**Министерство сельского хозяйства и продовольствия Республики Беларусь**

**УО «Столинский государственный аграрно-экономический колледж»**

**Отделение «Бухгалтерский учет и информационные технологии»**

ДОПУЩЕН К ЗАЩИТЕ

Заведующий учебно-производственной практикой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.С. Ярмошик

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**ПРОЕКТА**

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

(Тема проекта)

Специальность

2-40 01 01 «Программное обеспечение информационных технологий»

Специализация

2-40 01 01 35 «Программное обеспечение обработки экономической и деловой информации»

Учащийся

группа \_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

Руководитель

(подпись) (Ф.И.О.)

Отметка

Столин 20\_\_

**Министерство сельского хозяйства и продовольствия Республики Беларусь**

**УО «Столинский государственный аграрно-экономический колледж**

**ЗАДАНИЕ**

на выполнение проекта учащемуся

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

(Фамилия, Имя, Отчество)

1. Тема работы:

2. Исходные данные к проекту

2.1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. СОСТАВ ПРОЕКТА

3.1 Содержание расчётно-пояснительной записки (перечень подлежащих разработке вопросов)

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

3.2 Перечень графического материала (с точным указанием обязательных чертежей)

3.2.1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2.2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2.3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.2.4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_20\_\_г.

5. Срок сдачи учащимся законченной работы \_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.И.О.

(подпись)

Задание принял к исполнению

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_20\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.И.О.

**РЕФЕРАТ**

Пояснительная записка проекта: 120 с, 11 рис., 19 табл., 21 источник, 9 прил., 4 плаката (формат А1).

Объектом разработки является...

Цель проекта...

В процессе проектирования выполнены следующие разработки ...

Элементами практической значимости полученных результатов являются...

Областью возможного практического применения являются ....

В ходе проектирования прошли апробацию такие предложения, как...

Результатами внедрения явились...

Учащийся-дипломник подтверждает, что приведенный в проекте расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологиче-ские положения и концепции сопровождаются ссылками на их авторов.

**Перечень условных обозначений**

ПО – программное обеспечение

ОС – операционная система

ОЗУ – оперативное запоминающее устройство

БИС – большая интегральная схема

ИС – интегральная схема

СБИС – сверхбольшая интегральная схема

ПЛМ – программируемая логическая матрица

ГСП – генератор случайных последовательностей

ГПП – генератор псевдослучайных последовательностей

СЛН – средства локализации неисправностей

**СОДЕРЖАНИЕ**

[Введение 6](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536499)

[1. Анализ исходных данных и постановка цели задачи 7](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536500)

[2. Техническое задание 8](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536501)

[3. Функциональная модель ПС 9](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536502)

[4. Концептуальная модель UML 10](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536503)

[4.1. Вариантов использования 10](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536504)

[4.2. Прецедентов 11](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536505)

[4.3. Деятельности 12](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536506)

[4.4. Состояния 13](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536507)

[4.5. Классов 14](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536508)

[4.6. Контекстная 15](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536509)

[4.7. Компонентов 16](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536510)

[4.8. Развёртывания 17](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536511)

[5. Проектирование интерфейса 18](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536512)

[5.1 Общая инфраструктура пользовательского интерфейса 18](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536513)

[5.2. Проектирование окон 19](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536514)

[6. Кодирование модулей системы 20](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536515)

[7. Реализация и тестирование 21](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536516)

[8. Оценка стоимости 22](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536517)

[9. Заключение 23](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536518)

[10. Список использованных источников 24](file:///C:\Users\100nout.by\Downloads\Otchyot_Avtosokhranenny.doc#_Toc90536519)

# Введение

Программное обеспечение разработано как набор программ, каждая из которых реализует заданное множество операций, где под операцией понимается некоторое логически законченное действие над информацией, хранящейся в Базе Данных, инициированное пользователем системы. Все программы осуществляют диалоговое взаимодействие с пользователями.

В проекте представлена программа, позволяющая совершать покупки, изменять данные.

Для разработки базы данных была выбрана система управления базами данных MySQL, которая является общепризнанным лидером среди инструментов для создания приложений и систем.

Объектом данного проекта является автоматизация работы учебного отдела.

# 1. Анализ исходных данных и постановка цели задачи

Исходные данные: учащиеся, работники отдела.

Перечисленные исходные данные являются основной информацией, которая используется в учебном отделе.

В данном случае постановщиком задачи и пользователем программы является заведующий учебным отделом.

Цель – написание качественного, отлаженного и работоспособного программного продукта по теме «Автоматизация работы учебного отдела».

Данная система позволяет решать следующие задачи:

1) Вести учет учащихся.

2) Вести учет отчисленных учащихся.

3) Вести отчет учащихся которые находятся в академическом отпуске.

4) Просмотр информации об работниках учебного отдела.

5) Добавление записи

6) Поиск записи

# 2. Техническое задание

# 1. Общие сведения

## 1.1 Формулировка задания

## Разработать программный продукт (ПП), предназначенный для автоматизации работы учебного отдела.

## 1.2 Цели, достигаемые разработкой

Автоматизация обработки информации учебного отдела.

## 1.3 Категории пользователей

ПП должен быть рассчитан на работников учебного отдела.

## 1.4 Наименование организации заказчика

Заказчиком на разработку данного ПП является «IBM».

(225725 г. Пинск, ул. Калинина 12/ 12, тел/ факс 278-65-42).

## 1.5 Основание для проведения работ

Основанием для разработки данного ПП является договор между

Заказчиком («IBM») и Разработчиком («СГАЭК») от 01.09.2021 г.

# 2. Описание предметной области

## 2.1 Описание (схемы) бизнес-процессов

Автосервис «IBM» выполняет производитель и поставщик аппаратного и программного обеспечения, а также ИТ-сервисов и консалтинговых услуг. IBM активно развивает свой аналитический бизнес.

Уровень конкуренции для компании в последнее время возрос, так как на рынок вышли два новых конкурента, к которым перешла часть клиентов и ряд наиболее квалифицированных сотрудников «IBM».

По предварительным планам, Компания намерена открыть также дочернее предприятие для организации производства в непосредственной близости к своим заказчикам.

С помощью разрабатываемого ПП должна осуществляться автоматизация работы учебного отдела.

## 2.2 Состав данных и алгоритмы обработки информации

Входными данными для разрабатываемого ПП являются:

Выходными данными разрабатываемого ПП являются:

## 2.3 Недостатки существующих проектных решений

На данный момент существует несколько аналогов, таких как: Word, Excel, Access. Однако все эти ПП не обеспечивают полноценной автоматизация обработки текстовой информации, а лишь автоматизируют ее часть.

## 2.4 Текущий уровень автоматизации

На данный момент Заказчик не располагает каким-либо ПП, обеспечивающим полноценную автоматизацию всей его деятельности.

# 3. Требования к разработке

## 3.1 Информационная модель

Информационная модель представлена на рисунке 3.1.

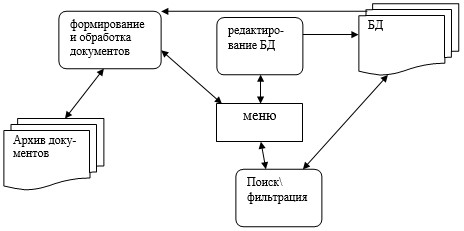


Рисунок 3.1 – Информационная модель проекта

## 3.2 Структура меню

Главное меню

Ученики в УО

Ученики в академическом отпуске

Отчисленные ученики

Состав отдела

О программе

## 3.3 Функциональные требования

Разрабатываемое ПО должно реализовать следующие функции:

* Изменение информации в базе данных
* Копирование выделенного текста
* Просмотр информации
* Удаление информации
* Добавление информации

3.4 Требования к информационному обеспечению

Разрабатываемый ПП должен:

* Запуск программы
* Просмотр вкладок
* Добавление информации на вкладку
* Поиск информации на вкладке
* Переход из вкладки на главное окно
* Изменение формата шрифта

## 3.5 Требования к интерфейсу

Разрабатываемый ПП должен быть обеспечен упрощенной формой интерфейса:

* глубина вложенности окон должна быть не более семи;
* содержать краткие описания элементов меню;
* должен быть понятным и простым в использовании.
* панель интерфейса должна содержать кнопку, предназначенную для сохранения необходимой информации.

При необходимости интерфейс изменяется только Разработчиком.

## 3.6 Требования к алгоритмам

Разрабатываемые алгоритмы должны легко модифицироваться. В течение сроков, установленных в договоре между Заказчиком и Разработчиком в ПП могут быть внесены изменения, не требующие глобальной переработки алгоритмов.

3.7 *Прочие требования*

Разрабатываемый ПП должен функционировать в ОС Windows XP, 7, 10.

Другие ОС не поддерживают формат создаваемых файлов в процессе работы с ПП.

ПП должен использовать нижние регистры памяти, что обеспечивает защиту от сбоев на 95 %.

# 4. Порядок контроля и обеспечение качества

## 4.1 Экспертиза

Не требуется разработка макета на стадии «Технический проект». Требуется проведение экспертизы при участии представителя Заказчика.

## 4.2 Тестирование

Тестирование должно проводиться Разработчиком в соответствии с установленными правилами. Присутствие представителя Заказчика необязательно.

## 4.3 Опытная эксплуатация

Опытная эксплуатация проводится Заказчиком, при контроле Разработчика, в соответствии с договором.

# 5. Требования к документированию

## 5.1 Требования к справочной подсистеме

Справочная система должна содержать описание всех команд, использующихся в ПП. Справочная система должна быть встроенной в интерфейс и располагаться непосредственно в окнах.

5.2 *Требования к документации*

Разрабатываемый ПП должен сопровождаться полным пакетом конструкторской и эксплуатационной документации, перечисленной в договоре.

# Разработанный ПП сопровождается полным пакетом документов, предусмотренным ГОСТ 7685 – 2000, а именно: лицензионное право использования, разработанного ПП; документация по эксплуатации; талон на последующее обслуживание ПП; документы, подтверждающие монопольное использование разработанного ПП.

# 3. Функциональная модель ПС

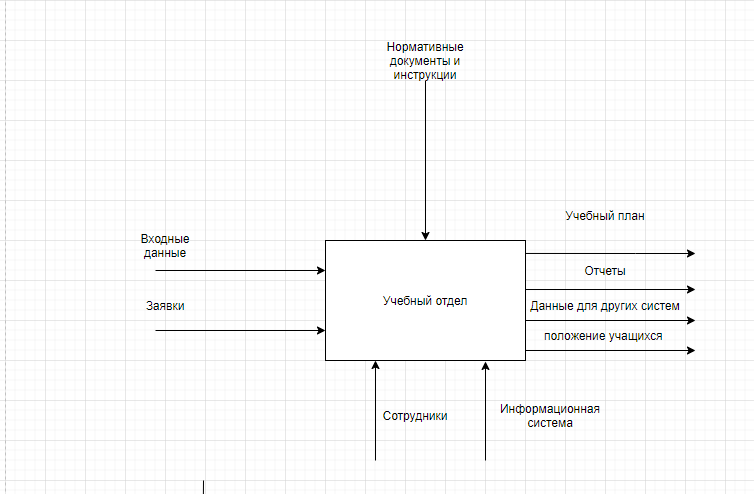


Рисунок 3.1. Контекстная диаграмма

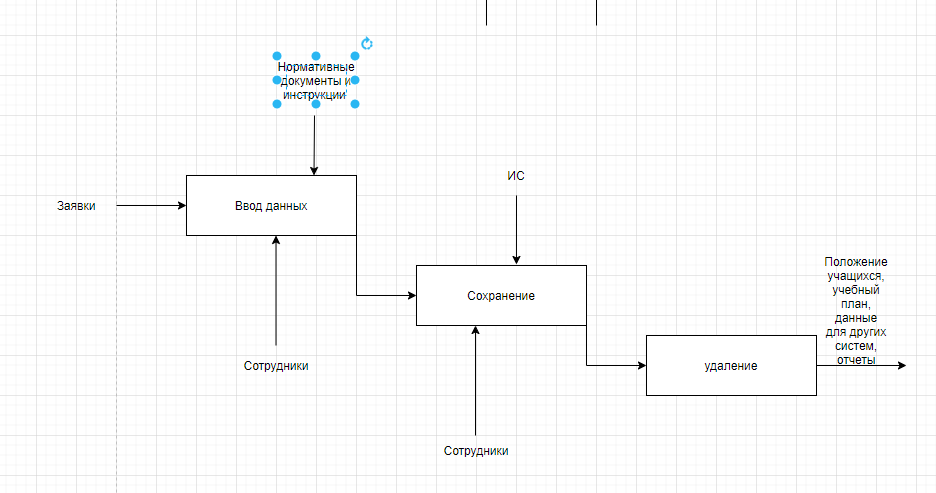


Рисунок 3.2. Декомпозиция функциональных диаграмм

# 4. Концептуальная модель UML

## 4.1. Вариантов использования

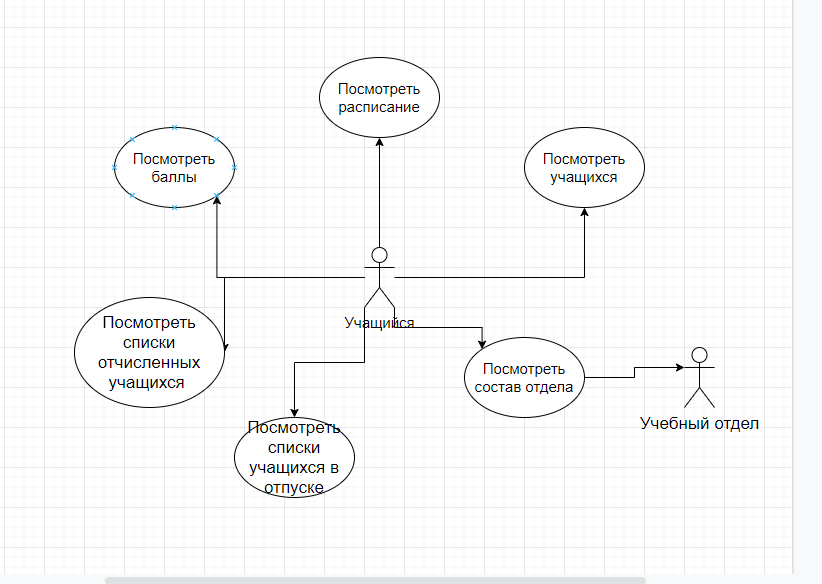


Рисунок 4.1.1. Диаграмма вариантов использования

## 4.2. Прецедентов

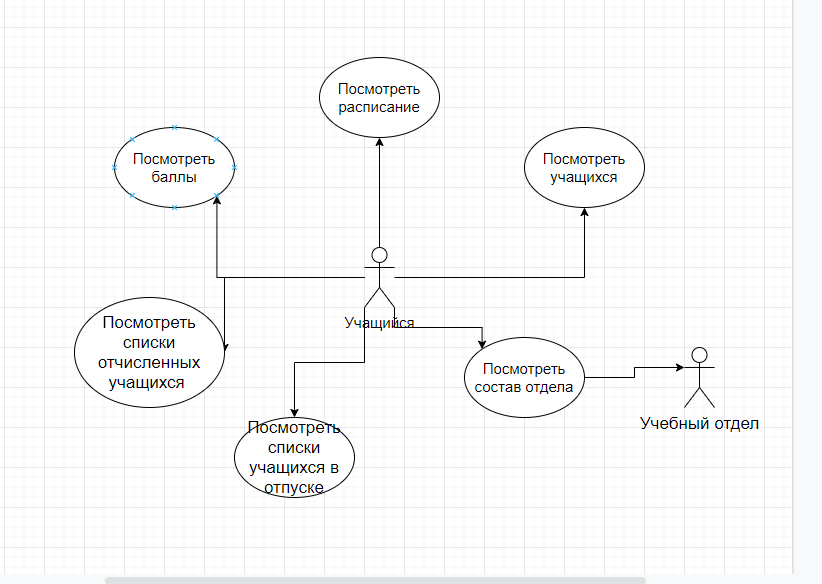


Рисунок 4.2.1. Диаграмма прецедентов

## 4.3. Деятельности

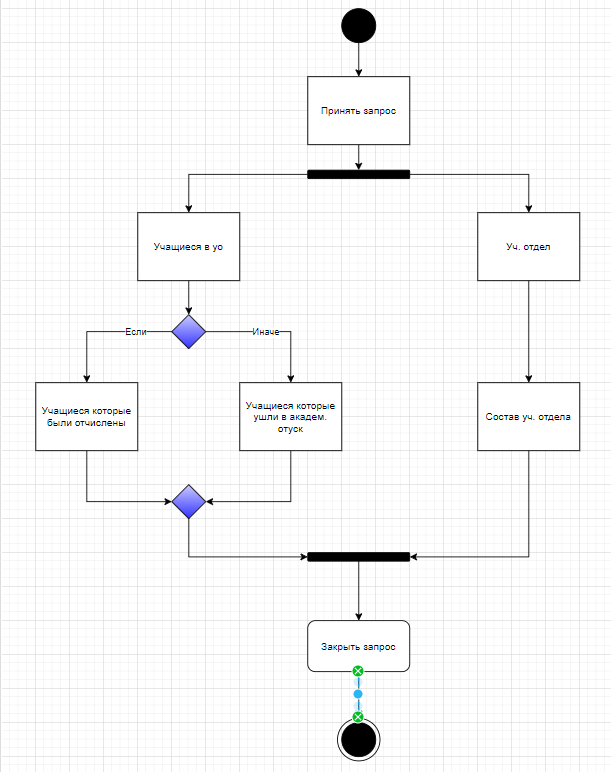


Рисунок 4.3.1. Диаграмма деятельности

## 4.4. Состояния

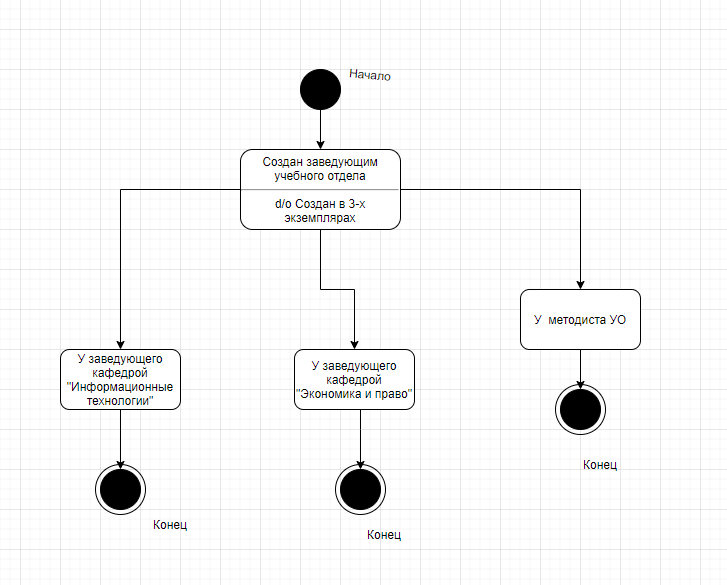


Рисунок 4.4.1. Диаграмма состояния

## 4.5. Классов

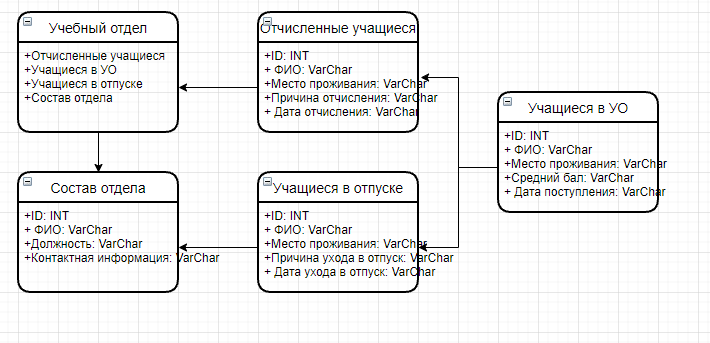
****

Рисунок 4.5.1. Диаграмма классов

## 4.6. Контекстная

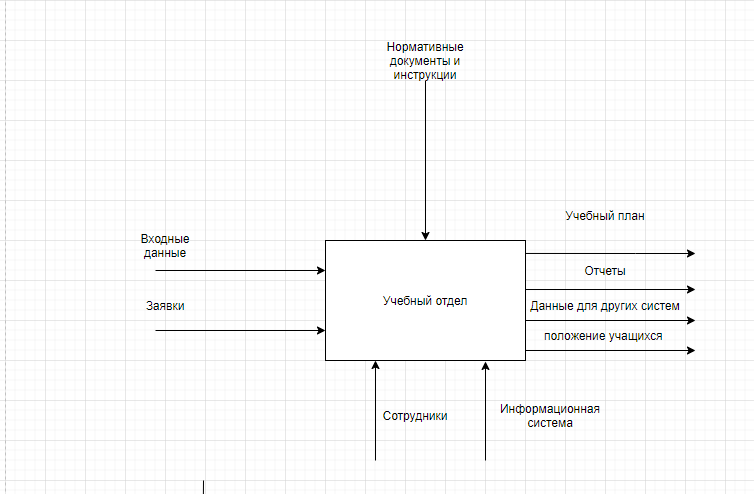


Рисунок 4.6.1. Контекстная диаграмма

## 4.7. Компонентов

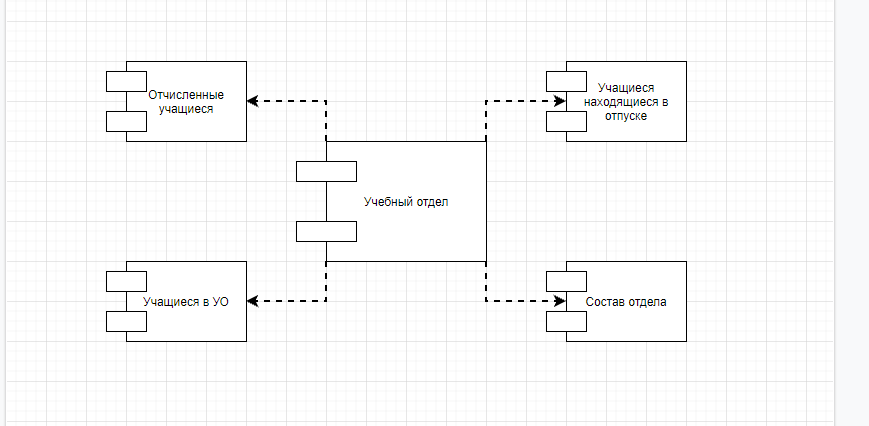


Рисунок 4.7.1. Диаграмма компонентов.

## 4.8. Развёртывания

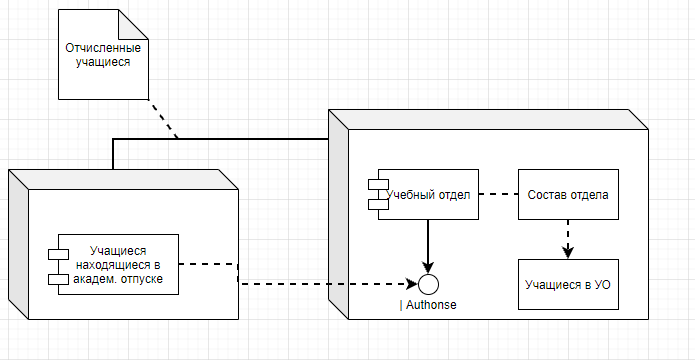


Рисунок 4.8.1. Диаграмма развертывания.

# 5. Проектирование интерфейса

## 5.1 Общая инфраструктура пользовательского интерфейса

## 5.2. Проектирование окон

На рисунках приведены скриншоты окон программы.

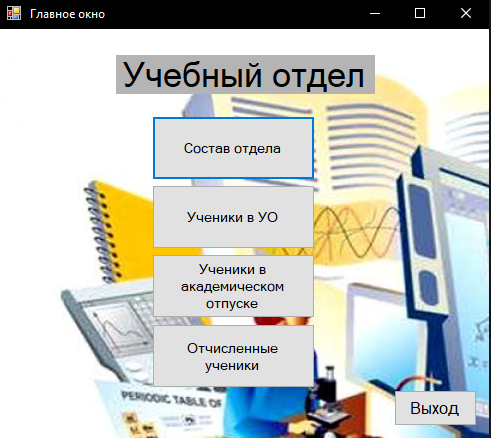


Рисунок 5.2.1. Интерфейс программы

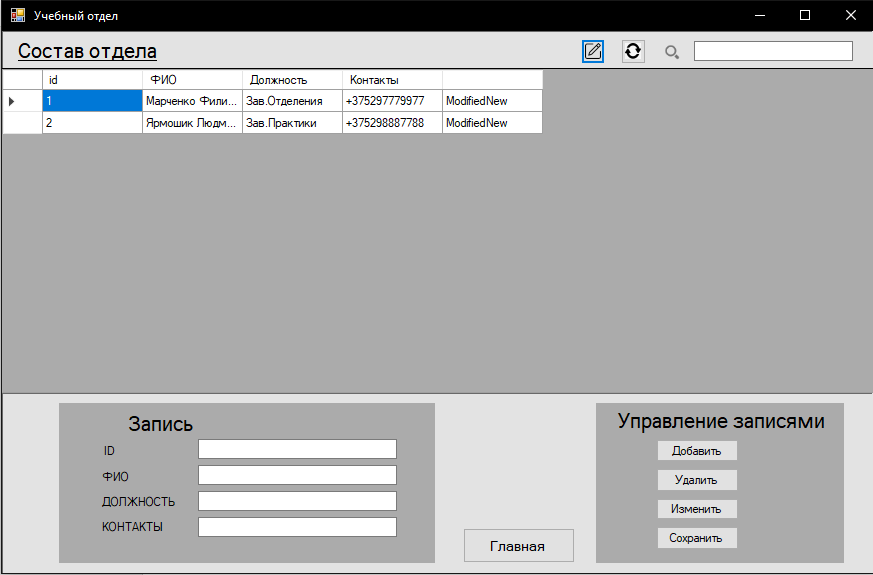


Рисунок 5.2.2. Вкладка «Состав отдела»

Все остальные формы одинаковые и не нуждаются в проектировании

**Листинг программных модулей**

**Листинг 1 - Модуль Program.cs**

{===================================================================================}

{*Модуль* Program.cs*}*

*{гр. П-16}*

*{Разработчик: Мозоль П.В.}*

*{Модифицирован: 07 декабря 2022}*

*{-----------------------------------------------------------------------------------}*

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Учебный\_отдел

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

# 6. Кодирование модулей системы

Применяя визуальную среду провел кодирование модулей программного продукта. Создал первоначальный интерфейс пользователя, который приведен на рисунке 1. Ниже приведен листинг модуля Form1.Designer.cs.

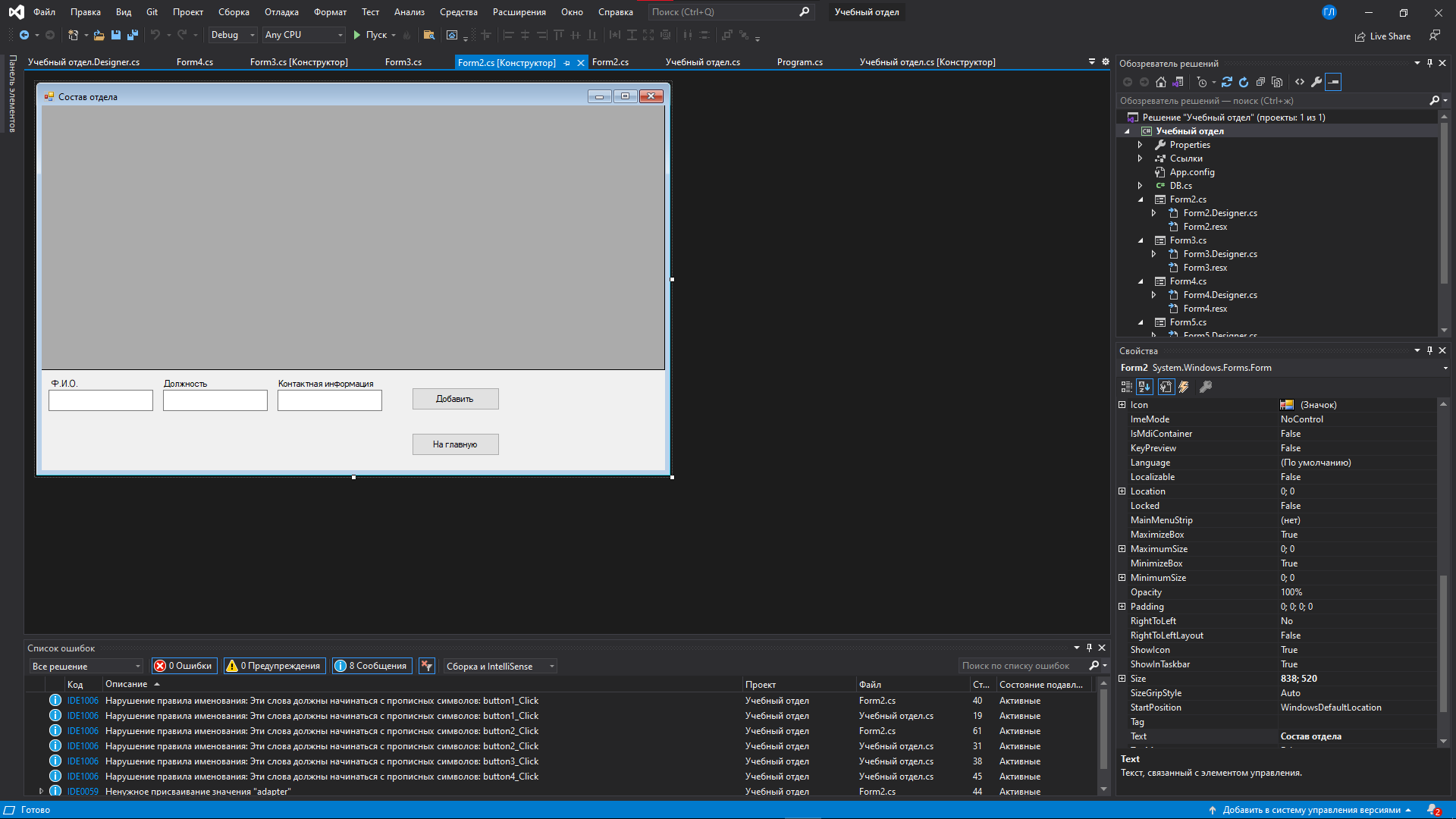


Рисунок 1. Создание интерфейса

**Листинг программных модулей**

**Листинг 1 - Модуль Form1.Designer.cs**

{===================================================================================}

{*Модуль* Form1.Designer.cs*}*

*{гр. П-15}*

*{Разработчик: Лосацкий Г. О.}*

*{Модифицирован: 22 ноября 2021}*

*{-----------------------------------------------------------------------------------}*

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

namespace Учебный\_отдел

{

partial class Form1

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.label1 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.button3 = new System.Windows.Forms.Button();

this.button4 = new System.Windows.Forms.Button();

this.SuspendLayout();

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Times New Roman", 36F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label1.ForeColor = System.Drawing.SystemColors.ActiveCaptionText;

this.label1.Location = new System.Drawing.Point(60, 9);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(353, 55);

this.label1.TabIndex = 0;

this.label1.Text = "Учебный отдел";

//

// button1

//

this.button1.Location = new System.Drawing.Point(140, 79);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(165, 60);

this.button1.TabIndex = 1;

this.button1.Text = "Состав отдела";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.Location = new System.Drawing.Point(140, 147);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(165, 60);

this.button2.TabIndex = 2;

this.button2.Text = "Ученики в УО";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// button3

//

this.button3.Location = new System.Drawing.Point(140, 213);

this.button3.Name = "button3";

this.button3.Size = new System.Drawing.Size(165, 60);

this.button3.TabIndex = 3;

this.button3.Text = "Ученики в академ отпуске";

this.button3.UseVisualStyleBackColor = true;

this.button3.Click += new System.EventHandler(this.button3\_Click);

//

// button4

//

this.button4.Location = new System.Drawing.Point(140, 279);

this.button4.Name = "button4";

this.button4.Size = new System.Drawing.Size(165, 60);

this.button4.TabIndex = 4;

this.button4.Text = "Отчисленные ученики";

this.button4.UseVisualStyleBackColor = true;

this.button4.Click += new System.EventHandler(this.button4\_Click);

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(467, 354);

this.Controls.Add(this.button4);

this.Controls.Add(this.button3);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Controls.Add(this.label1);

this.Name = "Form1";

this.Text = "Главное окно";

this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.Form1\_FormClosed);

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Button button3;

private System.Windows.Forms.Button button4;

}

Применяя визуальную среду провел кодирование дополнительных модулей программной системы. Добавил основные компоненты на форму, а также обработчики событий для вкладок «Состав отдела» и функционал вкладок. Ниже приведены скриншоты вкладок и листинг обработчиков событий для вкладок.

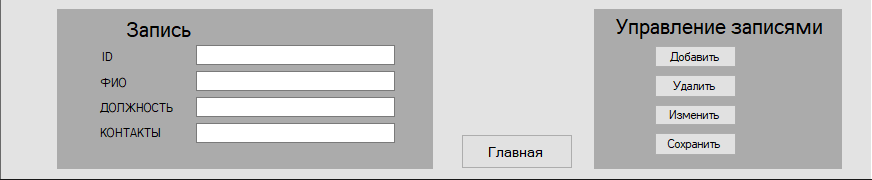


Рисунок 1. Добавление поля



Рисунок 2. Создание кнопки возврата на главное меню

**Листинг программных модулей**

**Листинг 1 - Модуль Form1.cs**

{===================================================================================}

{*Модуль* Form1.cs*}*

*{гр. П-16}*

*{Разработчик: Мозоль П.В.}*

*{Модифицирован: 11 декабря 2022}*

*{-----------------------------------------------------------------------------------}*

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Учебный\_отдел

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("Select \* FROM `sostav\_otdela`", db.GetConnection());

//Заносим полученные данные в таблицу, если есть хоть 1 строка(запись) то тогда появляется нужное окно

adapter.SelectCommand = command;

adapter.Fill(table);

dvg1.DataSource = table;

}

private void Form2\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("INSERT INTO `sostav\_otdela`(`Ф.И.О`, `Должность`, `Контактная информация`) VALUES (@Name, @Surname, @patronymic)", db.GetConnection());

command.Parameters.Add("@Name", MySqlDbType.VarChar).Value = textBox1.Text;

command.Parameters.Add("@Surname", MySqlDbType.VarChar).Value = textBox2.Text;

command.Parameters.Add("@patronymic", MySqlDbType.VarChar).Value = textBox3.Text;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Добавление прошло успешно");

else

MessageBox.Show("Что-то пошло не так, попробуй позже");

//Закрываем базу данных

db.closeConnection();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 f = new Form1();

f.Show();

}

}

}

Используя разработанную структуру программы проверил правильность построения модульной структуры и выполнения их работы, исправил ошибки.

**Листинг 1 - Модуль Form1.cs**

{===================================================================================}

{*Модуль* Form1.cs*}*

*{гр. П-16}*

*{Разработчик: Мозоль П.В.}*

*{Модифицирован: 10 декабря 2022}*

*{-----------------------------------------------------------------------------------}*

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Учебный\_отдел

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("Select \* FROM `sostav\_otdela`", db.GetConnection());

//Заносим полученные данные в таблицу, если есть хоть 1 строка(запись) то тогда появляется нужное окно

adapter.SelectCommand = command;

adapter.Fill(table);

dvg1.DataSource = table;

}

private void Form2\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("INSERT INTO `sostav\_otdela`(`Ф.И.О`, `Должность`, `Контактная информация`) VALUES (@Name, @Surname, @patronymic)", db.GetConnection());

command.Parameters.Add("@Name", MySqlDbType.VarChar).Value = textBox1.Text;

command.Parameters.Add("@Surname", MySqlDbType.VarChar).Value = textBox2.Text;

command.Parameters.Add("@patronymic", MySqlDbType.VarChar).Value = textBox3.Text;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Добавление прошло успешно");

else

MessageBox.Show("Что-то пошло не так, попробуй позже");

//Закрываем базу данных

db.closeConnection();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 f = new Form1();

f.Show();

}

}

}

# 7. Реализация и тестирование

Таблица 1 – Тестирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Выполненные действия | Шаги | Ожидаемый результат |
| 1 | Сворачивание приложения | 1. Запустить программа  2. Нажать кнопку «Свернуть» на устройстве. | 1. Программа запустится  2. Программа свернётся |
| 2 | Открытие свёрнутого приложения | 1. Нажать кнопку «Свернуть» на устройстве.  2. Выбрать программу IBM Editor | 1. Откроется список со свёрнутыми приложениями  2. Программа откроется. |
| 3 | Открытие программа | 1. Запустите программа | 1. Программа запустится |
| 4 | Закрытие программы | 1. Запустите программу  2. Дважды нажать кнопку «Назад» | 1. Программа запустится  2. Программа закроется |
| 5 | Проверка режима нескольких окон | 1. Запустить программа  2. Удержать кнопку «Свернуть» на устройстве | 1. Программа запустится  2. Программа растянется на половину экрана. |
| 6 | Проверка работы режима «В отдельном окне» | 1. Запустить программу  2. Включить режим «В отдельном окне» | 1. Программа запуститься  2. Программа перейдёт в режим «В отдельном окне» |

Таблица 2 – Тестирование функционала

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | Ввод текста русского алфавита | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Ввести текст на русском языке | 1. Программа запуститься  2. Текст отобразился, стабильная работа программы |
| 8 | Ввод текста английского алфавита | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Переключиться на английскую раскладку  4. Ввести текст на английском языке | 1. Программа запуститься  2. Текст отобразился, стабильная работа программы |
| 9 | Ввод специальных символов | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Ввести специальные символы | 1. Программа запуститься  2. Специальные символы отобразился, стабильная работа программы |
| 10 | Вырезать выделенный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Выделить произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Вырезать» | 1. Программа запустится  2. Произойдет вырезка выделенного текста |
| 11 | Копировать выделенный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Выделить произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Копировать» | 1. Программа запустится  2. Произойдет копирование выделенного текста |
| 12 | Вставить скопированный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Предварительно выделить и скопировать произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Вставить» | 1. Программа запустится  2. Произойдет вставка предварительно скопированного текста |
| 13 |  |  |  |
| 14 | Выделить весь текст | 1. Запустить программу  2. Ввести произвольный текст  3. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Выделить всё» | 1. Программа запустится  2. Произойдёт выделение всего текста в документе |
| 15 | Просмотр справки | 1. Запустить программу  2. Нажать на кнопку «Справка»  3. Выбрать в списке кнопку «Просмотреть справку» | 1. Программа запустится  2. Откроется руководство пользователя по использованию программы |
| 16 | Просмотр О программе | 1. Запустить программу  2. Нажать на кнопку «Справка»  3. Выбрать в списке кнопку «О программе» | 1. Программа запустится  2. Откроется окно «О программе», содержащие информацию о версии приложения и тд. |

# 8. Оценка стоимости

# 9. Заключение

# 10. Список использованных источников