МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ.02 «Осуществление интеграции программных модулей»**

Подготовил студент

Ермаков Павел Андреевич

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Директор

КОГПОБУ «Слободской колледж педагогики и социальных отношений  
 Наименование организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/О.М. Шеренцова

Подпись расшифровка

М.П.

Руководитель практики от организации

Подпись расшифровка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Калинин Арсений Олегович

Подпись расшифровка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Калинин Арсений Олегович

Руководители практики от колледжа:

Калинин Арсений Олегович

2024 год

**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ
2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
3. РУКОВОДСТВО ОПЕРАТОРА
4. РАБОТА В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ
5. РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ
6. ОТЛАДКА ПРОГРАММНОГО МОДУЛЯ
7. ЗАКЛЮЧЕНИЕ
8. ПРИЛОЖЕНИЯ

**ВВЕДЕНИЕ**

Практику по данному модулю я проходил в КОГПОБУ «Слободской колледж педагогики и социальных отношений» с 6 по 17 мая. Целью данного модуля практики является осуществление интеграции программных модулей разрабатываемых нами систем.

Некоторые задачи практики:

* Создать приложение для медицинской лаборатории
* Создать приложение для телефонного справочника

**АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Медицинская лаборатория предоставляет специализированные услуги по проведению исследований биоматериалов для поликлиник города. Было сделано множество диаграмм и техническое задание.

ERD-Диаграмма(Рис.1)

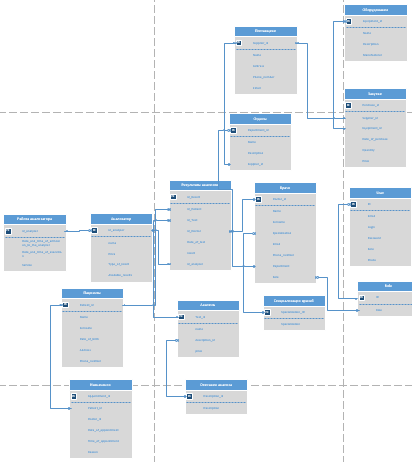
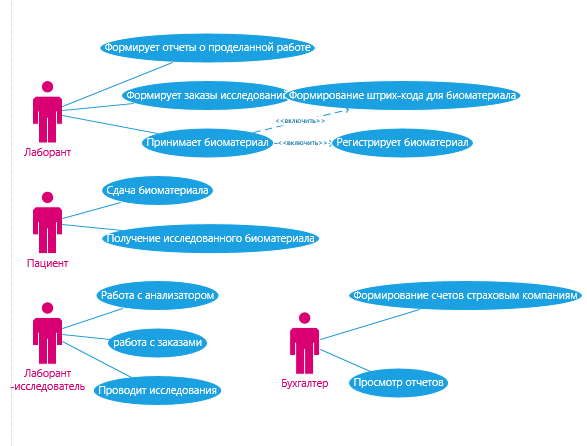


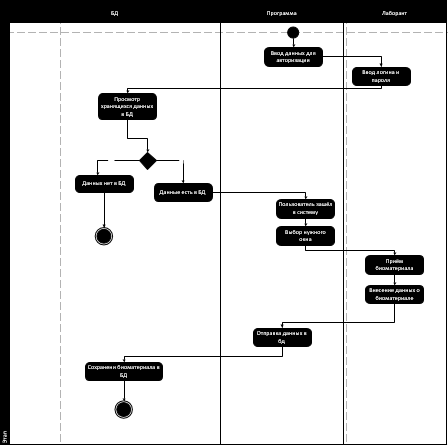
Рис.1

Диаграмма вариантов использования(Рис.2)



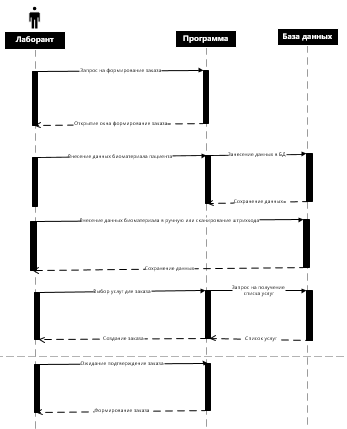
(Рис.2)

Диаграмма деятельности(Рис.3)



(Рис.3)

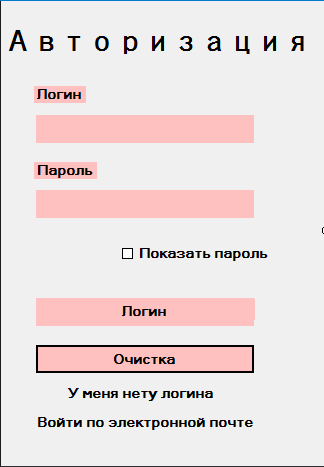
Диаграмма последовательности(Рис.4)



(Рис.4)

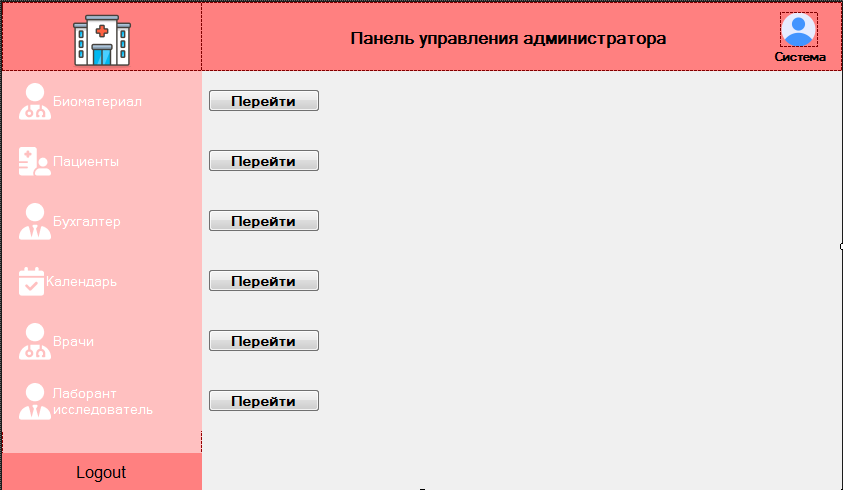
**РУКОВОДСТВО ОПЕРАТОРА**

Окно авторизации представлено двумя полями для ввода логина и пароля (Рис.5)



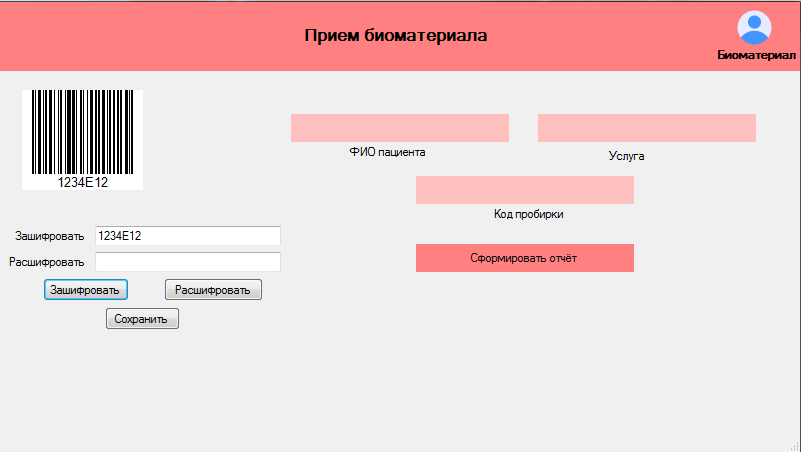
(Рис.5)

Главное окно(Рис.6)



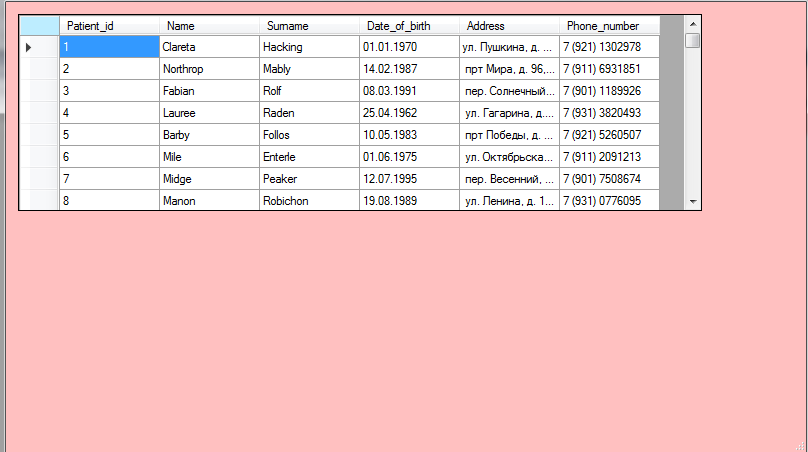
(Рис.6)

Окно приема биоматериала(Рис.7)



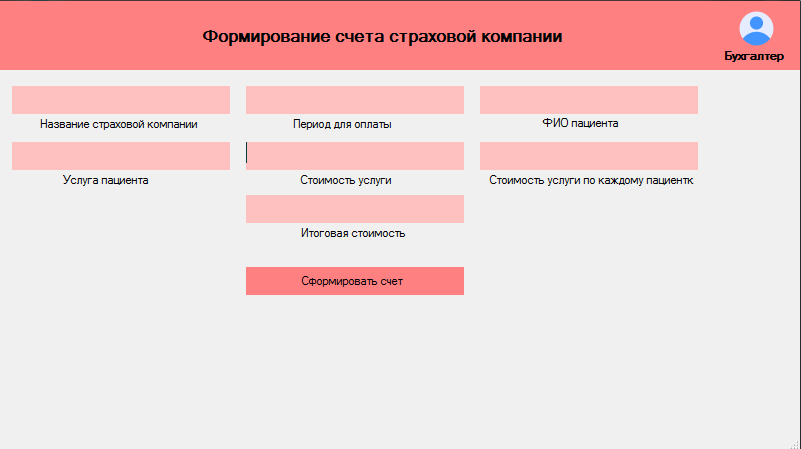
(Рис.7)

Окно списка пациентов(Рис.8)



(Рис.8)

Окно для формирования счета страховым компаниям(Рис.9)



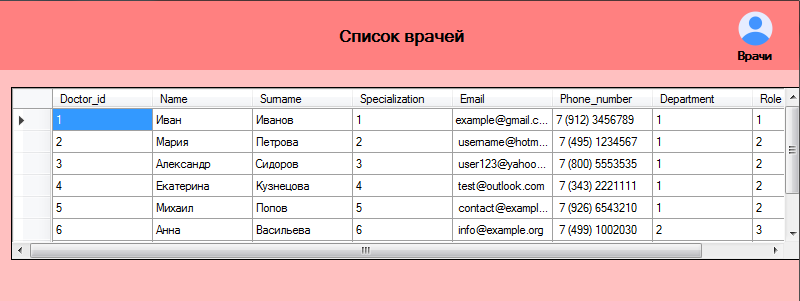
(Рис.9)

Календарь(Рис.10)



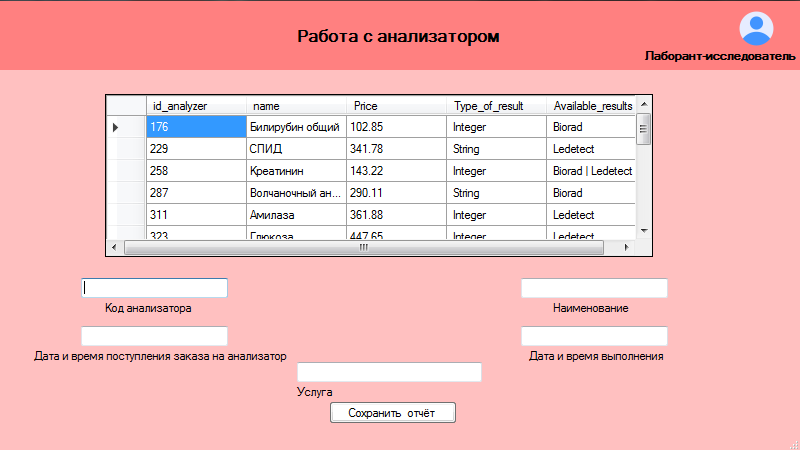
(Рис.10)

Окно списка врачей(Рис.11)



(Рис.11)

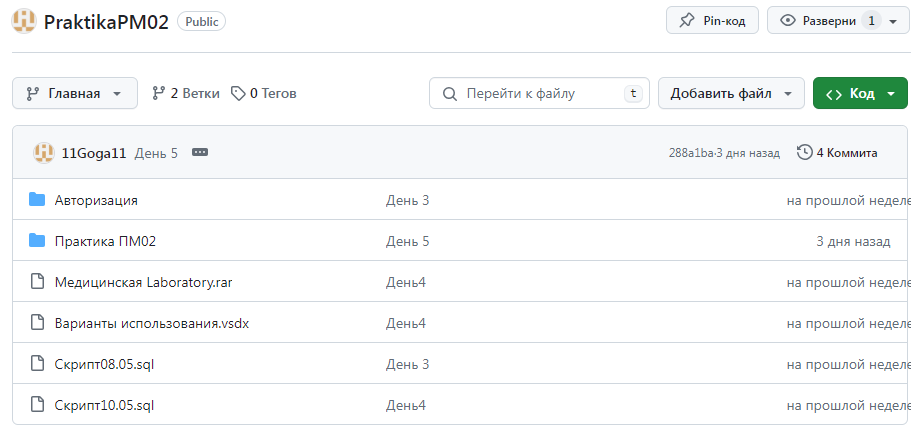
Окно работы с анализатором(Рис.10)



(Рис.12)

**РАБОТА В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ**

За все время практики я работал с системой контроля версий GitHub. Я создал несколько репозиториев, в которые загружал все выполняемые мной задания: (Рис.12)



(Рис.12)

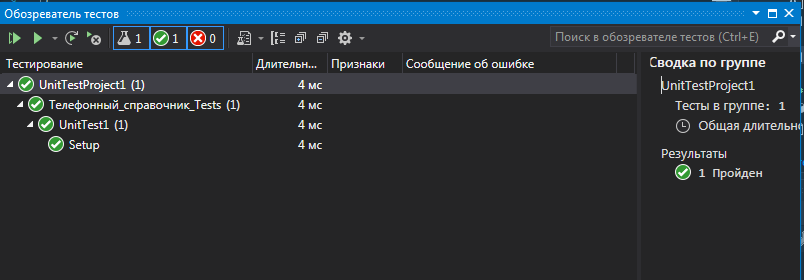
**РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ**

**Тестирование программы Телефонный справочник**

**Функциональное тестирование:**

**Цель:** Проверка функциональности приложения «Телефонный справочник» для обеспечения соответствия требованиям.

Для телефонного справочника я создал Unit.Test



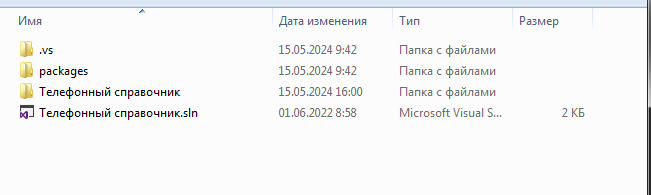
(Рис.13)

Код Unit.Test

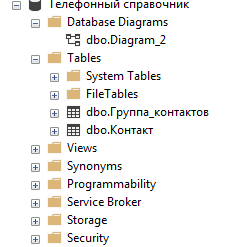
|  |
| --- |
| using Microsoft.VisualStudio.TestTools.UnitTesting;  using System;  using System.Collections.Generic;  using Телефонный\_справочник;  namespace Телефонный\_справочник\_Tests  {  [TestClass]  public class UnitTest1  {  private Mock<IContactRepository> \_mockContactRepository;  private PhoneBookService \_phoneBookService;  [TestMethod]  public void Setup()  {  \_mockContactRepository = new Mock<IContactRepository>();  \_phoneBookService = new PhoneBookService(\_mockContactRepository.Object);  }  public void AddContact\_WhenCalled\_ShouldAddContactToRepository()  {  // Arrange  var contact = new Contact { Name = "John Doe", PhoneNumber = "1234567890" };  // Act  \_phoneBookService.AddContact(contact);  }  public void GetAllContacts\_WhenCalled\_ShouldReturnAllContactsFromRepository()  {  // Arrange  var contacts = new List<Contact>  {  new Contact { Name = "John Doe", PhoneNumber = "1234567890" },  new Contact { Name = "Jane Doe", PhoneNumber = "0987654321" }  };  \_mockContactRepository.Setup((contacts));  // Act  var result = \_phoneBookService.GetAllContacts();  // Assert  Assert.AreEqual(contacts, result);  }  }  internal class Times  {  internal static object Once;  }  internal class Mock<T>  {  internal IContactRepository Object;  internal void Setup(Func<object, object> p)  {  throw new NotImplementedException();  }  internal void Setup(List<Contact> contacts)  {  throw new NotImplementedException();  }  internal void Verify(Func<object, object> p, object once)  {  throw new NotImplementedException();  }  } |

Тестирование приложения “Телефонный справочник”

1. Подготовка к тестированию

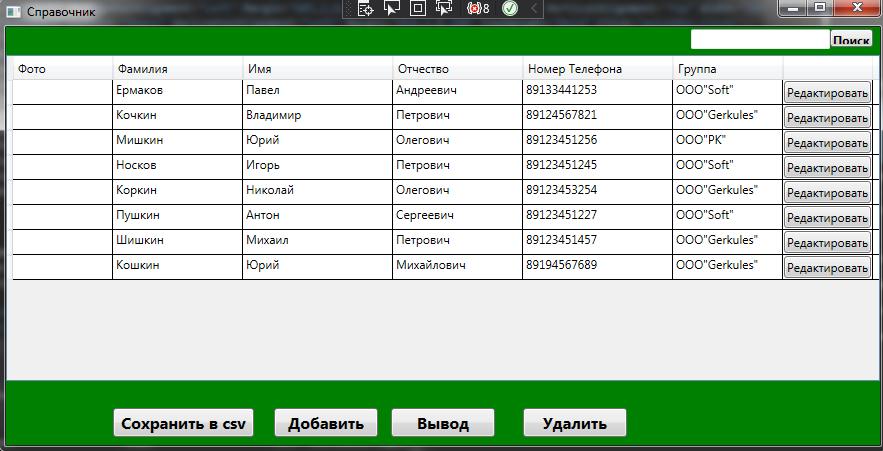


(Рис.14)



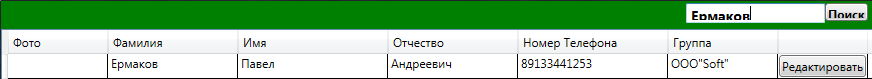
(Рис.15)

1. Проверка главной страницы



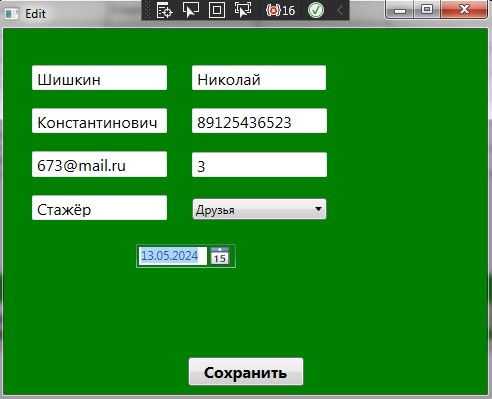
(Рис.16)

1. Проверка поиска контактов

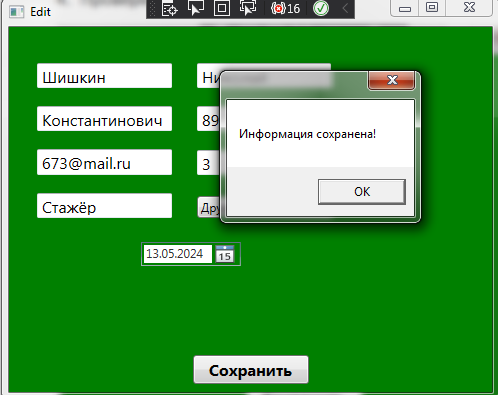


(Рис.17)

1. Проверка добавления контакта



(Рис.18)

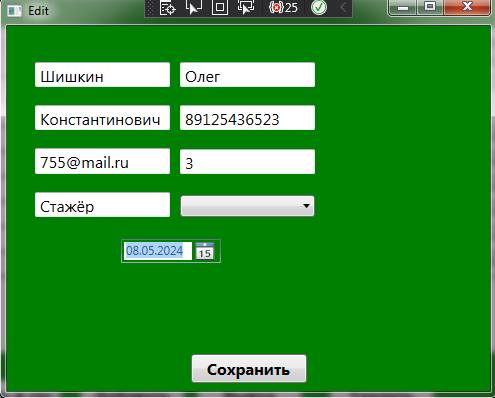


(Рис.19)

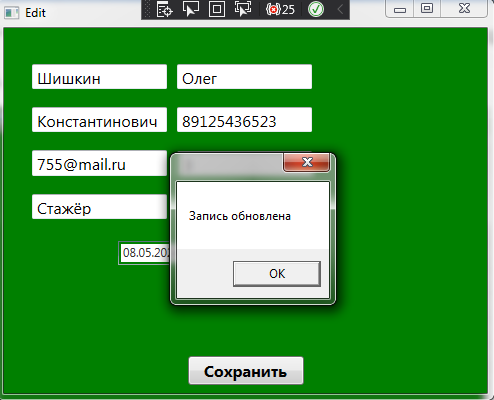


(Рис.20)

1. Проверка редактирования контакта

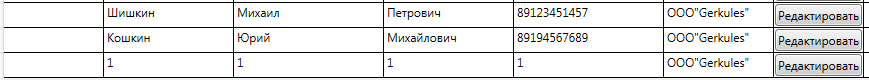


(Рис.21)

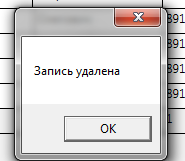


(Рис.22)

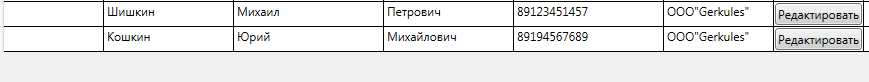
1. Проверка удаления контакта



(Рис.23)

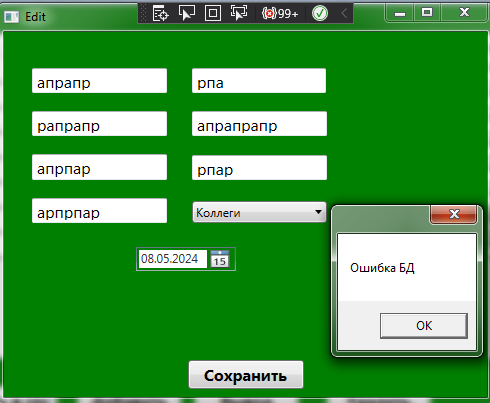


(Рис.24)



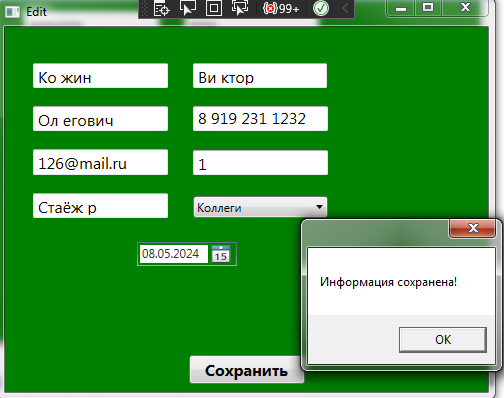
(Рис.25)

1. Проверка состояния ошибки



(Рис.26)

1. Проверка работы с пробелами и специальными символами

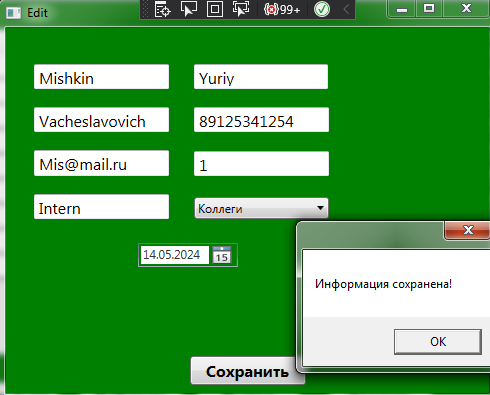


(Рис.27)



(Рис.28)

1. Проверка работы приложения в разных языковых настройках



(Рис.29)

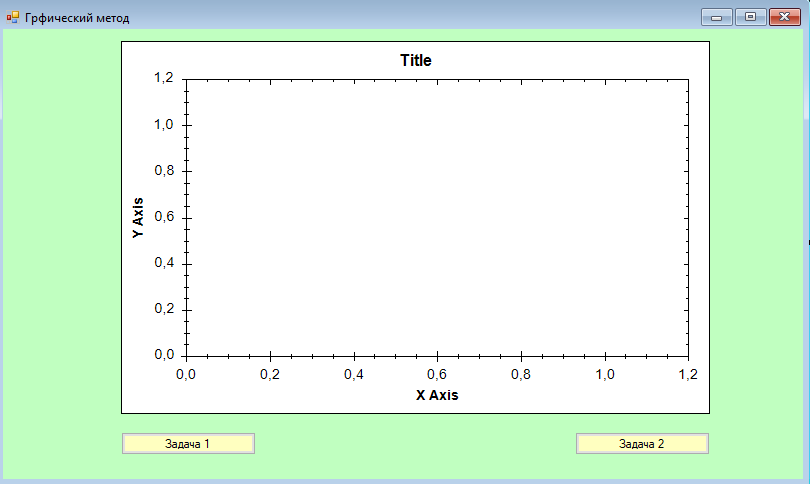


(Рис.30)

[Графический метод/ЗЛП И ТЗ](https://github.com/HoromiUzumaki/Practice_PM02/tree/master/%D0%93%D1%80%D0%B0%D1%84.%20%D0%BC%D0%B5%D1%82%D0%BE%D0%B4/%D0%97%D0%9B%D0%9F%20%D0%98%20%D0%A2%D0%97).

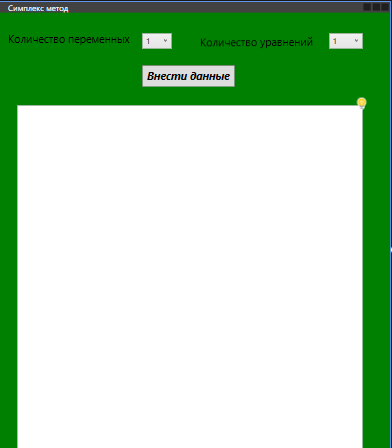
Я написал два приложения для решения задач.

1. Приложение для решения задач линейным способом

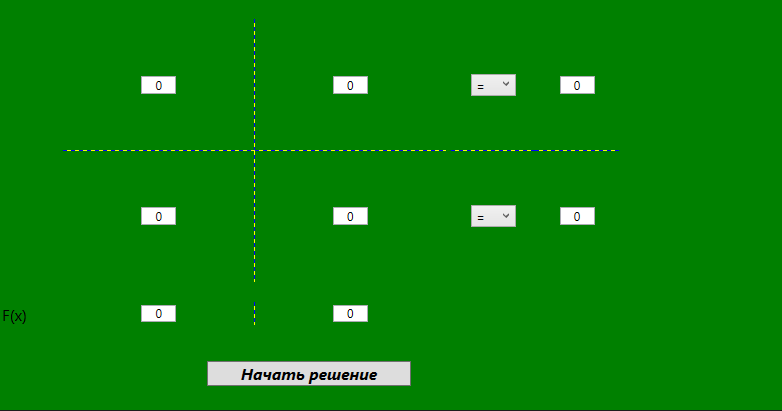


(Рис.31)

1. Приложение для решения задач Simplex методом



(Рис.32)



(Рис.33)

**ЗАКЛЮЧЕНИЕ**

Учебная практика дала мне ценный опыт, который помог расширить мои профессиональные знания и навыки. Во время практики, я изучил осуществление интеграции программных модулей и вспомнил, то чему учился на протяжении учебного года.