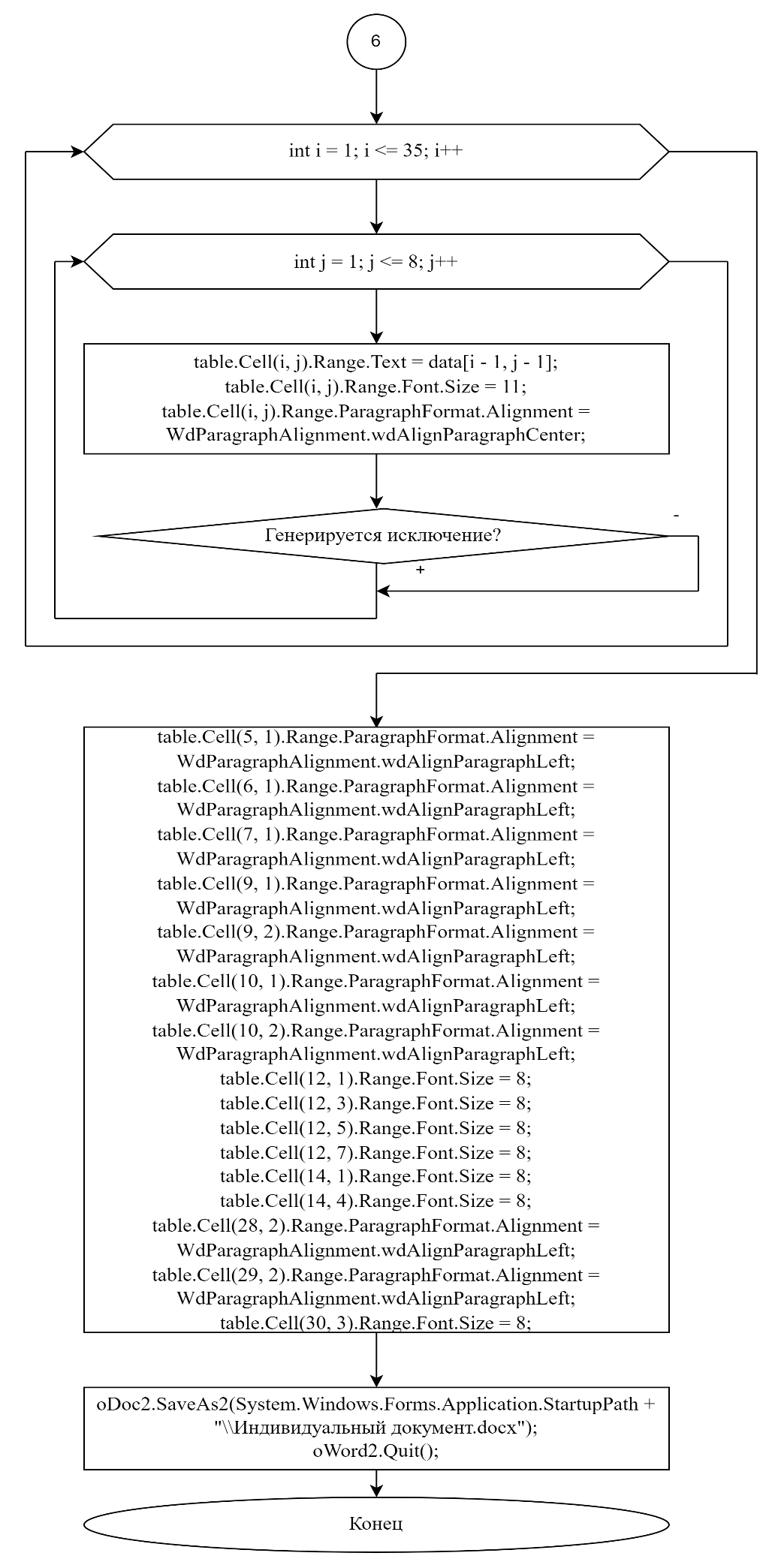


Рисунок 8 – Блок-схема метода createATitlePage\_Click

# **5. Подбор тестовых примеров**

Первый тестовый пример общей части задания

Вид отчетного документа: Отчет

Номер работы: 3

Тема работы: Циклы

Наименование дисциплины: Программирование

Преподаватель: Иванов И.И.

Галочка «Добавить разделы отчета» выключена

Второй тестовый пример общей части задания

Вид отчетного документа: Отчет

Номер работы: 3

Тема работы: Циклы

Наименование дисциплины: Программирование

Преподаватель: Иванов И.И.

Галочка «Добавить разделы отчета» включена

Тестовый пример индивидуальной части задания:

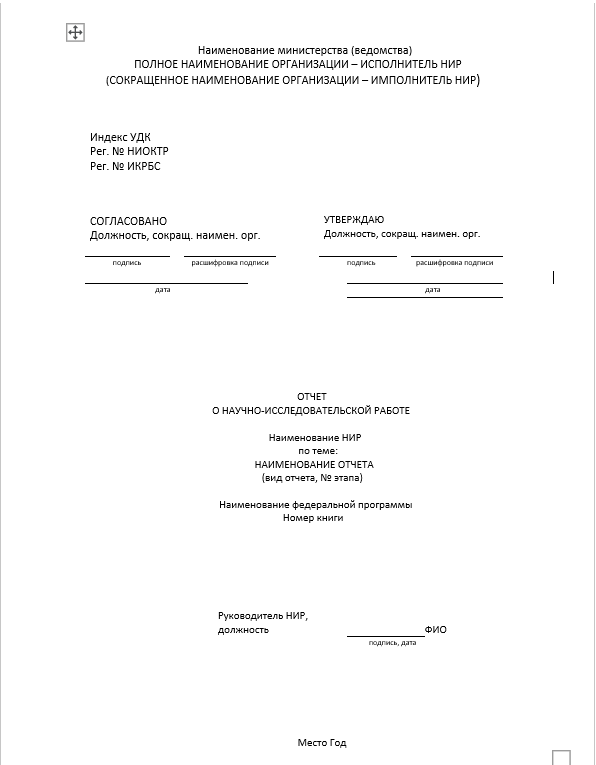


Рисунок 9 – Тестовый пример индивидуальной части задания

# **6. Листинг (код) составленного программного обеспечения**

# using System;

# using System.Windows.Forms;

# using Word = Microsoft.Office.Interop.Word;

# using System.Reflection;

# using Microsoft.Office.Interop.Word;

# namespace laboratornaya\_rabota\_17

# {

# public partial class wordAutomation : Form

# {

# object ObjMissing = Missing.Value;

# private string[] typeOfReportingDocument = { "Отчёт", "Реферат", "Эссе", "Курсовой проект",

# "Курсовая работа", "Доклад", "Домашнее задание"};

# private string[] typeOfWork = { "Лабораторная работа", "Практическая работа", "Индивидуальное задание",

# "Учебная практика", "Производственная практика", "Преддипломная практика"};

# private string[] numberOfWork = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10" };

# public wordAutomation()

# {

# InitializeComponent();

# reportingDocument.Items.AddRange(typeOfReportingDocument);

# workType.Items.AddRange(typeOfWork);

# number.Items.AddRange(numberOfWork);

# 

# }

# private static void SetParagraphText(Paragraph paragraph, string text, int fontSize, string font, string alignment)

# {

# paragraph.Range.Text = text;

# paragraph.Range.Font.Size = fontSize;

# paragraph.Range.Font.Name = font;

# switch (alignment)

# {

# case "Right":

# paragraph.Alignment = Word.WdParagraphAlignment.wdAlignParagraphRight;

# break;

# case "Left":

# paragraph.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

# break;

# case "Center":

# paragraph.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

# break;

# default:

# break;

# }

# paragraph.Range.InsertParagraphAfter();

# }

# private void createATitlePage\_Click(object sender, EventArgs e)

# {

# Word.Document oDoc;

# Word.Paragraph oPr;

# Word.Application oWord = new Word.Application();

# oDoc = oWord.Documents.Add();

# oPr = oDoc.Paragraphs.Add();

# 

# SetParagraphText(oPr, "Министерство транспорта Российской Федерации", 14, "Times new roman", "Center");

# SetParagraphText(oPr, "Федеральное государственное автономное образовательное", 14, "Times new roman", "Center");

# SetParagraphText(oPr, "учреждение высшего образования", 14, "Times new roman", "Center");

# SetParagraphText(oPr, "«Российский университет транспорта»", 14, "Times new roman", "Center");

# SetParagraphText(oPr, "(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)", 14, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, "Институт транспортной техники и систем управления", 14, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, "Кафедра «Управление и защита информации»", 14, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertParagraphAfter();

# string changingText = (reportingDocument.SelectedIndex >= 0 ? reportingDocument.Text : workType.Text) + " №" + number.Text;

# SetParagraphText(oPr, changingText, 28, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, $"по дисциплине: «{nameOfTheDiscipline.Text}»", 14, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, $"на тему: «{topicOfWork.Text}»", 14, "Times new roman", "Center");

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, "Выполнил: ст. гр. ТУУ-211", 14, "Times new roman", "Right");

# SetParagraphText(oPr, "Баранов А.А.", 14, "Times new roman", "Right");

# SetParagraphText(oPr, "Вариант №7", 14, "Times new roman", "Right");

# SetParagraphText(oPr, $"Проверил: {teacher.Text}", 14, "Times new roman", "Right");

# oPr.Range.InsertParagraphAfter();

# SetParagraphText(oPr, "Москва – 2024 г.", 14, "Times new roman", "Center");

# if (addReportSections.Checked)

# {

# oPr.Range.Font.Bold = 1;

# SetParagraphText(oPr, "1. Цель работы", 14, "Times new roman", "Left");

# oPr.Range.InsertParagraphAfter();

# object breakType = Word.WdBreakType.wdPageBreak;

# oPr.Range.InsertBreak(ref breakType);

# SetParagraphText(oPr, "2. Задача", 14, "Times new roman", "Left");

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertBreak(ref breakType);

# SetParagraphText(oPr, "3. Содержательная часть", 14, "Times new roman", "Left");

# oPr.Range.InsertParagraphAfter();

# oPr.Range.InsertBreak(ref breakType);

# SetParagraphText(oPr, "4. Вывод", 14, "Times new roman", "Left");

# }

# oDoc.SaveAs2(System.Windows.Forms.Application.StartupPath + "\\Титульный лист.docx");

# oWord.Quit();

# }

# private void createADocument\_Click(object sender, EventArgs e)

# {

# 

# Word.Application oWord2 = new Word.Application();

# Word.Document oDoc2 = oWord2.Documents.Add();

# Word.Paragraph oPr2 = oDoc2.Paragraphs.Add();

# Word.Table table = oDoc2.Tables.Add(oDoc2.Range(0, 0), 48, 1);

# foreach (Word.Row row in table.Rows)

# {

# row.Height = 11;

# row.HeightRule = Word.WdRowHeightRule.wdRowHeightExactly;

# }

# table.Rows.Height = 15;

# table.Cell(4, 1).Merge(table.Cell(6, 1));

# table.Cell(8, 1).Merge(table.Cell(10, 1));

# table.Cell(9, 1).Split(1, 2);

# table.Cell(10, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Split(1, 2);

# table.Cell(12, 1).Width = 85;

# Word.Border topBorder = table.Cell(12, 1).Borders[Word.WdBorderType.wdBorderTop];

# topBorder.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# topBorder.Color = Word.WdColor.wdColorBlack;

# table.Cell(12, 2).Width = 14;

# table.Cell(12, 3).Width = 92;

# Word.Border topBorder2 = table.Cell(12, 3).Borders[Word.WdBorderType.wdBorderTop];

# topBorder2.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder2.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# topBorder2.Color = Word.WdColor.wdColorBlack;

# table.Cell(12, 4).Width = 41;

# table.Cell(12, 5).Width = 77;

# Word.Border topBorder3 = table.Cell(12, 5).Borders[Word.WdBorderType.wdBorderTop];

# topBorder3.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder3.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# topBorder3.Color = Word.WdColor.wdColorBlack;

# table.Cell(12, 6).Width = 14;

# table.Cell(12, 7).Width = 92;

# Word.Border topBorder4 = table.Cell(12, 7).Borders[Word.WdBorderType.wdBorderTop];

# topBorder4.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder4.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# topBorder4.Color = Word.WdColor.wdColorBlack;

# table.Cell(12, 8).Width = 53;

# table.Cell(14, 1).Split(1, 2);

# table.Cell(14, 1).Split(1, 2);

# table.Cell(14, 3).Split(1, 2);

# table.Cell(14, 4).Split(1, 2);

# table.Cell(14, 1).Width = 163;

# Word.Border topBorder5 = table.Cell(14, 1).Borders[Word.WdBorderType.wdBorderTop];

# topBorder5.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder5.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# table.Cell(14, 2).Width = 71;

# table.Cell(14, 3).Width = 29;

# table.Cell(14, 4).Width = 156;

# Word.Border topBorder6 = table.Cell(14, 4).Borders[Word.WdBorderType.wdBorderTop];

# topBorder6.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder6.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# table.Cell(14, 5).Width = 49;

# table.Cell(15, 1).Merge(table.Cell(20, 1));

# table.Cell(26, 1).Merge(table.Cell(30, 1));

# table.Cell(28, 1).Split(1, 3);

# table.Cell(28, 1).Width = 128;

# table.Cell(28, 2).Width = 106;

# table.Cell(28, 3).Width = 233;

# table.Cell(29, 1).Split(1, 3);

# table.Cell(29, 1).Width = 128;

# table.Cell(29, 2).Width = 106;

# table.Cell(29, 3).Width = 233;

# table.Cell(30, 1).Split(1, 4);

# table.Cell(30, 1).Width = 234;

# table.Cell(30, 2).Width = 28;

# table.Cell(30, 3).Width = 78;

# Word.Border topBorder7 = table.Cell(30, 3).Borders[Word.WdBorderType.wdBorderTop];

# topBorder7.LineStyle = Word.WdLineStyle.wdLineStyleSingle;

# topBorder7.LineWidth = Word.WdLineWidth.wdLineWidth025pt;

# table.Cell(30, 4).Width = 127;

# //table.Cell(31, 1).Merge(table.Cell(33, 1));

# table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleNone;

# string[,] data = {

# {"Наименование министерства (ведомства)", "", "", "", "", "", "", ""},

# {"ПОЛНОЕ НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ – ИСПОЛНИТЕЛЬ НИР", "", "", "", "", "", "", ""},

# {"(СОКРАЩЕННОЕ НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ – ИМПОЛНИТЕЛЬ НИР)", "", "", "", "", "", "", ""},

# {"", "", "", "", "", "", "", ""},

# {"Индекс УДК", "", "", "", "", "", "", ""},

# {"Рег. № НИОКТР", "", "", "", "", "", "", ""},

# {"Рег. № ИКРБС", "", "", "", "", "", "", ""},

# {"", "", "", "", "", "", "", ""},

# {"СОГЛАСОВАНО", "УТВЕРЖДАЮ", "", "", "", "", "", ""},

# {"Должность, сокращ. наимен. орг.", "Должность, сокращ. наимен. орг.", "", "", "", "", "", ""},

# {"", "", "", "", "", "", "", ""},

# {"подпись", "", "расшифровка подписи", "", "подпись", "", "расшифровка подписи", ""},

# { "", "", "", "", "", "", "", ""},

# { "дата", "", "", "дата", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "ОТЧЕТ", "", "", "", "", "", "", ""},

# { "О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "Наименование НИР", "", "", "", "", "", "", ""},

# { "по теме:", "", "", "", "", "", "", ""},

# { "НАИМЕНОВАНИЕ ОТЧЕТА", "", "", "", "", "", "", ""},

# { "(вид отчета, № этапа)", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "Наименование федеральной программы", "", "", "", "", "", "", ""},

# { "Номер книги", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "", "Руководитель НИР,", "", "", "", "", "", ""},

# { "", "должность", "ФИО", "", "", "", "", ""},

# { "", "", "подпись, дата", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "", "", "", "", "", "", "", ""},

# { "Место Год", "", "", "", "", "", "", ""}

# };

# for (int i = 1; i <= 35; i++)

# {

# for (int j = 1; j <= 8; j++)

# {

# try

# {

# table.Cell(i, j).Range.Text = data[i - 1, j - 1];

# table.Cell(i, j).Range.Font.Size = 11;

# table.Cell(i, j).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphCenter;

# }

# catch

# {

# 

# }

# }

# }

# //Настройка отдельных ячеек

# table.Cell(5, 1).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(6, 1).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(7, 1).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(9, 1).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(9, 2).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(10, 1).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(10, 2).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(12, 1).Range.Font.Size = 8;

# table.Cell(12, 3).Range.Font.Size = 8;

# table.Cell(12, 5).Range.Font.Size = 8;

# table.Cell(12, 7).Range.Font.Size = 8;

# table.Cell(14, 1).Range.Font.Size = 8;

# table.Cell(14, 4).Range.Font.Size = 8;

# table.Cell(28, 2).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(29, 2).Range.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

# table.Cell(30, 3).Range.Font.Size = 8;

# oDoc2.SaveAs2(System.Windows.Forms.Application.StartupPath + "\\Индивидуальный документ.docx");

# oWord2.Quit();

# }

# }

# }

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

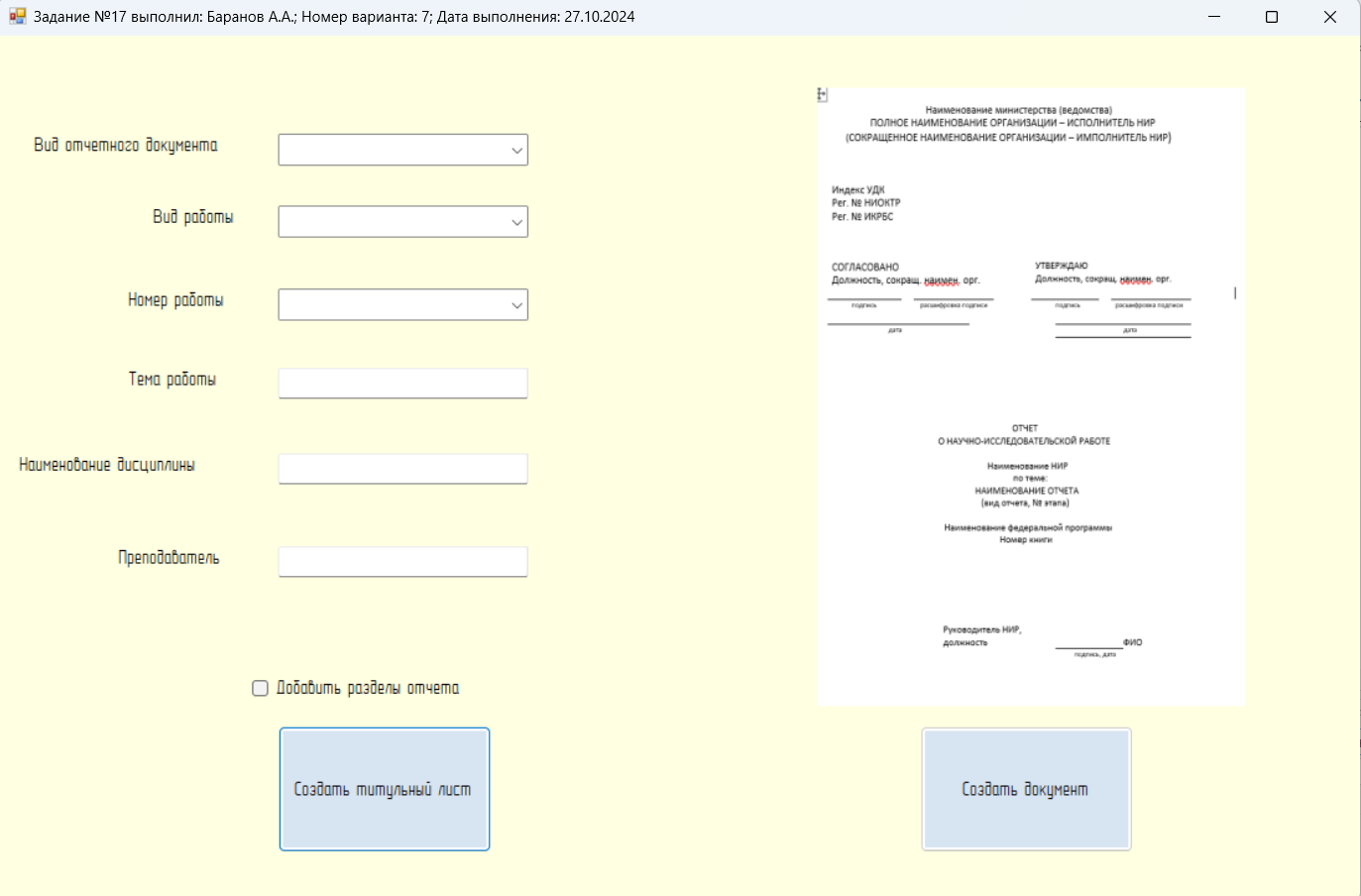


Рисунок 10 – Графический пользовательский интерфейс программного обеспечения

На главной экранной форме располагаются текстовые поля для настройки титульного листа, кнопки для создания титульного листа и создания документа, галочка, при включении которой добавляются разделы отчета, текстовые элементы, которые помогают пользователю настроить титульный лист, а также предпросмотр индивидуального документа, для того, чтобы пользователь мог понять какой документ сгенерируется при нажатии на кнопку «Создать документ».

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

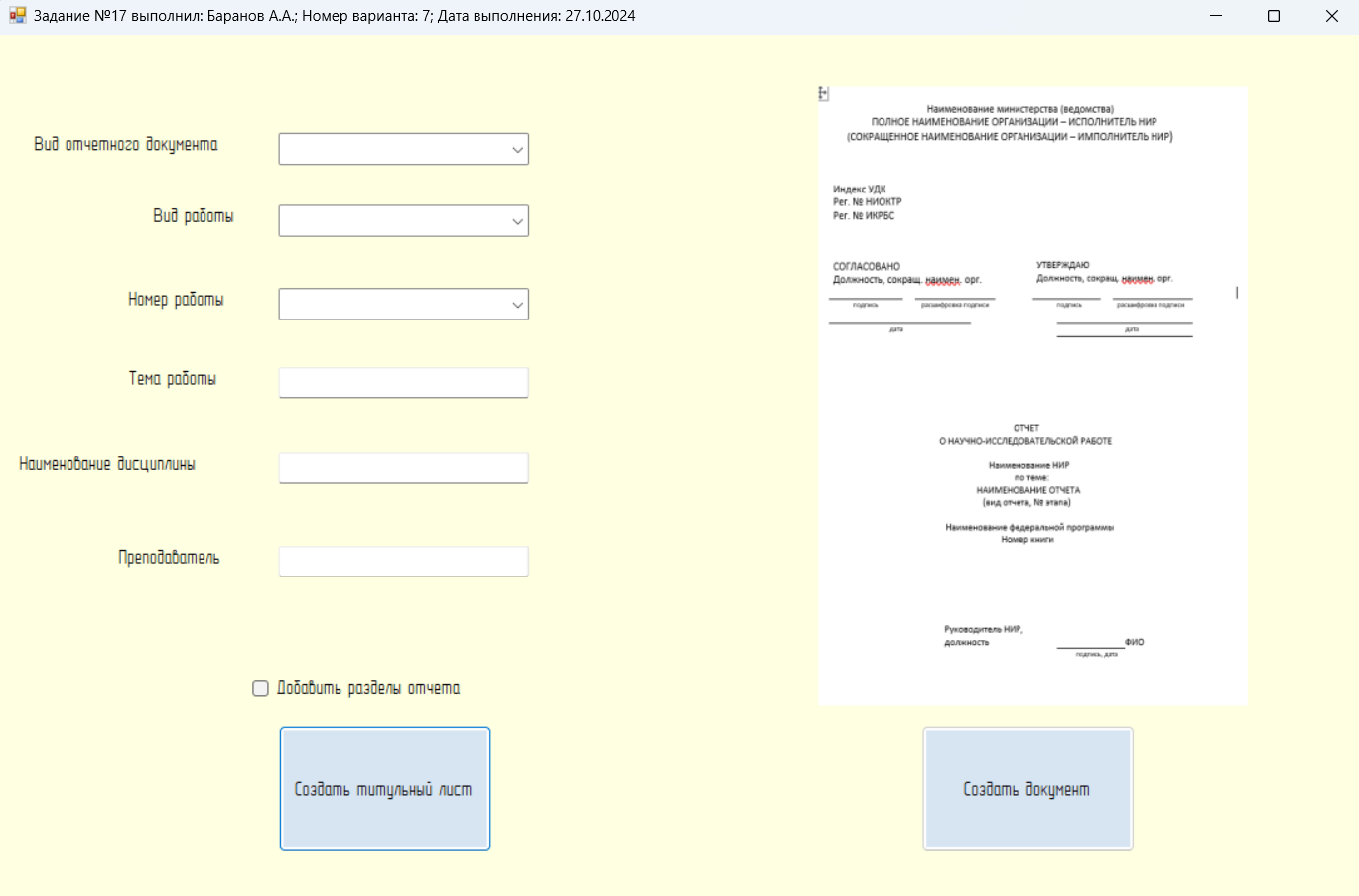


Рисунок 11 – Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению

На экранной форме располагаются элементы, согласно описанию графического пользовательского интерфейса.

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

Тестирование первого примера для общей части задания

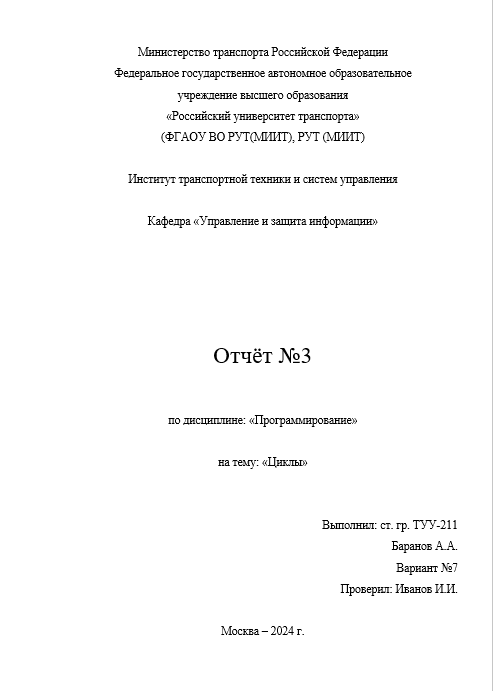


Рисунок 12 – Тестирование первого примера для общей части задания

Тестирование второго примера для общей части задания

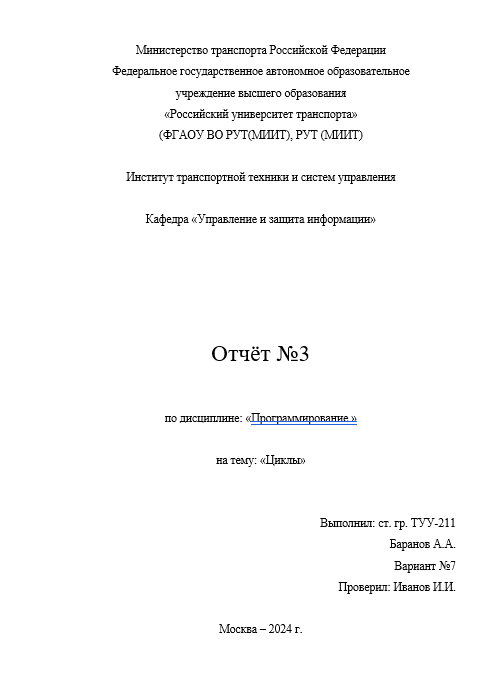


Рисунок 13 – Тестирование второго примера для общей части задания



Тестирование второго примера для общей части задания

Рисунок 14 – Тестирование второго примера для общей части задания



Рисунок 15 – Тестирование второго примера для общей части задания



Рисунок 16 – Тестирование второго примера для общей части задания



Рисунок 16 – Тестирование второго примера для общей части задания

Тестирование общей части задания

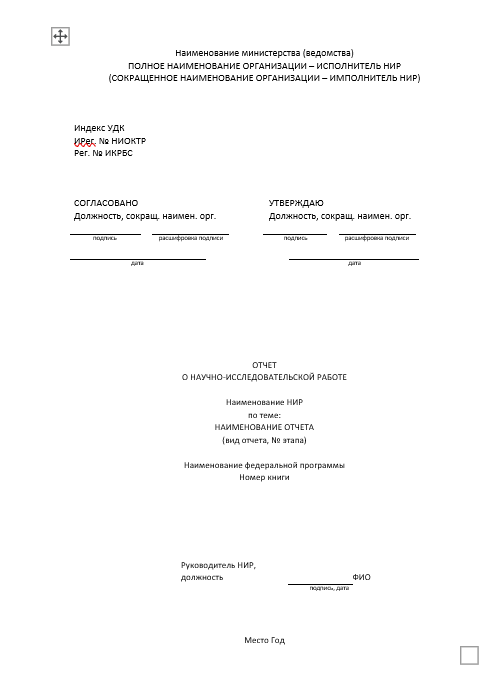


Рисунок 10 – Расчёт тестовых примеров с использованием составленного программного обеспечения

# **10. Формулировку вывода о проделанной работе**

В процессе работы получил навык подключения внешних модулей (библиотек классов), освоил работу с *Microsoft Office Word* программным, освоил работу с абзацами, редактирование текста, выравнинвание текста, сохранение документов, работу с таблицами, а так же получил практический навык работы с объектами и закрепил навык работы с *Windows Forms*.