Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (МОСКОВСКИЙ ПОЛИТЕХ)

КУРСОВАЯ РАБОТА

По курсу

«Программирование и алгоритмизация на языках высокого уровня»

ТЕМА

**«**Разработка программного обеспечения системы учета заказов ресторана с использованием C# и Windows Forms**»**

Выполнил **\_\_**Сычев Г.А.**\_\_**

*(ФИО)*

Группа **\_\_\_\_**231-328**\_\_\_\_\_**

Проверил **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(ФИО)*

Москва, 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (МОСКОВСКИЙ ПОЛИТЕХ)

УТВЕРЖДАЮ

заведующий кафедрой

/ \_\_\_\_\_\_\_\_\_\_/

« » 20 г.

# ЗАДАНИЕ

**на выполнение курсовой работы (проекта)**

Сычеву Глебу Артемовичу  
(фамилия, имя, отчество обучающегося)

обучающемуся группы 231-328

по дисциплине «Программирование и алгоритмизация на языках высокого уровня»

1. Исходные данные к работе (проекту):
2. Содержание задания по курсовой работе (проекту) – перечень вопросов,

подлежащих разработке:

|  |  |  |  |
| --- | --- | --- | --- |
| Разрабатываемый вопрос | Объем от всего  задания, % | Срок  выполнения | Примечание |
| Раздел 1. Дизайн | 30 | 15.03.2024 |  |
| Задача 1.1. разработка интерфейса | 30 | 18.04.2024 |  |
| Раздел 2. База данных | 20 | 28.04.2024 |  |
| Задача 2.1. разработка базы данных | 20 | 30.04.2024 |  |
| Раздел 3. Разработка | 50 | 28.05.2024 |  |
| Задача 3.1. подключение базы данных | 15 | 29.04.2024 |  |
| Задача 3.2. разработка ПО | 35 | 20.05.2024 |  |

Руководитель курсовой работы (проекта):*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

« » 20 г.

(дата) (подпись) (И.О. Фамилия)

Дата выдачи задания « »\_\_\_\_\_\_\_\_\_20\_\_г. Дата сдачи выполненной работы (проекта) « »\_\_\_\_\_\_\_\_\_20\_\_г. Задание принял к исполнению

« » 20\_\_г.

(дата) (подпись) (И.О. Фамилия)

**Оглавление**

[ВВЕДЕНИЕ 4](#_Toc167966891)

[БАЗА ДАННЫХ 5](#_Toc167966892)

[1. Проектирование базы данных 5](#_Toc167966893)

[1.1 Сущности и их атрибуты 5](#_Toc167966894)

[1.2 Описание сущностей и их использования 5](#_Toc167966895)

[ДИЗАЙН 6](#_Toc167966896)

[2. Описание основных элементов интерфейса 6](#_Toc167966897)

[2.1 Главное окно (Форма Cinema) 6](#_Toc167966898)

[2.2 Форма оплаты (Form2) 7](#_Toc167966899)

[3. Функциональность элементов 7](#_Toc167966900)

[СТРУКТУРА ПРОГРАММЫ 10](#_Toc167966901)

[4. Методы класса Cinema 10](#_Toc167966902)

[4.1 Поля класса 10](#_Toc167966903)

[4.2 Конструктор 10](#_Toc167966904)

[Методы для заполнения комбобоксов 10](#_Toc167966905)

[4.3 Методы для управления кнопками 11](#_Toc167966906)

[4.4 Методы для работы с выбором мест 11](#_Toc167966907)

[4.5 Обработчики событий 11](#_Toc167966908)

[4.6 Вспомогательные методы 12](#_Toc167966909)

[5. Класс Form2 12](#_Toc167966910)

[5.1 Методы класса Form2 12](#_Toc167966911)

[ЗАКЛЮЧЕНИЕ 14](#_Toc167966912)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc167966913)

[ПРИЛОЖЕНИЯ 16](#_Toc167966914)

[Приложение 1 16](#_Toc167966915)

[Приложение 2 22](#_Toc167966916)

[Приложение 3 26](#_Toc167966917)

[Приложение 4 33](#_Toc167966918)

**ВВЕДЕНИЕ**

В современном мире автоматизация производства становится все более востребованной тенденцией, охватывающей не только традиционные заводы и фабрики, но и такие отрасли, как ресторанный бизнес. Внедрение автоматизированных систем позволяет предприятиям существенно сократить издержки за счет оптимизации рутинных процессов.

Целью данной курсовой работы является разработка ПО системы учета заказов в ресторане с использованием C# и Windows Form. Данный выбор обусловлен тем, что C# и Windows Form предоставляют обширные возможности для создания приложений с графическим интерфейсом, что позволит создать интуитивно понятную и функциональную систему учета заказов для ресторана.

В ходе разработки программного обеспечения будут проанализированы требования, разработаны дизайн и архитектура, реализованы функциональные возможности, и проведено тестирование. Важное внимание будет уделено созданию интуитивно понятного интерфейса, интеграции с базой данных для хранения информации о заказах и работниках ресторана.

Эта курсовая работа направлена на получение практических навыков в сфере разработки программного обеспечения, а также создание системы для учета и автоматизации заказов в ресторане.

**БАЗА ДАННЫХ**

1. **Проектирование базы данных**

В системе бронирования билетов в кинотеатре используются две таблицы: **Sittings** и **Tickets**. Эти таблицы обеспечивают хранение информации о сеансах и забронированных местах [1].

**1.1 Сущности и их атрибуты**

**Таблица Sittings:**

* **sitting** (INT): Уникальный идентификатор сеанса.
* **film\_name** (LONG TEXT): Название фильма.
* **hall** (LONG TEXT): Номер зала.
* **time** (LONG TEXT): Время начала сеанса.

**Таблица Tickets:**

* + - **sitting** (INT): Уникальный идентификатор сеанса.
    - **row** (INT): Номер ряда.
    - **place** (INT): Номер места.

**1.2 Описание сущностей и их использования**

Таблица Sittings: хранит информацию о каждом киносеансе, включая уникальный идентификатор сеанса, название фильма, зал и время показа. Эта информация необходима для отображения расписания сеансов и их характеристика.

Таблица Tickets: хранит информацию о забронированных местах для каждого сеанса, включая идентификатор сеанса, номер ряда и номер места. Эта информация используется для отслеживания забронированных мест и предотвращения двойного бронирования.

**ДИЗАЙН**

В рамках разработки системы бронирования билетов в кинотеатре на языке C# с использованием Windows Forms особое внимание уделено созданию удобного и интуитивно понятного интерфейса. Дизайн системы обеспечивает пользователям легкость навигации и простоту выполнения основных задач, таких как выбор фильма, сеанса, мест и бронирование билетов.

1. **Описание основных элементов интерфейса**

**2.1 Главное окно (Форма Cinema)**

* **GroupBox (groupBox1):** Область для отображения плана посадки. Фон области светло-голубого цвета. Этот элемент включает интерактивное отображение свободных и занятых мест в зале.
* **ComboBox (comboBox1):** Выпадающий список для выбора фильма. Расположен в верхней части формы и содержит названия доступных фильмов.
* **Label (label1):** Метка с текстом "Фильм". Расположена рядом с comboBox1, чтобы обозначить назначение выпадающего списка.
* **ComboBox (comboBox2):** Выпадающий список для выбора сеанса. Расположен рядом с comboBox1 и содержит доступные сеансы выбранного фильма.
* **Label (label2):** Метка с текстом "Сеанс". Расположена рядом с comboBox2, чтобы обозначить назначение выпадающего списка.
* **PictureBox (pictureBox1, pictureBox2, pictureBox3, pictureBox4):** Картинки, обозначающие различные состояния мест в зале: свободно (зелёный цвет), VIP (фиолетовый цвет), выбрано (жёлтый цвет) и занято (красный цвет).
* **Label (label3, label4, label5, label6):** Метки с текстом для пояснения значений картинок PictureBox.
* **Button (button1):** Кнопка "Применить". Используется для подтверждения выбора фильма и сеанса.
* **ListBox (listBox1):** Список выбранных мест. Отображает номера выбранных пользователем мест.
* **Label (label7):** Метка с текстом "Ваше имя". Расположена рядом с текстовым полем для ввода имени.
* **TextBox (textBox1):** Текстовое поле для ввода имени пользователя.
* **Button (button2):** Кнопка "Забронировать". Используется для завершения процесса бронирования.

**2.2 Форма оплаты (Form2)**

* **TextBox (textBoxCardNumber, textBoxExpiry, textBoxCVV, textBoxCardHolder):** Текстовые поля для ввода номера карты, срока действия карты, CVV и имени держателя карты соответственно.
* **Button (buttonPay):** Кнопка "Оплатить". Используется для подтверждения оплаты.
* **Label (labelCardNumber, labelExpiry, labelCVV, labelCardHolder):** Метки с текстом для обозначения назначения соответствующих текстовых полей.
* **Label (labelResult):** Метка для отображения результата оплаты.
* **Label (labelCardType):** Метка для отображения типа карты.
* **PictureBox (pictureBoxQRCode):** Картинка для отображения QR-кода, связанного с платежом.

1. **Функциональность элементов**

* **Выбор фильма и сеанса:** при выборе фильма из comboBox1 автоматически обновляется список доступных сеансов в comboBox2. После выбора сеанса и нажатия кнопки "Применить", в области groupBox1 отображается план посадки зала для выбранного сеанса [2].
* **Отображение плана посадки:** Свободные и занятые места обозначаются различными цветами. Пользователь может выбирать свободные места, которые добавляются в listBox1 [3].
* **Бронирование билетов:** после выбора мест и ввода имени пользователя нажатием кнопки "Забронировать" осуществляется переход к форме оплаты.
* **Оплата:** Пользователь вводит данные карты и подтверждает оплату нажатием кнопки "Оплатить". В результате успешной оплаты отображается результат и генерируется QR-код для подтверждения бронирования.

Дизайн системы бронирования билетов в кинотеатре обеспечивает интуитивное взаимодействие пользователя с приложением. Четко структурированные элементы интерфейса и понятные обозначения позволяют пользователям быстро и легко выполнять необходимые действия, что повышает удобство использования системы (Рисунок 1, Рисунок 2, Приложение 1, Приложение 2) [4].

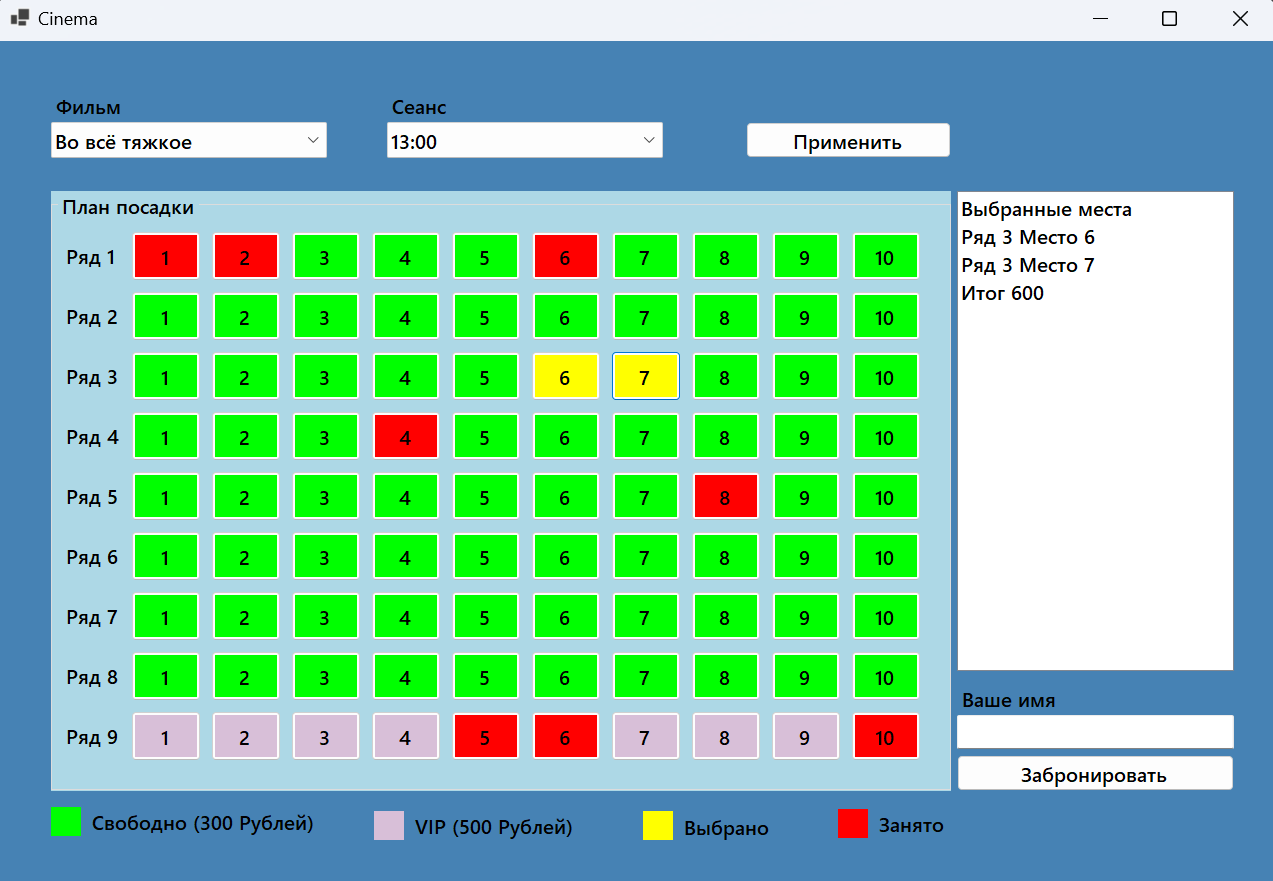
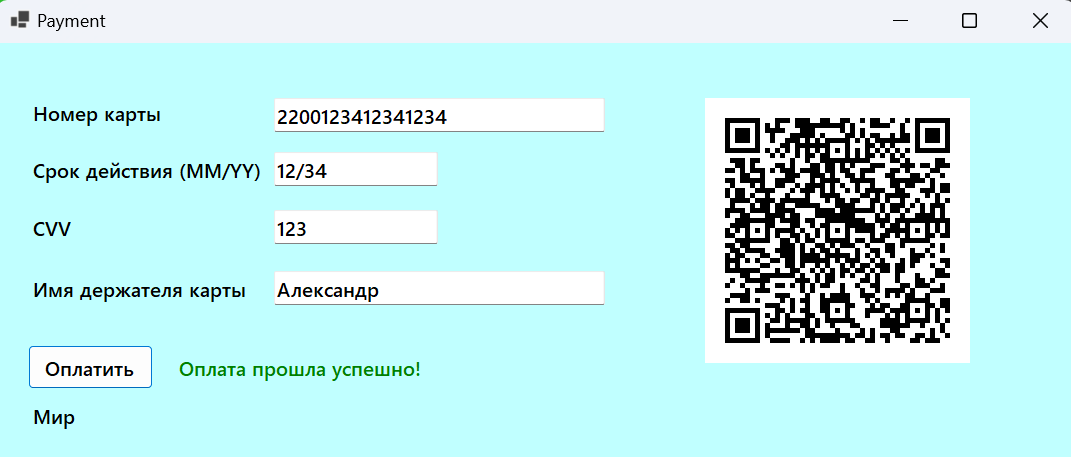
Рисунок 1 – Form1

Рисунок 2 – Form2

**СТРУКТУРА ПРОГРАММЫ**

Проект WinFormsCinema представляет собой приложение на основе Windows Forms для бронирования билетов в кинотеатре. Приложение позволяет пользователям выбирать фильм, сеанс, места в зале и оплачивать билеты. Основной функционал реализован в двух формах: Cinema и Form2.

* **Форма Cinema:** Основная форма для выбора фильма, сеанса и мест, а также для отображения общей стоимости и сохранения данных о бронировании (Приложение 3).
* **Форма Form2:** Форма для ввода данных оплаты и генерации QR-кода после успешной оплаты (Приложение 4).

1. **Методы класса Cinema**

**4.1 Поля класса**

1. **myConn:** Объект для подключения к базе данных Access.
2. **adapter:** Объект для адаптации данных, используется для работы с базой данных.
3. **buttons:** Двумерный массив кнопок, представляющий места в кинозале.
4. **totalCost:** Переменная для хранения общей стоимости выбранных билетов.
5. **myPlaces:** Список выбранных мест, где каждый элемент представляет собой список из двух чисел (ряд и место).
6. **save:** Логическая переменная, используемая для управления состоянием сохранения данных.
   1. **Конструктор**

* **Конструктор Cinema:** инициализирует компоненты формы, создает кнопки для мест в зале и заполняет комбобокс с названиями фильмов. Также задает размер зала (9 рядов и 10 мест в каждом ряду).

**Методы для заполнения комбобоксов**

1. **comboFilm:** открывает соединение с базой данных и заполняет комбобокс (comboBox1) названиями фильмов, выполняя запрос SELECT DISTINCT film\_name FROM Sittings.
2. **comboBox1\_SelectedIndexChanged:** обновляет содержимое комбобокса (comboBox2) временем сеансов на основе выбранного фильма. Выполняет запрос SELECT time FROM Sittings WHERE film\_name = ? с параметром, который передается в запрос.

**4.3 Методы для управления кнопками**

* **CreateButtonsInGroupBox:** создает кнопки, представляющие места в зале, и добавляет их в groupBox1. Устанавливает начальные цвета кнопок и задает обработчики событий для кликов по кнопкам.

**4.4 Методы для работы с выбором мест**

1. **delChoise:** удаляет место из списка myPlaces, если оно было отменено пользователем.
2. **getDatas:** обновляет содержимое listBox1, добавляя туда все выбранные места и общую стоимость.

**4.5 Обработчики событий**

1. **Button\_Click:** обрабатывает клики по кнопкам мест. Если место доступно (не занято), оно добавляется в список myPlaces, а цвет кнопки меняется на желтый. Если место уже выбрано, оно убирается из списка, а цвет кнопки меняется обратно. Обновляет общую стоимость .
2. **ResetButtonColors:** сбрасывает цвета всех кнопок мест в их начальное состояние (доступные места - зеленые, VIP-места - фиолетовые).
3. **buttonSitting:** сбрасывает состояние мест и обновляет их цвета. Очищает список выбранных мест и обнуляет общую стоимость. Затем подключается к базе данных, выполняет запрос для получения ID сеанса, и отмечает занятые места красным цветом.
4. **buttonResult:** обрабатывает завершение бронирования. Проверяет, введено ли имя пользователя и выбраны ли места. Если данные корректны и оплата прошла успешно, сохраняет информацию о бронировании в базе данных и очищает все данные на форме. Также генерирует и сохраняет чек в текстовом файле [5].

**4.6 Вспомогательные методы**

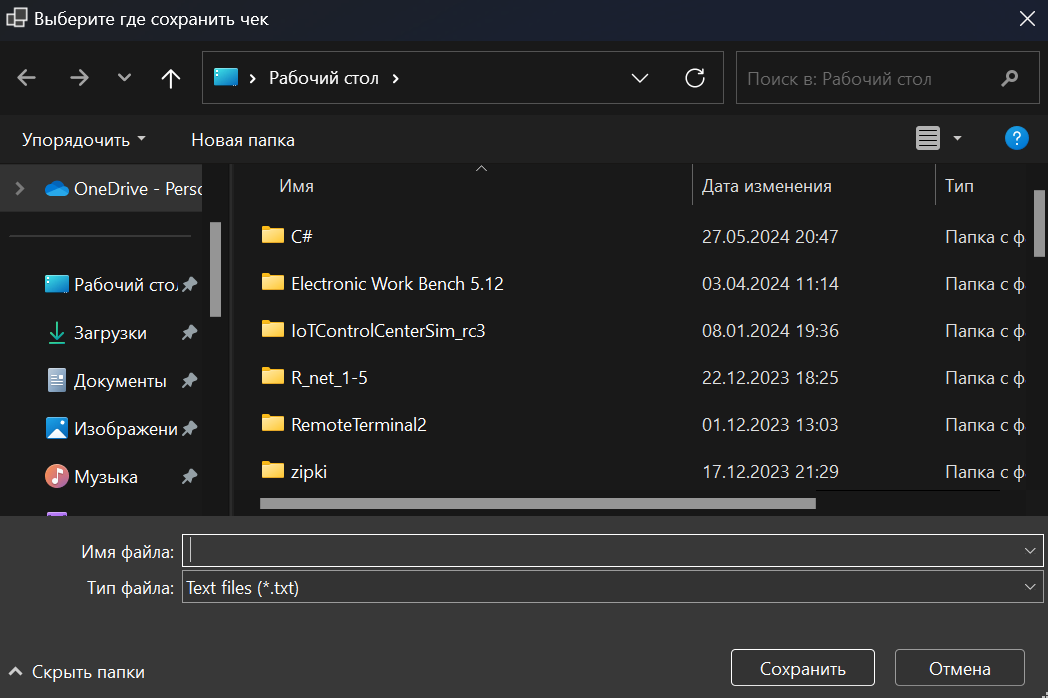
* **SaveListBoxToFile:** сохраняет содержимое listBox1 в текстовый файл. Использует SaveFileDialog для выбора места сохранения файла (Рисунок 3).

Рисунок 3 – сохранение файла

1. **Класс Form2**

**5.1 Методы класса Form2**

1. **buttonPay\_Click:** обрабатывает нажатие кнопки оплаты. Проверяет корректность введенных данных (номер карты, срок действия, CVV и имя держателя карты). Если данные корректны, отображает сообщение об успешной оплате и генерирует QR-код с информацией о карте.
2. **textBoxCardNumber\_TextChanged:** обрабатывает изменение текста в поле номера карты. Определяет тип карты (Visa, MasterCard, American Express, Discover, Мир) и обновляет метку с типом карты.
3. **IsValidCardNumber:** проверяет корректность номера карты (16 цифр).
4. **IsValidExpiry:** проверяет корректность срока действия карты (формат MM/YY).
5. **IsValidCVV:** проверяет корректность CVV-кода (3 цифры).
6. **GetCardType:** определяет тип карты по первым цифрам номера карты.
7. **GenerateQRCode:** генерирует QR-код с информацией о карте (имя держателя, номер карты, срок действия) и отображает его в pictureBoxQRCode [6].

**ЗАКЛЮЧЕНИЕ**

Проект WinFormsCinema представляет собой полноценное приложение для бронирования билетов в кинотеатр, реализованное с использованием Windows Forms. Приложение предоставляет пользователю удобный интерфейс для выбора фильма, сеанса и мест в зале. Благодаря интеграции с базой данных Access, все данные о фильмах, сеансах и билетах хранятся и обрабатываются централизованно.

Ключевые особенности проекта включают:

* **Гибкое управление данными:** Использование ADO.NET и базы данных Access для хранения информации о сеансах и билетах.
* **Интерактивный пользовательский интерфейс:** Визуальное представление мест в зале с возможностью выбора и отмены мест, а также динамическое обновление стоимости.
* **Обработка платежей:** Форма для ввода данных платежной карты с проверкой корректности и генерацией QR-кода после успешной оплаты.
* **Сохранение данных о бронировании:** Возможность сохранения чека в текстовый файл для удобного доступа и хранения данных.

Проект демонстрирует основные принципы разработки приложений на платформе Windows Forms и интеграции с базой данных, а также основы валидации данных и обработки платежей.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Stack Overflow: Работа с базами данных в C#. — URL: https://stackoverflow.com/questions/41402048/how-to-connect-to-a-database-in-c-sharp (дата обращения 10.05.2024).
2. Microsoft Docs: ADO.NET Overview. — URL: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview (дата обращения 12.05.2024).
3. Stack Overflow: Создание и использование Windows Forms. — URL: https://stackoverflow.com/questions/6803073/what-is-the-best-way-to-learn-winforms (дата обращения 15.05.2024).
4. CodeProject: Примеры использования Windows Forms. — URL: https://www.codeproject.com/Articles/1597/Windows-Forms-Programming (дата обращения 15.05.2024).
5. Stack Overflow: Обработка событий в C#. — URL: https://stackoverflow.com/questions/7863835/what-is-the-best-way-to-handle-events-in-c-sharp (дата обращения 20.05.2024).
6. Microsoft Docs: Best Practices for Using Strings in .NET. — URL: https://docs.microsoft.com/en-us/dotnet/standard/base-types/best-practices-strings (дата обращения 22.05.2024).

**ПРИЛОЖЕНИЯ**