

УДК 004.9

# ПРИЛОЖЕНИЕ WINDOWS FORMS ДЛЯ РАСПРОСТРАНЕНИЯ БИЛЕТОВ В ТЕАТРЕ

**КОМРАКОВА ЕВГЕНИЯ ВЛАДИМИРОВНА**

старший преподаватель

**ПИКУЛИНА КСЕНИЯ ИГОРЕВНА**

студент

Гомельский государственный технический университет им. П.О. Сухого

**Аннотация:** проведен аналитический анализ программных средств реализации приложений для распространения билетов, разработана иерархия классов проектируемого приложения. Были рассмотрены и применены различные шаблоны проектирования. Разработано приложение для системы распространения билетов в театре. Найдены критерии удобного приложения. Для реализации приложения были использованы средства языка программирования C# Windows Forms.

**Ключевые слова:** аккаунт, пользователь, регистрация, приложение, театр, модель, «репозиторий», библиотека классов.

## WINDOWS FORMS APPLICATION FOR DISTRIBUTION OF TICKETS IN THE THEATER

**Komrakova Evgenia Vladimirovna,  
Pikulina Ksenia Igorevna**

**Abstract:** An analytical analysis of software tools for implementing applications for the distribution of tickets has been carried out, a hierarchy of classes of the designed application has been developed. Various design patterns have been considered and applied. An application for the theater ticket distribution system has been developed. The criteria for a convenient application are found. To implement the application, the tools of the C# Windows Forms programming language were used.

**Keywords:** account, user, registration, application, theater, model, repository, class library.

Сегодня интернет ресурсы имеют широкие возможности. Не осталась в стороне и сфера развлечения. В современном мире для театра и кино разработано множество приложений. Такие приложения удобны для всех видов пользователей, так как ими можно пользоваться везде и в любое время. Благодаря им, можно вести учет проданных билетов и полученной выручки, вносить изменения в информационную базу приложения при необходимости.

При разработке приложения были выявлены несколько критерий удобного приложения:

- кнопки регистрации, входа и выхода из аккаунта должны быть яркими и заметными и для удобства находиться в левом верхнем углу;
- шрифт информации должен быть не тусклым и читабельным;
- действия покупки, выхода из аккаунта должны подтверждаться выплывающим окном;
- важная информация о пользователе, пароли должны кодироваться;
- обязательные поля для ввода подсвечиваться красным;
- билеты пользователей и чек об оплате должны отправлять на их почту;
- необходима обратная связь с администратором для пользователей, в различных экстренных ситуациях.

При разработке приложения использовалась платформа Microsoft .NET. Так же был применен язык программирования C#, помощью которого создаются надёжные и безопасные приложения. Для интерфейса использовалась подсистема *Windows Forms*. *Windows Forms* – интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде [1, с. 125].

Так же было применено несколько шаблонов проектирования, таких как «Репозиторий» и «Unit of work». Как правило, репозиторий реализует CRUD-интерфейс и привязан к одной конкретной сущности или модели, данными которой он управляет [2, с. 359].

При разработке приложения структура задачи была разбита на более мелкие подзадачи, что дало возможность значительно упростить создание программы, а также ускорить процесс ее разработки. После разбиения – происходило написание этих подзадач, при этом каждая из них проходила тестирование на ошибки, что позволило своевременно устранить их на начальных этапах разработки. После этого подзадачи объединялись, при этом образуя одно цельное решение.

При разбиении задачи была создана общая функциональная схема приложения, в которой была предоставлена последовательность работы ее элементов, от точки входа в программу, то есть запуска, до генерации объектов и от рисовки графического интерфейса. Схема предоставлена на рисунке 1.



Рис. 1. Обобщенная функциональная схема приложения

Архитектура приложения была разделена на 3 фундаментальные части:

- библиотека классов, описывающая сущности театра;
- библиотека классов, описывающая логику и взаимодействие с программой;
- библиотека классов для работы с графическим интерфейсом *Windows Forms*.

Первая библиотека описывает сущности программы: пользователя и его роли, театр, билеты и их категории, спектакль и его жанры, места в театре.

Библиотека, с помощью репозитория, реализует функционирование сущностей и взаимодействия с ними. С помощью коллекции *IEnumerable* были написаны методы для вывода списка сущностей.

Вторая библиотека описывает логику взаимодействия пользователя с сущностями. Например,

добавлять, изменять и удалять спектакли может только администратор. А так же, библиотека содержит классы для авторизации пользователя.

Третья библиотека – это библиотека для взаимодействия с графикой Windows Forms. WFA содержит множество элементов для создания визуально приятного и понятного приложения. А также в этой библиотеке классов содержатся пользовательские элементы управления.

В результате проделанной работы было создано «Windows Forms приложение для системы распространения билетов в театре», где имеется 3 вида пользователей:

- гость;
- зарегистрированный пользователь;
- администратор.

Для реализации приложения были использованы средства языка программирования C# Windows Forms.

Был проведен аналитический анализ программных средств реализации приложений, разработана иерархия классов проектируемого приложения. Также рассмотрены и применены различные шаблоны проектирования, в особенности такие как: Repository, Unit of Work.

В процессе создания приложения был создан интуитивно понятный пользовательский интерфейс для любого вида пользователя.

С помощью библиотеки модульных тестов протестированы участки «Windows Form приложение для системы распространения билетов в театре». Функции протестированных методов:

- корректность выбора определенного места;
- корректность создание новой категории билетов и проверка ее на существование;
- корректность создания мест;
- создание нулевого аргумента и проверка на появление исключения.

Без правильной работы данных методов, приложение будет работать некорректно.

Все классы программы содержат элементы автодокументирования.

#### Список источников

1. Тролсен, Э. Язык программирования C# 7 и платформа .NET и .NET Core. 8-е издание / Э. Тролсен – ООО «Диалектика», 2018 – 1328 с.
2. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# 4-е издание / Д. Рихтер – «Питер», 2021 – 896 с.