**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
(тема курсовой работы)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе  
по дисциплине

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
(полное наименование учебной дисциплины)

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (индекс группы) | (подпись и дата) | (инициалы и фамилия) |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (должность и ученая степень) | (подпись и дата) | (инициалы и фамилия) |

ТУЛА 2024

Бланк заданий

**Оглавление**

[ВВЕДЕНИЕ 4](#_Toc156771849)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc156771850)

[2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc156771851)

[**2.1 Назначение разработки** 7](#_Toc156771852)

[**2.2. Требования к программе** 7](#_Toc156771853)

[**2.3 Стадии и этапы разработки** 8](#_Toc156771854)

[3. МОДЕЛИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ НА ОСНОВЕ ПРИНЦИПОВ ООП 10](#_Toc156771855)

[4. ОПИСАНИЕ ПРОГРАММЫ 14](#_Toc156771856)

[**4.1 Функциональное назначение** 14](#_Toc156771857)

[**4.2 Описание логической структуры** 14](#_Toc156771858)

[**4.3 Входные и выходные данные** 14](#_Toc156771859)

[5. ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ ПРОГРАММЫ 15](#_Toc156771860)

[**5.1 Назначение программы** 15](#_Toc156771861)

[**5.2 Выполнение программы** 15](#_Toc156771862)

[ЗАКЛЮЧЕНИЕ 19](#_Toc156771863)

[Список использованной литературы 20](#_Toc156771864)

[Приложение А – Класс room 21](#_Toc156771865)

[Приложение Б – Класс schedule 25](#_Toc156771866)

[Приложение В – Класс MyForm 27](#_Toc156771867)

[Приложение Г – Исполняемый файл MyForm 55](#_Toc156771868)

# ВВЕДЕНИЕ

Данная курсовая работа направлена на анализ предложенной предметной области, разработку базы данных для нее и создание прикладного программного продукта на языке C++ с использованием Windows Forms для построения графического интерфейса.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область, для которой разрабатывается программный продукт, является гостиница, здания, сооружения и номера помещений в ней. В данной области есть ряд особенностей и бизнес-процессов, которые нужно учесть при разработке продукта.

Особенности и бизнес-процессы:

* Бронирование и регистрация: гостиницы предоставляют услуги по бронированию номеров и регистрации постояльцев, что включает в себя сбор и обработку информации о гостях.
* Номерной фонд: необходимость эффективного учета доступных номеров, их категорий, состояния (свободен, занят) и оснащения для удовлетворения потребностей гостей.

Проблемы и способы их решения:

* Бронирование и регистрация: для упрощения процесса регистрации гостей, необходимо внедрить программу, позволяющую в реальном времени забронировать номер в отеле.
* Номерной фонд: для повышения эффективности управления номерным фондом, рекомендуется внедрить систему автоматизированного учета доступности номеров и моментального обновления информации о занятости.

Место разрабатываемого продукта в организационной структуре предприятия: разрабатываемый продукт может интегрироваться в уже действующую организационную структуру. Программа будет помогать портье в автоматизации процесса регистрации гостей в гостинице

Из существующих программ, представленных на рынке, для работы бара, возьмем на рассмотрение: Protel, Amadeus, Cloudbeds

* Protel

Программа предоставляет следующие функции: управление бронированием, работа с номерами, учет услуг, аналитику и отчётность.

* Amadeus

Программа имеет следующие особенности: интеграция с другими сервисами (по типу множества платежных систем и т. д.), интегрированные решения для цепочек отелей, удобный и понятный интерфейс

* Cloudbeds

Программа имеет следующие особенности: гибкость и масштабируемость, интеграция с онлайн бронированием, многоязычность и локализация, а также программа лояльности.

Проанализировав существующие приложения в рассматриваемой предметной области, было принято решение создать программный продукт с функциями просмотра информации о номерах, расписании и управлением бронирования.

**Цель данной курсовой работы** заключается в создании прикладного программного продукта на языке С++ с графическим интерфейсом, который поможет портье быстрее осуществлять регистрацию гостей.

**Задачи курсовой работы**:

1. Разработать логику продукта.
2. Разработать дизайн продукта.
3. Изучить теорию по созданию приложений с графическим интерфейсом на языке С++ с использованием WinForms.
4. Реализовать практическую часть работы.

# 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## **2.1 Назначение разработки**

Исходные данные: предметная область – гостиница.

Тема курсовой работы: Прикладной программный продукт «Гостиничный бизнес. Фонд недвижимого имущества (здания / сооружения, номера помещения)»

Функциональное назначение программы: оптимизация работы портье, реализовав в программе удобную систему бронирования.

## **2.2. Требования к программе**

Функциональные характеристики программы: программа предоставляет следующий функционал:

* Вывод информации о номере
* Бронирование номера
* Вывод расписания заселения номера
* Просмотр занятости номера на главной странице программы

Язык программирования: для разработки программного продукта будет использоваться язык программирования С++, поскольку он предоставляет широкий спектр возможностей, включая объектно-ориентированное программирование, шаблоны, стандартную библиотеку шаблонов (STL) и другие инструменты.

Технические средства: среда разработки VisualStudio 2022 со следующими предустановленными компонентами: «.NET desktop development», «desktop development with C++» , «C++/CLI support», а также библиотека libxl для работы с Excel файлами. Программа будет разрабатываться с использованием ПК с операционной системой Windows 11.

Входные и выходные данные: входные данные программы будут представлять собой базу данных гостиницы. Выходные данные – информация о номерах и их расписании.

## **2.3 Стадии и этапы разработки**

Стадии разработки:

1) Изучение предметной области. Сроки: 06.10.2023 – 15.10.2023.

Содержание работы: изучение бизнес-процессов гостиниц, выяснение основных функциональных требований к программе, изучение работы портье на ресепшене.

2) Изучение программ аналогичного назначения. Сроки: 16.10.2023 – 18.10.2023

Содержание работы: поиск и анализ программ со схожим функционалом.

3) Изучение принципов создание графического интерфейса на языке C++ с использованием Windows Forms. Сроки: 20.10.2023 – 20.11.2023

Содержание работы: изучение методичек и документаций.

4) Создание базы данных в программе Excel. Сроки: 21.11.2023 – 23.11.2023

Содержание работы: создание файла Excel с информацией о номерах и их расписанием.

5) Разработка интерфейса. Сроки: 23.11.2023 – 29.11.2023

Содержание работы: создание дизайна программы.

6) Разработка классов. Сроки: 29.11.2023 – 03.12.2023

Содержание работы: продумывание и разработка классов и их методов.

7) Реализация программной части продукта. Сроки: 03.12.2023– 15.12.2023

Содержание работы: написание программного кода.

8) Тестирование продукта. Сроки: 15.12.2023– 17.12.2023

Содержание работы: тестирование программы на ошибки.

9) Отладка программы. Сроки: 17.12.2023– 21.12.2023

Содержание работы: исправление ошибок, возникших при тестировании продукта.

9) Завершение работы. Сроки: 22.12.2023

Содержание работы: продукт готов к использованию.

# 3. МОДЕЛИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММЫ НА ОСНОВЕ ПРИНЦИПОВ ООП

Продукт состоит из двух частей: база данных (файл Excel) и приложение, позволяющее с ней взаимодействовать.

В файле Excel содержатся 2 листа. Лист 1 содержит следующие поля:

1. Этаж, на котором расположен номер. Будет содержать значение от 1 до 5
2. Номер. Будет содержать значение от 1 до 40.
3. Занятость номера. Будет содержать значение да/нет.
4. Имя. Будет содержать поле «не занят» или фамилию и инициалы гостя.
5. Тип номера. Будет содержать значение «люкс», «президентский» или «обычный».
6. Цена за ночь. Будет содержать значение 6000, 13000 или 27000, зависящее от типа номера.

Лист 2 содержит следующие поля:

1. Номер. Будет содержать значение от 1 до 40.
2. Дата заселения. Будет содержать дату, когда заселится гость.
3. Дата выселения. Будет содержать дату, когда гость выселится.
4. Имя. Будет содержать фамилию и инициалы гостя.

Для работы с данными будут созданы 2 класса, room и schedule.

1) Класс **room**. Представляет объект комнаты в гостиничной системе и включает в себя ряд полей и методов для управления информацией о комнатах и гостях. Основные элементы этого класса:

* Приватные члены класса:

**floor:** Этаж комнаты (тип данных System::String^).

**number:** Номер комнаты (тип данных System::String^).

**busy:** Статус комнаты (занята/свободна) (тип данных System::String^).

**fam:** Фамилия и инициалы гостя (тип данных System::String^).

**type:** Тип номера (тип данных System::String^).

**price:** Цена за номер (тип данных System::String^).

* Публичные методы класса:

**GetStr():** Возвращает строку с информацией о комнате.

**Search(String^ number):** Поиск комнаты по номеру и возвращение информации о ней.

**Color(System::Windows::Forms::Button^ tmp, String^ number)**: Установка цвета кнопки в зависимости от статуса комнаты.

**AddGuest(System::String^ teg, int RowNumber, System::Windows::Forms::Form^)**: Добавление гостя к комнате, если она свободна.

* Приватный метод класса:

**LoadFromExcel(System::String^ excelFilePath, int RowNumber):** Загрузка информации о комнате из файла Excel.

* Конструктор класса:

**room(int RowNumber):** Конструктор, загружающий информацию о комнате из файла Excel по указанному номеру строки.

Класс использует библиотеку libxl для работы с файлами Excel. Он также включает в себя логику для изменения статуса комнаты, добавления гостя, а также загрузки информации о комнате из файла Excel.

2) Класс **schedule**. Представляет объект расписания заселения и выселения гостей в номерах гостиницы. Основные элементы этого класса:

* Приватные члены класса:

**komnata**: Номер комнаты (тип данных System::String^).

**date**: Дата заселения (тип данных System::String^).

**famil**: Фамилия и инициалы гостя (тип данных System::String^).

**date1**: Дата выселения (тип данных System::String^).

* Публичные методы класса:

**GetInfo():** Возвращает строку с информацией о заселении и выселении гостя в комнате.

**GetRoom(String^ number):** Возвращает информацию о заселении и выселении для указанной комнаты.

* Приватные методы класса:

**LoadFromExcel(System::String^ excelFilePath, int RowNumber)**: Загружает информацию о заселении и выселении гостя из файла Excel.

**ExcelDateToString(double excelDate)**: Конвертирует числовое представление даты Excel в строку формата "dd.MM.yyyy".

* Конструктор класса:

**schedule(int RowNumber):** Конструктор, загружающий информацию о расписании заселения и выселения из файла Excel по указанному номеру строки.

Класс использует библиотеку libxl для работы с файлами Excel и содержит логику для загрузки и предоставления информации о заселении и выселении гостей в комнатах гостиницы.

В результате проектирования была создана диаграмма классов, представленная на рисунке 1.



Рисунок 1 – UML диаграмма класса

# 4. ОПИСАНИЕ ПРОГРАММЫ

## **4.1 Функциональное назначение**

Создаваемая программа предназначена для оптимизации работы портье за ресепшеном, быстрой регистрации гостей в отеле.

В качестве среды реализации выбрана среда VisualStudio 2022.

База данных хранится в виде файла .xls. При запуске программы вся база данных считывается из файла и выводится в таблицы программы. Так же имеется функция обновления базы данных вручную через файл или через программу.

Помимо этого, программа имеет очень низкие требования к среде выполнения. Необходимо только скачать и установить библиотеку libxl, для работы с файлами Excel.

## **4.2 Описание логической структуры**

Программа имеет следующую структуру:

Класс room представляет комнату в гостинице, загружает информацию из файла Excel, предоставляет методы для получения подробностей о комнате, поиска по номеру, изменения цвета кнопки в зависимости от занятости, а также добавления гостя с обновлением статуса комнаты и сохранением данных в файле Excel. Код представлен в приложении А.

Класс schedule представляет гостевой график, загружает информацию из файла Excel и предоставляет методы для получения подробностей о гостях, включая даты заселения и выселения, номер комнаты и имя гостя. Код представлен в приложении Б.

## **4.3 Входные и выходные данные**

Входными данными программы являются база данных и действия пользователя.

Выходными данные являются выводимая на экран при нажатии на кнопки информация о номерах и их расписании.

База данных хранится в файле. Данные указываются в следующем порядке: этаж, номер, занят/нет, имя, тип номера и цена за ночь. Пример 1 строки из базы данных:

1 1 да Игнатьев А. В. Президентский 27000

Расписание номеров будет храниться в следующем порядке: номер, дата заселения, имя, дата выселения. Пример:

1 15.01.2024 Токаев К. Е. 17.01.2024

# 5. ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ ПРОГРАММЫ

## **5.1 Назначение программы**

Программа предназначена для автоматизации работы портье и быстром заселении гостей отеля в номера.

В качестве исполняющей среды программа может использовать операционные системы семейства Windows не ниже версии 7.

## **5.2 Выполнение программы**

Для запуска продукта требуется иметь ПК с установленным ПО, которое описано в пункте 2.2

После запуска программы выводится окно, на котором нужно ввести пароль, чтобы зайти в программу. Окно авторизации показано на рисунке 2.

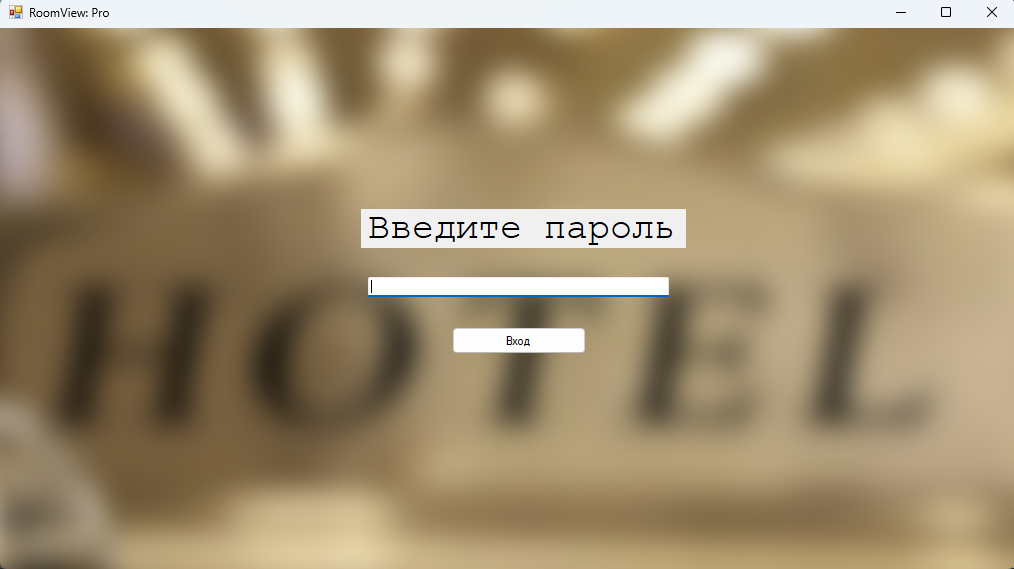


Рисунок 2 – окно авторизации

Если неправильно ввести пароль, программа выдаст ошибку (рис. 3).

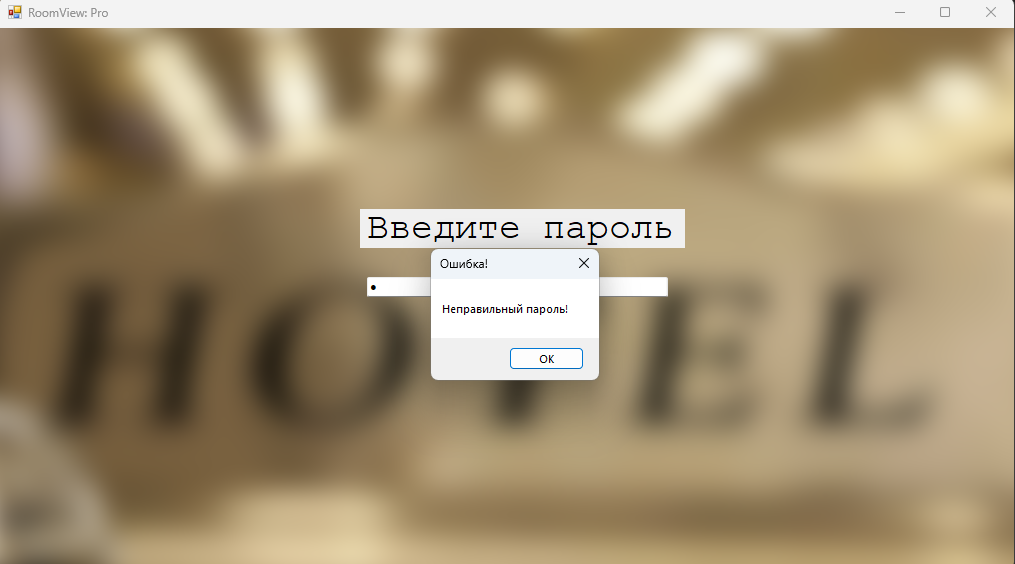


Рисунок 3 – окно ошибки авторизации

После авторизации выводится окно, на котором показаны все номера в отеле. Окрашенные красным цветом – занятые номера, синие – соответственно, свободные (рис. 4).

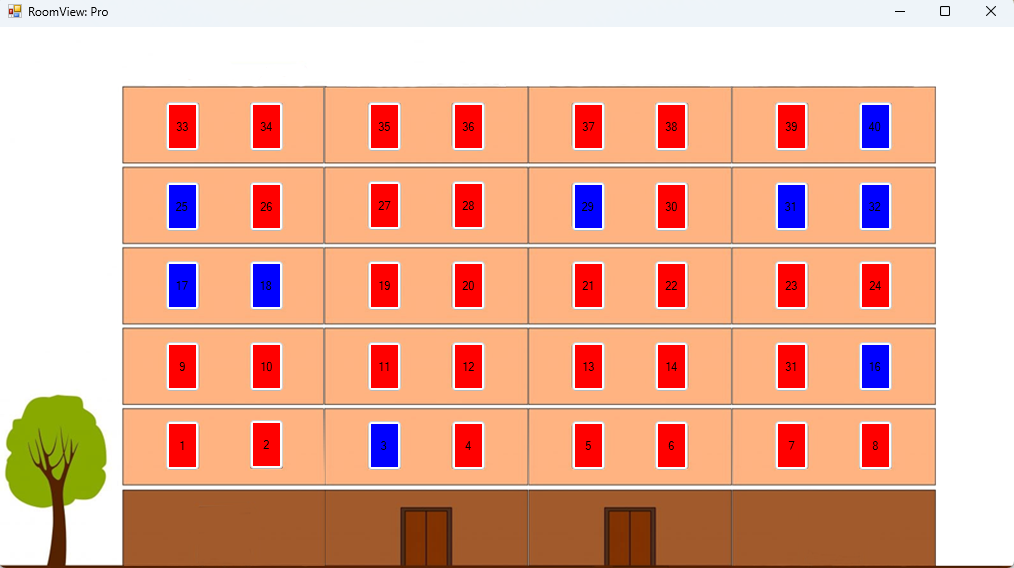


Рисунок 4 – основное окно программы

При нажатии на любую кнопку открывается панель с информацией о номере (рис. 5).

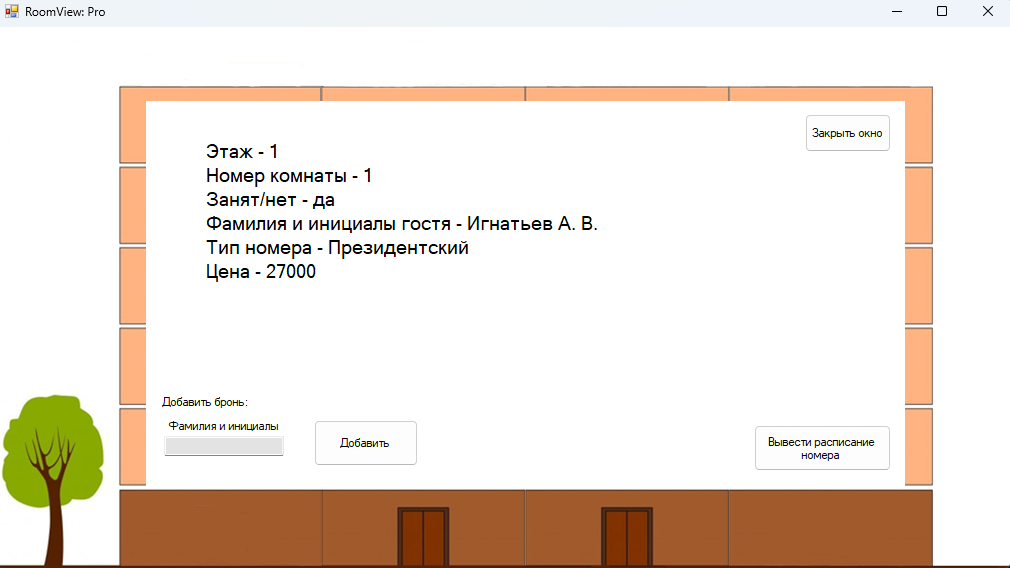


Рисунок 5 – панель с информацией о номере

При нажатии на кнопку «Вывести расписание номера», открывается новая панель, с расписанием номера (рис. 6).

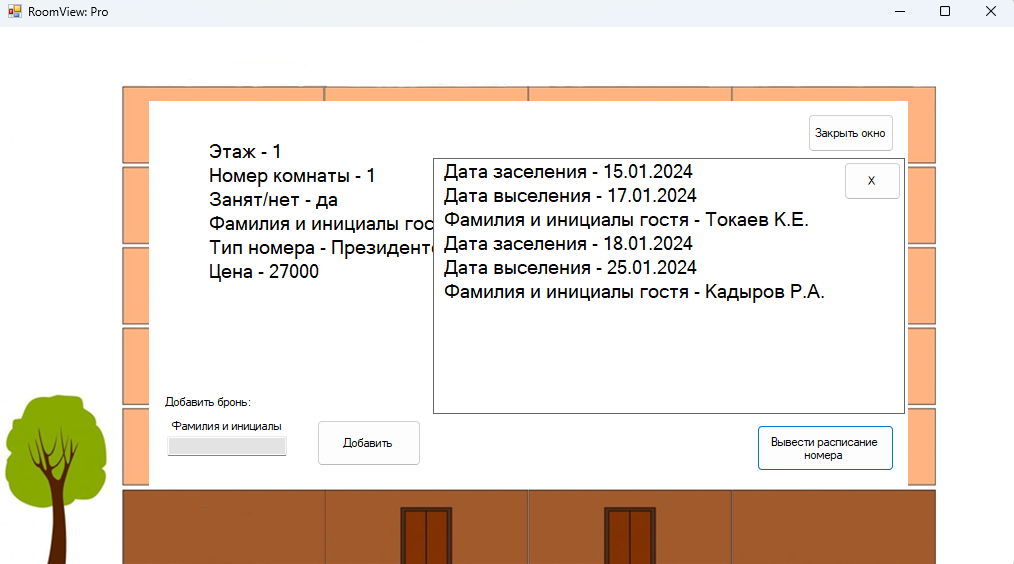


Рисунок 5 – панель с расписанием номера

При нажатии на кнопку «Добавить», если номер занят, появляется сообщение (рис. 6):

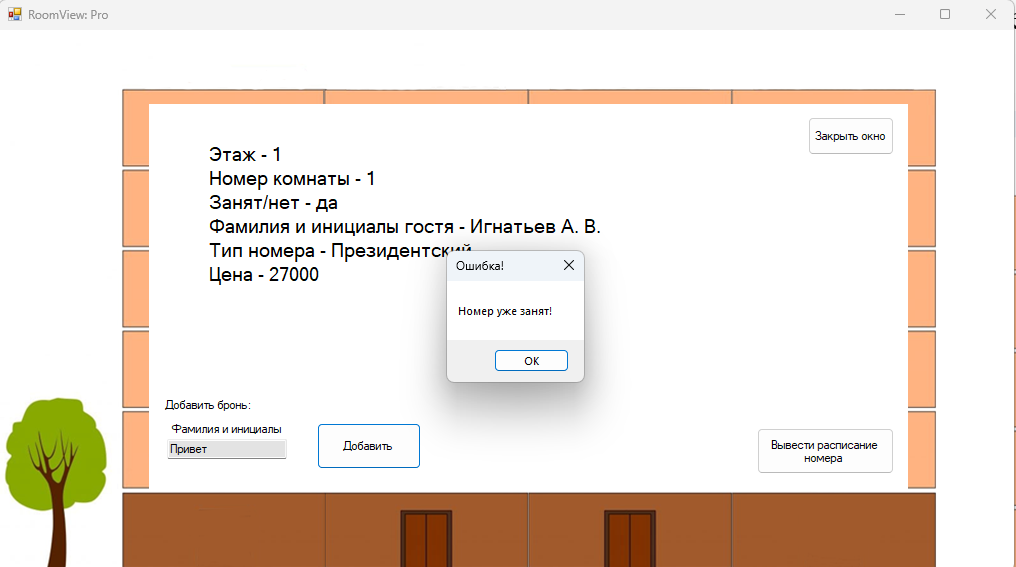


Рисунок 6 – сообщение ошибки

Если же номер не занят, сообщение следующее (рис. 7):

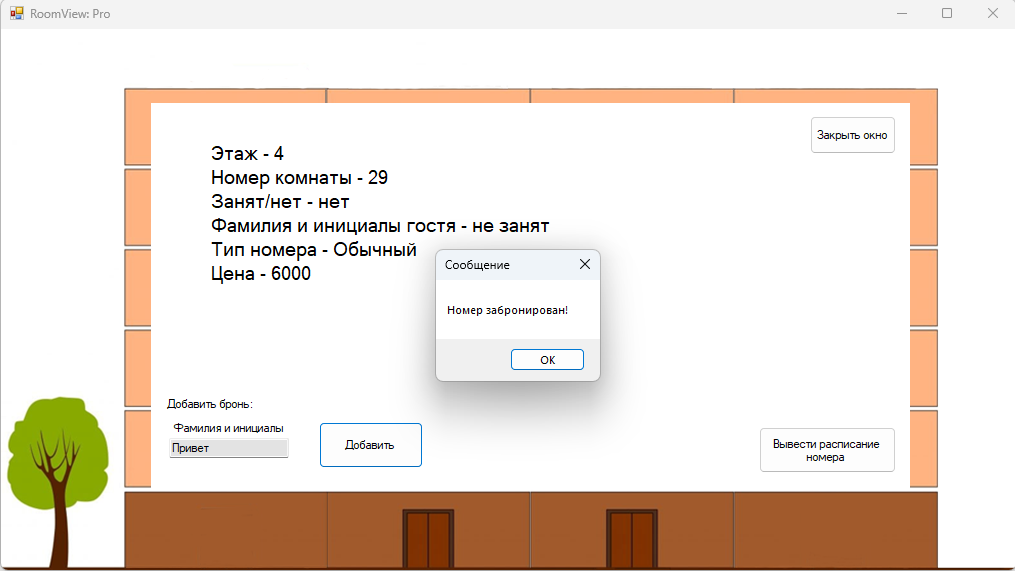


Рисунок 7 – сообщение о бронировании номера

## **5.3 Описание контрольного примера**

Контрольный пример будет проводиться на базе данных, написанной для гостиницы. Добавим гостя в свободный номер 3 (рис. 8):

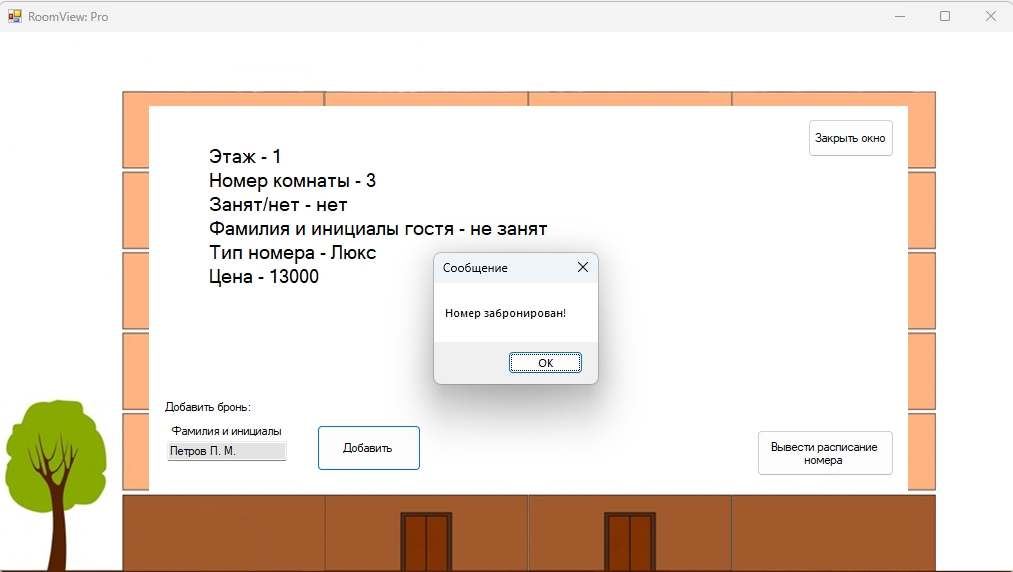


Рисунок 8 – бронирование номера

Проверяем, забронировался ли номер 3 (рис. 9, 10).

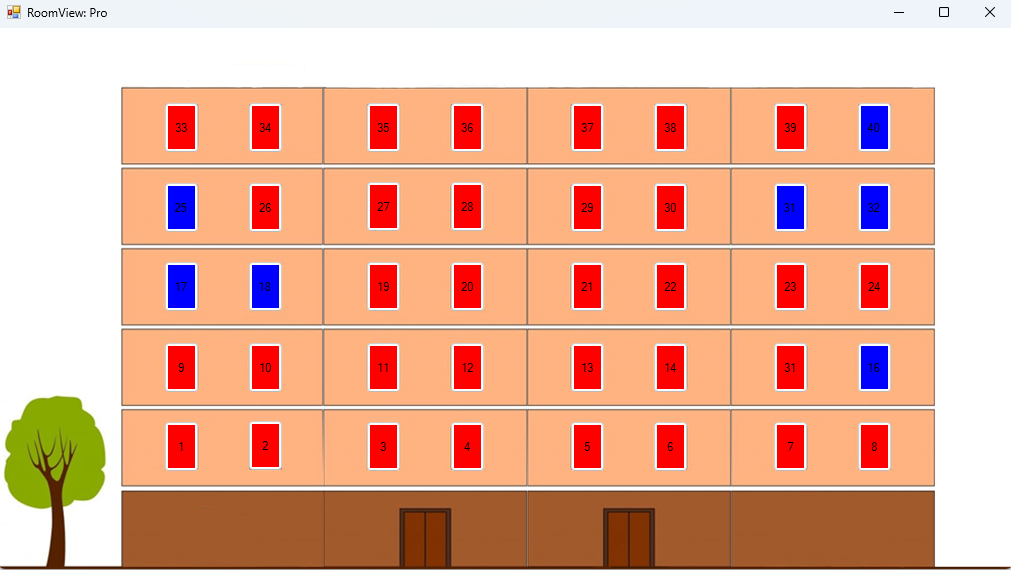


Рисунок 9 – окно программы

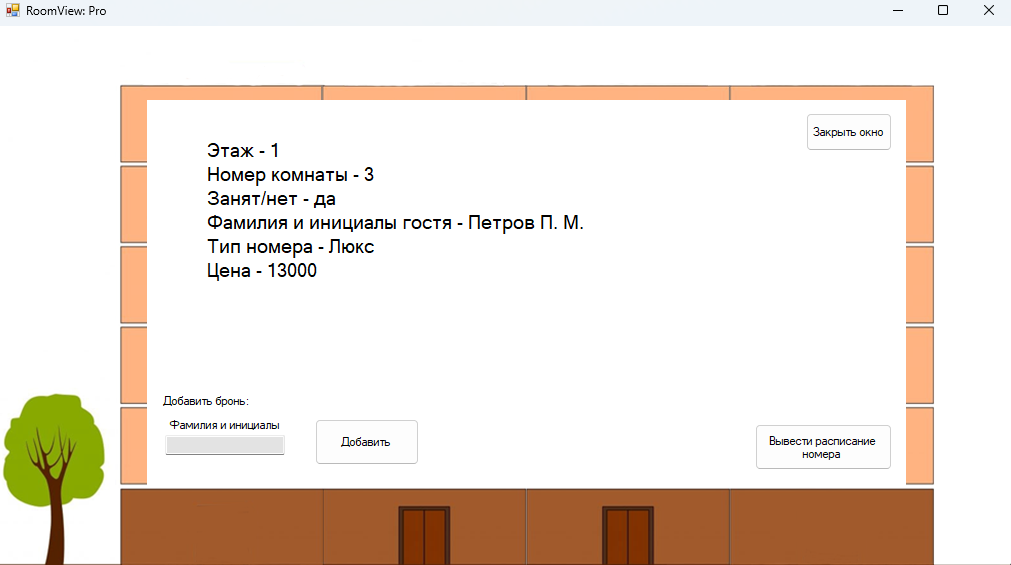


Рисунок 10 – панель с информацией о номере

На рисунках видно, что свободный номер 3 окрасился в красный цвет и информация о госте добавилась в базу данных, следовательно программа работает.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы был спроектирован и создан прикладной программный продукт «Гостиничный бизнес. Фонд недвижимого имущества (здания / сооружения, номера помещения)». Процесс проектирования включал в себя тщательное изучение предметной области, определение функционала и дизайна программы, а также реализацию технической части с последующим тестированием. В ходе тестирования выявленные ошибки были успешно устранены, обеспечивая полноценную работоспособность разработанного продукта.

Также были закреплены навыки самостоятельного использования теоретического материала и получен практический опыт в разработке прикладного программного продукта.

# Список использованной литературы

1. URL:[https://learn.microsoft.com/ru-ru/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp?view=msvc-170 (Дата](https://learn.microsoft.com/ru-ru/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp?view=msvc-170%20%20(Дата) обращения 10.11.23)
2. URL: https://habr.com/ru/articles/572234/ (Дата обращения 20.11.23)
3. URL: https://e.lanbook.com/book/329549 (дата обращения: 26.11.2024).
4. Жвакина А. В. Разработка Windows-приложений в среде визуального программирования: пособие. – 2016.
5. Барков, И. А. Объектно-ориентированное программирование — 2-е изд., стер. — Санкт-Петербург : 2023.

# Приложение А – Класс room

#pragma once

#include "libxl.h"

#include <iostream>

#include <stdio.h>

using namespace libxl;

using namespace System;

using namespace System;

using namespace System::Runtime::InteropServices;

ref class room

{

private:

System::String^ floor;

System::String^ number;

System::String^ busy;

System::String^ fam;

System::String^ type;

System::String^ price;

public:

String^ GetStr() {

String^ tmp = "Этаж - " + floor + "\nНомер комнаты - " + number + "\nЗанят/нет - " + busy

+ "\nФамилия и инициалы гостя - " + fam + "\nТип номера - " + type + "\nЦена - " + price;

return tmp;

}

room(int RowNumber)

{

LoadFromExcel("bd.xls", RowNumber);

}

String^ Search(String^ number) {

if (this->number == number) {

return GetStr();

}

}

void Color(System::Windows::Forms::Button^ tmp, String^ number) {

if (this->number == number) {

if (this->busy == "нет") {

tmp->BackColor = System::Drawing::Color::Blue;

}

else {

tmp->BackColor = System::Drawing::Color::Red;

}

}

}

String^ AddGuest(System::String^ teg, int RowNumber, System::Windows::Forms::Form^)

{

if (this->busy == "нет")

{

Book\* book = xlCreateBook();

book->setKey(L"ERICK RUBEN GONZALEZ BELTRAN", L"windows-202c280c0ac0ec046db46a60a9obmek6");

if (book)

{

if (book->load(L"bd.xls"))

{

Sheet\* sheet = book->getSheet(0);

if (sheet->lastRow() >= 1)

{

String^ guestSurname = teg;

if (guestSurname != "") {

sheet->writeStr(RowNumber, 2, (const wchar\_t\*)Marshal::StringToHGlobalUni("да").ToPointer());

sheet->writeStr(RowNumber, 3, (const wchar\_t\*)Marshal::StringToHGlobalUni(guestSurname).ToPointer());

System::Windows::Forms::MessageBox::Show("Номер забронирован!", "Сообщение", System::Windows::Forms::MessageBoxButtons::OK);

}

else {

System::Windows::Forms::MessageBox::Show("СВОСВОСВОСВОСВОСВО", "Ошибка!", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Warning);

}

book->save(L"bd.xls");

book->release();

return "Гость успешно добавлен!";

}

}

}

}

else {

System::Windows::Forms::MessageBox::Show("Номер уже занят!", "Ошибка!", System::Windows::Forms::MessageBoxButtons::OK);

}

}

private:

void LoadFromExcel(System::String^ excelFilePath, int RowNumber)

{

Book\* book = xlCreateBook();

if (book)

{

if (book->load(L"bd.xls"))

{

Sheet\* sheet = book->getSheet(0);

if (sheet->lastRow() >= 1)

{

floor = sheet->readNum(RowNumber, 0).ToString();

number = sheet->readNum(RowNumber, 1).ToString();

busy = gcnew System::String(sheet->readStr(RowNumber, 2));

fam = gcnew System::String(sheet->readStr(RowNumber, 3));

type = gcnew System::String(sheet->readStr(RowNumber, 4));

price = sheet->readNum(RowNumber, 5).ToString();

}

}

book->release();

}

}

};

# Приложение Б – Класс schedule

#pragma once

#include "libxl.h"

using namespace libxl;

using namespace System;

ref class schedule

{

private:

System::String^ komnata;

System::String^ date;

System::String^ famil;

System::String^ date1;

public:

String^ GetInfo() {

String^ tmp = "Дата заселения - " + date + "\nДата выселения - " + date1 + "\nФамилия и инициалы гостя - " + famil + "\n";

return tmp;

}

schedule(int RowNumber)

{

LoadFromExcel("bd.xls", RowNumber);

}

String^ GetRoom(String^ number) {

if (this->komnata == number) {

return GetInfo();

}

}

private:

void LoadFromExcel(System::String^ excelFilePath, int RowNumber)

{

Book\* book = xlCreateBook();

if (book)

{

if (book->load(L"bd.xls"))

{

Sheet\* sheet = book->getSheet(1);

if (sheet->lastRow() >= 1)

{

komnata = sheet->readNum(RowNumber, 0).ToString();

double excelDate = sheet->readNum(RowNumber, 1);

date = ExcelDateToString(excelDate);

famil = gcnew System::String(sheet->readStr(RowNumber, 2));

double excelDate1 = sheet->readNum(RowNumber, 3);

date1 = ExcelDateToString(excelDate1);

}

}

book->release();

}

}

System::String^ ExcelDateToString(double excelDate) {

DateTime dateTime = DateTime(1899, 12, 30).AddDays(excelDate);

return dateTime.ToString(L"dd.MM.yyyy");

}

};

# Приложение В – Класс MyForm

#pragma once

#include "libxl.h"

#include "room.h"

#include "schedule.h"

using namespace libxl;

namespace kerr {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::Collections::Generic;

using namespace std;

public ref class MyForm : public System::Windows::Forms::Form

{

public:

System::String^ Num;

System::String^ Tex;

public:

MyForm(void)

{

Book\* book = xlCreateBook();

book->load(L"bd.xls");

Sheet\* sheet = book->getSheet(0);

for (int i = 0; i < sheet->lastRow(); i++) {

this->komn->Add(gcnew room(i));

}

Book\* book1 = xlCreateBook();

book1->load(L"bd.xls");

Sheet\* sheet1 = book1->getSheet(1);

for (int i = 0; i < sheet1->lastRow(); i++) {

this->rok->Add(gcnew schedule(i));

}

InitializeComponent();

SetBTNColor();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: List<room^>^ komn = gcnew List<room^>();

private: List<schedule^>^ rok = gcnew List<schedule^>();

private: room^ tmp;

protected:

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button12;

private: System::Windows::Forms::Button^ button13;

private: System::Windows::Forms::Button^ button14;

private: System::Windows::Forms::Button^ button15;

private: System::Windows::Forms::Button^ button16;

private: System::Windows::Forms::Button^ button17;

private: System::Windows::Forms::Button^ button18;

private: System::Windows::Forms::Button^ button19;

private: System::Windows::Forms::Button^ button20;

private: System::Windows::Forms::Button^ button21;

private: System::Windows::Forms::Button^ button22;

private: System::Windows::Forms::Button^ button23;

private: System::Windows::Forms::Button^ button24;

private: System::Windows::Forms::Button^ button25;

private: System::Windows::Forms::Button^ button26;

private: System::Windows::Forms::Button^ button27;

private: System::Windows::Forms::Button^ button28;

private: System::Windows::Forms::Button^ button29;

private: System::Windows::Forms::Button^ button30;

private: System::Windows::Forms::Button^ button31;

private: System::Windows::Forms::Button^ button32;

private: System::Windows::Forms::Button^ button33;

private: System::Windows::Forms::Button^ button34;

private: System::Windows::Forms::Button^ button35;

private: System::Windows::Forms::Button^ button36;

private: System::Windows::Forms::Button^ button37;

private: System::Windows::Forms::Button^ button38;

private: System::Windows::Forms::Button^ button39;

private: System::Windows::Forms::Button^ button40;

private: System::Windows::Forms::Panel^ panel2;

private: System::Windows::Forms::Button^ button42;

private: System::Windows::Forms::Button^ button43;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Button^ button11;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Button^ button41;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::Button^ button44;

private: System::Windows::Forms::Panel^ panel3;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Button^ button47;

private: System::Windows::Forms::Label^ label4;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

void SetBTNColor(){

int i = 40;

Button^ tmp = gcnew Button();

for each (Control^ var in Controls)

{

if (var->GetType() == tmp->GetType()) {

komn[i--]->Color((Button^)var, i.ToString());

}

}

}

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button12 = (gcnew System::Windows::Forms::Button());

this->button13 = (gcnew System::Windows::Forms::Button());

this->button14 = (gcnew System::Windows::Forms::Button());

this->button15 = (gcnew System::Windows::Forms::Button());

this->button16 = (gcnew System::Windows::Forms::Button());

this->button17 = (gcnew System::Windows::Forms::Button());

this->button18 = (gcnew System::Windows::Forms::Button());

this->button19 = (gcnew System::Windows::Forms::Button());

this->button20 = (gcnew System::Windows::Forms::Button());

this->button21 = (gcnew System::Windows::Forms::Button());

this->button22 = (gcnew System::Windows::Forms::Button());

this->button23 = (gcnew System::Windows::Forms::Button());

this->button24 = (gcnew System::Windows::Forms::Button());

this->button25 = (gcnew System::Windows::Forms::Button());

this->button26 = (gcnew System::Windows::Forms::Button());

this->button27 = (gcnew System::Windows::Forms::Button());

this->button28 = (gcnew System::Windows::Forms::Button());

this->button29 = (gcnew System::Windows::Forms::Button());

this->button30 = (gcnew System::Windows::Forms::Button());

this->button31 = (gcnew System::Windows::Forms::Button());

this->button32 = (gcnew System::Windows::Forms::Button());

this->button34 = (gcnew System::Windows::Forms::Button());

this->button35 = (gcnew System::Windows::Forms::Button());

this->button36 = (gcnew System::Windows::Forms::Button());

this->button37 = (gcnew System::Windows::Forms::Button());

this->button38 = (gcnew System::Windows::Forms::Button());

this->button39 = (gcnew System::Windows::Forms::Button());

this->button40 = (gcnew System::Windows::Forms::Button());

this->panel2 = (gcnew System::Windows::Forms::Panel());

this->panel3 = (gcnew System::Windows::Forms::Panel());

this->label5 = (gcnew System::Windows::Forms::Label());

this->button47 = (gcnew System::Windows::Forms::Button());

this->label4 = (gcnew System::Windows::Forms::Label());

this->button44 = (gcnew System::Windows::Forms::Button());

this->label2 = (gcnew System::Windows::Forms::Label());

this->button43 = (gcnew System::Windows::Forms::Button());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->button42 = (gcnew System::Windows::Forms::Button());

this->button11 = (gcnew System::Windows::Forms::Button());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->button41 = (gcnew System::Windows::Forms::Button());

this->label3 = (gcnew System::Windows::Forms::Label());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->button33 = (gcnew System::Windows::Forms::Button());

this->panel2->SuspendLayout();

this->panel3->SuspendLayout();

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(166, 393);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(35, 51);

this->button1->TabIndex = 0;

this->button1->Text = L"1";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button2

//

this->button2->Location = System::Drawing::Point(250, 392);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(35, 51);

this->button2->TabIndex = 1;

this->button2->Text = L"2";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(368, 393);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(35, 51);

this->button3->TabIndex = 2;

this->button3->Text = L"3";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(452, 393);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(35, 51);

this->button4->TabIndex = 3;

this->button4->Text = L"4";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button5

//

this->button5->Location = System::Drawing::Point(572, 393);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(35, 51);

this->button5->TabIndex = 4;

this->button5->Text = L"5";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button6

//

this->button6->Location = System::Drawing::Point(655, 393);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(35, 51);

this->button6->TabIndex = 5;

this->button6->Text = L"6";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button7

//

this->button7->Location = System::Drawing::Point(775, 393);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(35, 51);

this->button7->TabIndex = 6;

this->button7->Text = L"7";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button8

//

this->button8->Location = System::Drawing::Point(859, 393);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(35, 51);

this->button8->TabIndex = 7;

this->button8->Text = L"8";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button9

//

this->button9->Location = System::Drawing::Point(166, 314);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(35, 51);

this->button9->TabIndex = 8;

this->button9->Text = L"9";

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button10

//

this->button10->Location = System::Drawing::Point(250, 314);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(35, 51);

this->button10->TabIndex = 9;

this->button10->Text = L"10";

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button12

//

this->button12->Location = System::Drawing::Point(452, 314);

this->button12->Name = L"button12";

this->button12->Size = System::Drawing::Size(35, 51);

this->button12->TabIndex = 11;

this->button12->Text = L"12";

this->button12->UseVisualStyleBackColor = true;

this->button12->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button13

//

this->button13->Location = System::Drawing::Point(572, 314);

this->button13->Name = L"button13";

this->button13->Size = System::Drawing::Size(35, 51);

this->button13->TabIndex = 12;

this->button13->Text = L"13";

this->button13->UseVisualStyleBackColor = true;

this->button13->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button14

//

this->button14->Location = System::Drawing::Point(655, 314);

this->button14->Name = L"button14";

this->button14->Size = System::Drawing::Size(35, 51);

this->button14->TabIndex = 13;

this->button14->Text = L"14";

this->button14->UseVisualStyleBackColor = true;

this->button14->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button15

//

this->button15->Location = System::Drawing::Point(775, 314);

this->button15->Name = L"button15";

this->button15->Size = System::Drawing::Size(35, 51);

this->button15->TabIndex = 14;

this->button15->Text = L"31";

this->button15->UseVisualStyleBackColor = true;

this->button15->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button16

//

this->button16->Location = System::Drawing::Point(859, 314);

this->button16->Name = L"button16";

this->button16->Size = System::Drawing::Size(35, 51);

this->button16->TabIndex = 15;

this->button16->Text = L"16";

this->button16->UseVisualStyleBackColor = true;

this->button16->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button17

//

this->button17->Location = System::Drawing::Point(166, 233);

this->button17->Name = L"button17";

this->button17->Size = System::Drawing::Size(35, 51);

this->button17->TabIndex = 16;

this->button17->Text = L"17";

this->button17->UseVisualStyleBackColor = true;

this->button17->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button18

//

this->button18->Location = System::Drawing::Point(250, 233);

this->button18->Name = L"button18";

this->button18->Size = System::Drawing::Size(35, 51);

this->button18->TabIndex = 17;

this->button18->Text = L"18";

this->button18->UseVisualStyleBackColor = true;

this->button18->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button19

//

this->button19->Location = System::Drawing::Point(368, 233);

this->button19->Name = L"button19";

this->button19->Size = System::Drawing::Size(35, 51);

this->button19->TabIndex = 18;

this->button19->Text = L"19";

this->button19->UseVisualStyleBackColor = true;

this->button19->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button20

//

this->button20->Location = System::Drawing::Point(452, 233);

this->button20->Name = L"button20";

this->button20->Size = System::Drawing::Size(35, 51);

this->button20->TabIndex = 19;

this->button20->Text = L"20";

this->button20->UseVisualStyleBackColor = true;

this->button20->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button21

//

this->button21->Location = System::Drawing::Point(572, 233);

this->button21->Name = L"button21";

this->button21->Size = System::Drawing::Size(35, 51);

this->button21->TabIndex = 20;

this->button21->Text = L"21";

this->button21->UseVisualStyleBackColor = true;

this->button21->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button22

//

this->button22->Location = System::Drawing::Point(655, 233);

this->button22->Name = L"button22";

this->button22->Size = System::Drawing::Size(35, 51);

this->button22->TabIndex = 21;

this->button22->Text = L"22";

this->button22->UseVisualStyleBackColor = true;

this->button22->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button23

//

this->button23->Location = System::Drawing::Point(775, 233);

this->button23->Name = L"button23";

this->button23->Size = System::Drawing::Size(35, 51);

this->button23->TabIndex = 22;

this->button23->Text = L"23";

this->button23->UseVisualStyleBackColor = true;

this->button23->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button24

//

this->button24->Location = System::Drawing::Point(859, 233);

this->button24->Name = L"button24";

this->button24->Size = System::Drawing::Size(35, 51);

this->button24->TabIndex = 23;

this->button24->Text = L"24";

this->button24->UseVisualStyleBackColor = true;

this->button24->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button25

//

this->button25->Location = System::Drawing::Point(166, 154);

this->button25->Name = L"button25";

this->button25->Size = System::Drawing::Size(35, 51);

this->button25->TabIndex = 24;

this->button25->Text = L"25";

this->button25->UseVisualStyleBackColor = true;

this->button25->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button26

//

this->button26->Location = System::Drawing::Point(250, 154);

this->button26->Name = L"button26";

this->button26->Size = System::Drawing::Size(35, 51);

this->button26->TabIndex = 25;

this->button26->Text = L"26";

this->button26->UseVisualStyleBackColor = true;

this->button26->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button27

//

this->button27->Location = System::Drawing::Point(368, 153);

this->button27->Name = L"button27";

this->button27->Size = System::Drawing::Size(35, 51);

this->button27->TabIndex = 26;

this->button27->Text = L"27";

this->button27->UseVisualStyleBackColor = true;

this->button27->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button28

//

this->button28->Location = System::Drawing::Point(452, 153);

this->button28->Name = L"button28";

this->button28->Size = System::Drawing::Size(35, 51);

this->button28->TabIndex = 27;

this->button28->Text = L"28";

this->button28->UseVisualStyleBackColor = true;

this->button28->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button29

//

this->button29->Location = System::Drawing::Point(572, 154);

this->button29->Name = L"button29";

this->button29->Size = System::Drawing::Size(35, 51);

this->button29->TabIndex = 28;

this->button29->Text = L"29";

this->button29->UseVisualStyleBackColor = true;

this->button29->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button30

//

this->button30->Location = System::Drawing::Point(655, 154);

this->button30->Name = L"button30";

this->button30->Size = System::Drawing::Size(35, 51);

this->button30->TabIndex = 29;

this->button30->Text = L"30";

this->button30->UseVisualStyleBackColor = true;

this->button30->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button31

//

this->button31->Location = System::Drawing::Point(775, 154);

this->button31->Name = L"button31";

this->button31->Size = System::Drawing::Size(35, 51);

this->button31->TabIndex = 30;

this->button31->Text = L"31";

this->button31->UseVisualStyleBackColor = true;

this->button31->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button32

//

this->button32->Location = System::Drawing::Point(859, 154);

this->button32->Name = L"button32";

this->button32->Size = System::Drawing::Size(35, 51);

this->button32->TabIndex = 31;

this->button32->Text = L"32";

this->button32->UseVisualStyleBackColor = true;

this->button32->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button34

//

this->button34->Location = System::Drawing::Point(250, 74);

this->button34->Name = L"button34";

this->button34->Size = System::Drawing::Size(35, 51);

this->button34->TabIndex = 33;

this->button34->Text = L"34";

this->button34->UseVisualStyleBackColor = true;

this->button34->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button35

//

this->button35->Location = System::Drawing::Point(368, 74);

this->button35->Name = L"button35";

this->button35->Size = System::Drawing::Size(35, 51);

this->button35->TabIndex = 34;

this->button35->Text = L"35";

this->button35->UseVisualStyleBackColor = true;

this->button35->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button36

//

this->button36->Location = System::Drawing::Point(452, 74);

this->button36->Name = L"button36";

this->button36->Size = System::Drawing::Size(35, 51);

this->button36->TabIndex = 35;

this->button36->Text = L"36";

this->button36->UseVisualStyleBackColor = true;

this->button36->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button37

//

this->button37->Location = System::Drawing::Point(572, 74);

this->button37->Name = L"button37";

this->button37->Size = System::Drawing::Size(35, 51);

this->button37->TabIndex = 36;

this->button37->Text = L"37";

this->button37->UseVisualStyleBackColor = true;

this->button37->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button38

//

this->button38->Location = System::Drawing::Point(655, 74);

this->button38->Name = L"button38";

this->button38->Size = System::Drawing::Size(35, 51);

this->button38->TabIndex = 37;

this->button38->Text = L"38";

this->button38->UseVisualStyleBackColor = true;

this->button38->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button39

//

this->button39->Location = System::Drawing::Point(775, 74);

this->button39->Name = L"button39";

this->button39->Size = System::Drawing::Size(35, 51);

this->button39->TabIndex = 38;

this->button39->Text = L"39";

this->button39->UseVisualStyleBackColor = true;

this->button39->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// button40

//

this->button40->Location = System::Drawing::Point(859, 74);

this->button40->Name = L"button40";

this->button40->Size = System::Drawing::Size(35, 51);

this->button40->TabIndex = 39;

this->button40->Text = L"40";

this->button40->UseVisualStyleBackColor = true;

this->button40->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// panel2

//

this->panel2->BackColor = System::Drawing::SystemColors::ControlLightLight;

this->panel2->Controls->Add(this->panel3);

this->panel2->Controls->Add(this->label4);

this->panel2->Controls->Add(this->button44);

this->panel2->Controls->Add(this->label2);

this->panel2->Controls->Add(this->button43);

this->panel2->Controls->Add(this->textBox2);

this->panel2->Controls->Add(this->label1);

this->panel2->Controls->Add(this->button42);

this->panel2->Location = System::Drawing::Point(150, 74);

this->panel2->Name = L"panel2";

this->panel2->Size = System::Drawing::Size(759, 386);

this->panel2->TabIndex = 41;

this->panel2->Visible = false;

this->panel2->Paint += gcnew System::Windows::Forms::PaintEventHandler(this, &MyForm::panel2\_Paint);

//

// panel3

//

this->panel3->BackColor = System::Drawing::SystemColors::ControlLightLight;

this->panel3->BorderStyle = System::Windows::Forms::BorderStyle::FixedSingle;

this->panel3->Controls->Add(this->label5);

this->panel3->Controls->Add(this->button47);

this->panel3->Location = System::Drawing::Point(284, 57);

this->panel3->Name = L"panel3";

this->panel3->Size = System::Drawing::Size(472, 256);

this->panel3->TabIndex = 42;

this->panel3->Visible = false;

this->panel3->Paint += gcnew System::Windows::Forms::PaintEventHandler(this, &MyForm::panel3\_Paint);

//

// label5

//

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label5->Location = System::Drawing::Point(6, 0);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(60, 24);

this->label5->TabIndex = 1;

this->label5->Text = L"label5";

//

// button47

//

this->button47->Location = System::Drawing::Point(410, 3);

this->button47->Name = L"button47";

this->button47->Size = System::Drawing::Size(57, 38);

this->button47->TabIndex = 0;

this->button47->Text = L"X";

this->button47->UseVisualStyleBackColor = true;

this->button47->Click += gcnew System::EventHandler(this, &MyForm::button47\_Click);

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(19, 318);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(118, 13);

this->label4->TabIndex = 6;

this->label4->Text = L"Фамилия и инициалы";

this->label4->Click += gcnew System::EventHandler(this, &MyForm::label4\_Click);

//

// button44

//

this->button44->Location = System::Drawing::Point(168, 319);

this->button44->Name = L"button44";

this->button44->Size = System::Drawing::Size(104, 46);

this->button44->TabIndex = 5;

this->button44->Text = L"Добавить";

this->button44->UseVisualStyleBackColor = true;

this->button44->Click += gcnew System::EventHandler(this, &MyForm::button44\_Click);

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(13, 294);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(93, 13);

this->label2->TabIndex = 4;

this->label2->Text = L"Добавить бронь:";

this->label2->Click += gcnew System::EventHandler(this, &MyForm::label2\_Click);

//

// button43

//

this->button43->Location = System::Drawing::Point(608, 324);

this->button43->Name = L"button43";

this->button43->Size = System::Drawing::Size(137, 46);

this->button43->TabIndex = 3;

this->button43->Text = L"Вывести расписание номера";

this->button43->UseVisualStyleBackColor = true;

this->button43->Click += gcnew System::EventHandler(this, &MyForm::button43\_Click);

//

// textBox2

//

this->textBox2->BackColor = System::Drawing::SystemColors::ControlLight;

this->textBox2->Location = System::Drawing::Point(18, 335);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(120, 20);

this->textBox2->TabIndex = 2;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label1->Location = System::Drawing::Point(56, 38);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(60, 24);

this->label1->TabIndex = 1;

this->label1->Text = L"label1";

//

// button42

//

this->button42->Location = System::Drawing::Point(659, 13);

this->button42->Name = L"button42";

this->button42->Size = System::Drawing::Size(86, 38);

this->button42->TabIndex = 0;

this->button42->Text = L"Закрыть окно";

this->button42->UseVisualStyleBackColor = true;

this->button42->Click += gcnew System::EventHandler(this, &MyForm::button42\_Click);

//

// button11

//

this->button11->BackColor = System::Drawing::Color::White;

this->button11->Location = System::Drawing::Point(368, 314);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(35, 51);

this->button11->TabIndex = 10;

this->button11->Text = L"11";

this->button11->UseVisualStyleBackColor = false;

this->button11->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(368, 249);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(301, 20);

this->textBox1->TabIndex = 0;

this->textBox1->UseSystemPasswordChar = true;

//

// button41

//

this->button41->Location = System::Drawing::Point(452, 299);

this->button41->Name = L"button41";

this->button41->Size = System::Drawing::Size(134, 27);

this->button41->TabIndex = 1;

this->button41->Text = L"Вход";

this->button41->UseVisualStyleBackColor = true;

this->button41->Click += gcnew System::EventHandler(this, &MyForm::button41\_Click);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Courier New", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label3->Location = System::Drawing::Point(361, 181);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(325, 39);

this->label3->TabIndex = 42;

this->label3->Text = L"Введите пароль";

//

// panel1

//

this->panel1->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"panel1.BackgroundImage")));

this->panel1->Controls->Add(this->label3);

this->panel1->Controls->Add(this->button41);

this->panel1->Controls->Add(this->textBox1);

this->panel1->Dock = System::Windows::Forms::DockStyle::Fill;

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(1015, 541);

this->panel1->TabIndex = 40;

this->panel1->Paint += gcnew System::Windows::Forms::PaintEventHandler(this, &MyForm::panel1\_Paint);

//

// button33

//

this->button33->Location = System::Drawing::Point(166, 74);

this->button33->Name = L"button33";

this->button33->Size = System::Drawing::Size(35, 51);

this->button33->TabIndex = 42;

this->button33->Text = L"33";

this->button33->UseVisualStyleBackColor = true;

this->button33->Click += gcnew System::EventHandler(this, &MyForm::button\_Click);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->ClientSize = System::Drawing::Size(1015, 541);

this->Controls->Add(this->panel1);

this->Controls->Add(this->panel2);

this->Controls->Add(this->button40);

this->Controls->Add(this->button39);

this->Controls->Add(this->button38);

this->Controls->Add(this->button37);

this->Controls->Add(this->button36);

this->Controls->Add(this->button35);

this->Controls->Add(this->button34);

this->Controls->Add(this->button33);

this->Controls->Add(this->button32);

this->Controls->Add(this->button31);

this->Controls->Add(this->button30);

this->Controls->Add(this->button29);

this->Controls->Add(this->button28);

this->Controls->Add(this->button27);

this->Controls->Add(this->button26);

this->Controls->Add(this->button25);

this->Controls->Add(this->button24);

this->Controls->Add(this->button23);

this->Controls->Add(this->button22);

this->Controls->Add(this->button21);

this->Controls->Add(this->button20);

this->Controls->Add(this->button19);

this->Controls->Add(this->button18);

this->Controls->Add(this->button17);

this->Controls->Add(this->button16);

this->Controls->Add(this->button15);

this->Controls->Add(this->button14);

this->Controls->Add(this->button13);

this->Controls->Add(this->button12);

this->Controls->Add(this->button11);

this->Controls->Add(this->button10);

this->Controls->Add(this->button9);

this->Controls->Add(this->button8);

this->Controls->Add(this->button7);

this->Controls->Add(this->button6);

this->Controls->Add(this->button5);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Location = System::Drawing::Point(166, 393);

this->Name = L"MyForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"RoomView: Pro";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

this->Click += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

this->panel2->ResumeLayout(false);

this->panel2->PerformLayout();

this->panel3->ResumeLayout(false);

this->panel3->PerformLayout();

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button41\_Click(System::Object^ sender, System::EventArgs^ e) {

if (this->textBox1->Text == "1") {

this->panel1->Visible = false;

}

else {

MessageBox::Show("Неправильный пароль!", "Ошибка!", MessageBoxButtons::OK);

}

}

private: System::Void panel1\_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e) {

}

private: System::Void panel2\_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e) {

}

private: System::Void button42\_Click(System::Object^ sender, System::EventArgs^ e) {

this->panel2->Visible = false;

}

private: System::Void button\_Click(System::Object^ sender, System::EventArgs^ e) {

this->panel2->Visible = true;

Button^tmp = (Button^) sender;

Num = tmp->Text;

for each (room^ var in komn)

{

this->label1->Text = var->Search(tmp->Text);

if (label1->Text != "") {

break;

}

}

}

private: System::Void label2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button43\_Click(System::Object^ sender, System::EventArgs^ e) {

this->panel3->Visible = true;

for each (schedule ^ var in rok)

{

Tex = Tex + var->GetRoom(Num);

}

this->label5->Text = Tex;

Tex = " ";

}

private: System::Void button47\_Click(System::Object^ sender, System::EventArgs^ e) {

this->panel3->Visible = false;

}

private: System::Void panel3\_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e) {

}

private: System::Void label4\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label7\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button44\_Click(System::Object^ sender, System::EventArgs^ e) {

room^ roomInstance = gcnew room(System::Convert::ToInt32(Num));

String^ result = roomInstance->AddGuest(this->textBox2->Text, System::Convert::ToInt32(Num), this);

if (result != "") {

Button^ tmp = (Button^)sender;

for each (Control ^ var in Controls)

{

if (var->GetType() == tmp->GetType()) {

if (var->Text == Num) {

var->BackColor = System::Drawing::Color::Red;

}

}

}

}

}

};

}

# Приложение Г – Исполняемый файл MyForm

#pragma once

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

int main(array<String^>^ args) {

Application::SetCompatibleTextRenderingDefault(false);

Application::EnableVisualStyles();

kerr::MyForm form;

Application::Run(% form);

}