

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный
университет информатики и
радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Программирование на языках высокого уровня

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему
«Система бронирования мест в кинотеатре»

Студент

Ф. И. Рыбак

Руководитель

Е. В. Богдан

МИНСК 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)

2023 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Рыбак Феона Игоревна

1. Тема проекта Система бронирования мест в кинотеатре
 2. Срок сдачи студентом законченного проекта 15 декабря 2023 г.
 3. Исходные данные к проекту: картинки png для иконок мест в папке img, cinema.sqlite(база данных)
 4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)
 1. Введение.
 2. Задание.
 3. Обзор литературы.
 - 3.1. Обзор методов и алгоритмов решения поставленной задачи.
 4. Функциональное проектирование.
 - 4.1. Структура входных и выходных данных.
 - 4.2. Разработка диаграммы классов.
 - 4.3. Описание классов.
 5. Разработка программных модулей.
 - 5.1. Разработка схем алгоритмов (два наиболее важных метода).
 - 5.2. Разработка алгоритмов (описание алгоритмов по шагам для двух методов).
 6. Результаты работы.
 7. Заключение
 8. Литература
 9. Приложения
5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)
1. Диаграмма классов.
 2. Схема алгоритма on btnBook_clicked().
 3. Схема алгоритма on btnCancel_clicked().
6. Консультант по проекту (с обозначением разделов проекта) Е.В. Богдан

7. Дата выдачи задания 15 сентября 2023 г.

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

1. Выбор задания. Разработка содержания пояснительной записки. Перечень графического материала – 15 %;

разделы 2, 3 – 10 %;

разделы 4 к – 20 %;

разделы 5 к – 35 %;

раздел 6,7,8 – 5 %;

раздел 9 к – 5%;

оформление пояснительной записки и графического материала к 15.12.23 – 10 %

Защита курсового проекта с 21.12 по 28.12.23г.

РУКОВОДИТЕЛЬ _____ Е. В. Богдан

(подпись)

Задание принял к исполнению _____ Ф.И.Рыбак

(дата и подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ.....	6
2 ОБЗОР ЛИТЕРАТУРЫ.....	7
3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ	9
3.1 Структура входных и выходных данных.....	9
3.2 Разработка диаграммы классов	9
3.3 Описание классов.....	10
4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ	15
4.1 Разработка схем алгоритмов.....	15
4.2 Разработка алгоритмов	15
5 РЕЗУЛЬТАТ РАБОТЫ	17
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24
ПРИЛОЖЕНИЕ А	25
ПРИЛОЖЕНИЕ Б.....	26
ПРИЛОЖЕНИЕ В	27
ПРИЛОЖЕНИЕ Г.....	28

ВВЕДЕНИЕ

Кинотеатры являются популярными местами для проведения досуга и развлечений. Однако, для многих посетителей, особенно в пиковые периоды или на популярные сеансы, может быть сложно получить желаемое место в зале без предварительного бронирования. Для упрощения этого процесса и обеспечения удобства посетителям разработано программное решение на основе языка программирования C++ с использованием фреймворка Qt и базы данных SQLite.

Целью данной курсовой работы является создание системы бронирования мест в кинотеатре. Она предоставляет возможность посетителям выбирать места, бронировать их и осуществлять отмену брони в случае необходимости. Система также обеспечивает удобное отображение доступных мест и информацию о забронированных местах.

Для разработки данного проекта используется язык программирования C++ с использованием фреймворка Qt. Qt обладает мощными инструментами для разработки пользовательских интерфейсов, что позволяет создать интуитивно понятный и привлекательный интерфейс для посетителей кинотеатра. Кроме того, в качестве базы данных выбран SQLite, так как она является легкой и компактной, что особенно важно для проектов с небольшим объемом данных, таких как система бронирования мест.

В процессе разработки системы будут использованы различные классы и объекты, такие как `MainWindow`, `book`, `cancel` и `MyDB`, которые обеспечат функциональность бронирования, отмены брони и управления базой данных соответственно. Классы будут взаимодействовать друг с другом через сигналы и слоты, обеспечивая согласованность данных и корректное выполнение операций.

Ожидается, что разработанная система бронирования мест в кинотеатре значительно упростит процесс бронирования для посетителей и повысит уровень обслуживания. В результате выполнения курсовой работы был получен опыт работы с языком C++, фреймворком Qt и опыт создания системы бронирования мест в кинотеатре.

1. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Овладеть практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием языка высокого уровня C++ ,овладеть практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием языка высокого уровня C++[1]. Разработать систему бронирования мест в кинотеатре на языке программирования C++ с использованием фреймворка Qt и базы данных SQLite. Понимание Основных Принципов ООП:

Цель: Освоить базовые концепции ООП, такие как инкапсуляция, наследование и полиморфизм. Почему это важно: Это обеспечит более глубокое понимание организации кода и его взаимодействия.

Навыки Проектирования Классов и Объектов:

Цель: Уметь создавать классы и объекты, определять их атрибуты и методы. Почему это важно: это является основой ООП и позволяет структурировать код для более легкого понимания и поддержки.

Применение Инкапсуляции:

Цель: Использовать инкапсуляцию для скрытия внутренних деталей реализации классов и предоставления публичного интерфейса. Почему это важно: это способствует безопасности кода и облегчает его сопровождение.

Мастерство В Обработке Наследования: Цель: понимать, как использовать наследование для создания иерархии классов и расширения функциональности. Почему это важно: это позволяет эффективно использовать и пере использовать код.

Овладение Исключениями и Обработкой Ошибок: Цель: Знание, как обрабатывать исключения и ошибки в объектно-ориентированных программах. Почему это важно: это повышает устойчивость программы к ошибкам и улучшает ее отказоустойчивость.

Работа с Дизайн-паттернами:

Цель: Знание и применение распространенных дизайн-паттернов. Почему это важно: Дизайн-паттерны предоставляют более эффективные и проверенные подходы к решению типичных задач в ООП. Эти цели могут служить отправной точкой для разработки программистом плана обучения и практического применения концепций ООП в реальных проектах.

Преимущество Qt:

Простота использования; Qt имеет хорошо документированную структуру, прост в освоении и имеет простую, однородную структуру.

Разработка GUI; Qt предоставляет широкий спектр компонентов для создания графических пользовательских интерфейсов, включая Qt Quick, который позволяет быстро и легко создавать интерфейс с использованием специального языка под названием QML.

Qt — это кроссплатформенный фреймворк для разработки приложений на C++. Он предоставляет разработчикам множество инструментов и библиотек для создания графического интерфейса пользователя, работы с сетью, базами данных, мультимедиа и многого другого. Ниже перечислены некоторые преимущества Qt:

Кроссплатформенность; Qt позволяет создавать приложения, которые могут работать на различных операционных системах, таких как Windows, macOS, Linux, Android и iOS. Это упрощает разработку и позволяет достичь большей аудитории.

Мощный графический интерфейс; Qt предоставляет разработчикам множество инструментов для создания красивых и функциональных пользовательских интерфейсов. Он также поддерживает множество стилей и тем оформления, что позволяет создавать приложения с различным дизайном. В целом, использование Qt в C++ дает много преимуществ для разработчиков, которые хотят создавать кроссплатформенные приложения с графическим пользовательским интерфейсом. Это мощный и гибкий инструмент, который можно использовать в широком спектре приложений, от настольных и мобильных приложений до специализированного оборудования и встроенных систем.

Богатая функциональность; Qt предоставляет множество библиотек и инструментов для работы с сетью, базами данных, мультимедиа, графикой и многим другим. Это позволяет разработчикам создавать приложения с различными функциями и возможностями.

Открытый исходный код; Qt является свободным и открытым фреймворком, что позволяет разработчикам использовать его бесплатно и вносить свои изменения в исходный код.

Широкое сообщество; Qt имеет большое сообщество разработчиков, которые создают и поддерживают множество библиотек и инструментов для Qt. Это позволяет разработчикам быстро находить решения для своих проблем и получать поддержку от других разработчиков.

Реализовать основные функциональности системы:

1. Отображение доступных мест в зале кинотеатра.
2. Возможность выбора и бронирования конкретного места.
3. Проверка валидности вводимых данных при бронировании (имя, фамилия).
4. Отмена бронирования места.
5. Обновление интерфейса после каждой операции бронирования или отмены.
6. Реализовать взаимодействие между классами через сигналы и слоты.

Например, при успешном бронировании места необходимо обновить интерфейс и отобразить соответствующее сообщение об успешном бронировании.

7. Обеспечить корректное сохранение и обновление данных в базе данных SQLite. Например, при бронировании места необходимо обновить статус места в базе данных на "забронировано" и добавить информацию о посетителе.

2. ОБЗОР ЛИТЕРАТУРЫ

Официальная документация: Сайт Qt: Официальный сайт Qt содержит обширную документацию, включая руководства, API—справочники, примеры кода и другие материалы.

Руководства и Введение в Qt:

Getting Started: Раздел "Getting Started" в документации обычно предоставляет информацию о том, как установить Qt, настроить среду разработки и создать простое приложение. Overview: Введение в фреймворк, его основные концепции и принципы.

Создание графического интерфейса: Qt Widgets: Информация о виджетах Qt, базовых элементах управления, таких как кнопки, поля ввода и другие. Qt Quick и QML: Документация о создании интерфейсов с использованием декларативного языка QML и фреймворка Qt Quick.

Работа с сетью и базами данных: Qt Network: Инструменты для работы с сетью, включая HTTP—запросы, сокеты и другие. Qt SQL: Информация о работе с базами данных, включая поддержку различных СУБД.

Многозадачность и Параллелизм: Qt Concurrency: Раздел, посвященный поддержке многозадачности и параллелизма в Qt.

Межплатформенная разработка: Platform Notes: Рекомендации и особенности для кроссплатформенной разработки на разных операционных системах. Deployment: Инструкции по развертыванию Qt—приложений на различных платформах.

Примеры кода и Учебные проекты: Qt Examples: Обширный набор примеров кода для различных компонентов Qt. Qt Tutorials: Учебные проекты и tutorиалы, позволяющие освоить различные аспекты фреймворка.

Обновления и Дополнительные ресурсы: Блог Qt: Официальный блог с новостями, статьями и обновлениями от команды разработчиков. Дополнительные ресурсы: Дополнительные материалы, такие как видеоуроки, вебинары и другие образовательные ресурсы. Qt предоставляет обширные средства для создания высококачественных приложений, и его документация является важным ресурсом для разработчиков, стремящихся освоить этот фреймворк.

2.1 Обзор методов и алгоритмов решения поставленной задачи

Создание пользовательского интерфейса. Для реализации пользовательского интерфейса системы бронирования мест в кинотеатре используется фреймворк Qt. Qt предоставляет богатый набор инструментов для создания графического интерфейса, включая виджеты, компоновщики, стили и темы.

Можно создать окно приложения с кнопками, текстовыми полями и другими элементами управления для взаимодействия с пользователями.

Управление базой данных. Для хранения информации о доступных местах, забронированных местах и других данных необходима база данных. В данном случае выбрана база данных SQLite, которая является легкой и простой в использовании. Для работы с базой данных SQLite используется класс MyDB, который обеспечивает подключение к базе данных, выполнение SQL-запросов и обработку результатов.

Отображение доступных мест. Для отображения доступных мест в зале кинотеатра можно использовать графический виджет, такой как таблица или сетка, где каждая ячейка представляет отдельное место. Места могут быть представлены в виде кнопок или других элементов, которые можно выбирать для бронирования.

2.2 Объектно-ориентированное программирование

Объектно-ориентированное программирование — это методология программирования, которая использует объекты для моделирования и организации кода. В C++ ООП реализуется с помощью классов и объектов [7]. Классы в C++ являются шаблонами для создания объектов. Они определяют, какие данные и функции будут содержать объекты этого класса.

Объектно-ориентированное программирование (ООП) - это парадигма программирования, которая организует программу вокруг объектов, представляющих сущности их свойствами и поведением. В ООП основное внимание уделяется моделированию реального мира, где объекты взаимодействуют друг с другом, передают сообщения и сотрудничают для выполнения задач. ООП в C++ поддерживает четыре основных принципа:

1. Инкапсуляция: Инкапсуляция означает сокрытие деталей реализации и объединение данных и методов в одном объекте;
2. Наследование: Наследование позволяет создавать новые классы на основе существующих, переиспользуя их код и добавляя новые функции;
3. Полиморфизм: Полиморфизм позволяет использовать один и тот же интерфейс для различных типов данных;
4. Абстракция: Абстракция означает представление сложных систем через более простые интерфейсы.

Итак, изучение объектно-ориентированного программирования (ООП) выдало себя как преобразующий этап в моем понимании и подходе к разработке программного обеспечения. Погружение в мир ООП расширило мои знания и компетенции, обогатив мой инструментарий в области программирования.

Одним из ключевых выводов, которые я сделал в процессе написания курсовой работы, является то, что ООП не только предоставляет эффективные средства организации кода, но и обеспечивает более высокий уровень абстракции, что делает разработку программ более интуитивной и гибкой. Использование классов и объектов позволяет создавать модульные, легко читаемые и поддерживаемые системы.

В завершение, несомненно, что изучение ООП открывает перед разработчиком множество возможностей для создания инновационных и эффективных решений. ООП становится неотъемлемым инструментом для современного разработчика, обеспечивая не только технические навыки, но и фундаментальное понимание архитектурных принципов, необходимых для успешного участия в динамичном мире разработки программного обеспечения.

2.3 Базы данных SQLite

Приложение также использует SQLite для работы с базой данных SQLite - это встроенная реляционная база данных, которая предоставляет легкую и эффективную систему управления базами данных. Она является простой в использовании и не требует настройки сервера или установки дополнительного программного обеспечения. Вот некоторые ключевые характеристики базы данных SQLite:

1. **Легковесность:** SQLite является компактной и легкой базой данных, которая может быть встроена непосредственно в приложение. База данных SQLite представляет собой один файл, который хранится на диске, и не требует запуска отдельного сервера для работы с ней;
2. **Кросс-платформенность:** SQLite поддерживается на различных платформах, включая Windows, macOS, Linux и другие операционные системы. Это обеспечивает универсальность и переносимость приложений, использующих SQLite;
3. **Транзакционность:** SQLite обеспечивает транзакционную обработку данных, что позволяет выполнить набор операций как единое целое. Это обеспечивает целостность данных и возможность отката транзакций в случае ошибок или сбоев.
4. **Поддержка многопользовательского доступа:** SQLite поддерживает одновременный доступ нескольких пользователей к базе данных. Однако он не поддерживает параллельную запись несколькими пользователями. Параллельная запись возможна только при условии, что каждый пользователь работает с отдельной таблицей или использует различные транзакции;
5. **Надежность и стабильность:** SQLite является надежной базой данных с хорошо проверенной стабильностью. Он обеспечивает целостность данных и резервное копирование базы данных для защиты от потери данных;
6. **Поддержка расширений:** SQLite предоставляет возможность использовать пользовательские функции и расширения, написанные на языках программирования C/C++ или других языках, для расширения возможностей базы данных.

SQLite является популярным выбором для различных типов приложений, включая мобильные приложения, настольные приложения, веб-сайты и другие проекты, где требуется легковесная и простая в использовании база данных.

2.4 Фреймворк QT

Qt - это кросс-платформенный фреймворк разработки приложений, который предоставляет инструменты и библиотеки для создания графического интерфейса, обработки событий, работы с сетью, базами данных и другими функциональными возможностями, это мощный инструмент для создания GUI в C++ [4]. Qt SQL. Qt – полностью объектно-ориентированная библиотека. Новая концепция ведения меж объектных коммуникаций, именуемая «сигналы и слоты», полностью заменяет былую, не вполне надежную модель обратных вызовов [6]. Графический интерфейс: Qt предоставляет мощные инструменты для создания графического интерфейса пользователя (GUI). Он включает в себя широкий набор виджетов, таких как кнопки, текстовые поля, таблицы, списки и другие элементы управления, а также возможности для настройки внешнего вида и стиля интерфейса. Также имеется возможность обработки событий, например нажатия клавиш клавиатуры, нажатия элементов интерфейса [5].

2.5 Разработка требований к функционалу

“Система бронирования мест в кинотеатре” - представляет собой программное приложение, разработанное с использованием фреймворка Qt и языка программирования C++.

В проекте есть несколько классов, каждый из которых представляет определенное окно или диалоговое окно в приложении.

Бронирование места. При выборе места пользователем и нажатии кнопки "Забронировать" необходимо выполнить следующие шаги:

1. Проверить введенные данные (например, имя и фамилию) на валидность;

2. Проверить доступность выбранного места (например, проверить его статус в базе данных). Если место доступно, обновить информацию в базе данных (изменить статус места на "забронировано" и сохранить информацию о посетителе);

3. Обновить интерфейс, отобразить сообщение об успешном бронировании.

Отмена бронирования. При выборе забронированного места и нажатии кнопки "Отменить" необходимо выполнить следующие шаги:

Проверить доступность отмены бронирования (например, проверить статус места и наличие информации о посетителе);

Если отмена возможна, обновить информацию в базе данных (изменить статус места на "доступно" и удалить информацию о посетителе);

4. Обновить интерфейс, отобразить сообщение об успешной отмене бронирования.

Обработка сигналов и слотов. Для обеспечения взаимодействия между различными классами и объектами системы используются сигналы и слоты. Например, при успешном бронировании или отмене бронирования класс MainWindow может отправлять сигналы, которые приводят к обновлению интерфейса в классах book и cancel.

Так же при разработке была использована официальная документация Qt project. Qt — это мощный фреймворк для разработки кроссплатформенных приложений на C++. Он предоставляет широкий набор инструментов и библиотек для создания графических интерфейсов, обработки событий, работы с сетью, базами данных и другими задачами.

2.6 Обзор аналогов

2.6.1 BookMyShow

BookMyShow[8] - является одним из самых широко используемых онлайн сервисов для бронирования мест в кинотеатрах. Он предоставляет возможность пользователям выбирать фильмы, кинотеатры и сеансы, а также бронировать места и приобретать билеты онлайн. BookMyShow также предлагает дополнительные функции, такие как обзоры фильмов, рейтинги, расписание показов и многое другое.

Преимущества: Широкий выбор кинотеатров и сеансов: BookMyShow предлагает обширную сеть кинотеатров и разнообразные сеансы, что позволяет пользователям выбирать наиболее удобное время и место для просмотра фильма.

Дополнительные функции: Помимо бронирования мест, BookMyShow предлагает полезные функции, такие как обзоры фильмов, рейтинги, расписание показов и информацию о новинках киноиндустрии.

2.6.2 Cinepolis

Cinepolis[9] - это одна из крупнейших сетей кинотеатров в мире. Они предлагают свою собственную систему бронирования мест, которая позволяет пользователям выбирать фильмы, кинотеатры и сеансы, а также бронировать места и приобретать билеты онлайн. Cinepolis также предлагает различные дополнительные услуги, включая программы лояльности и различные акции для постоянных клиентов.

Преимущества: Качество кинотеатров: Cinepolis славится своими современными кинотеатрами, которые обладают высоким уровнем комфорта и качественным звуком и изображением. Это делает просмотр фильмов более приятным для зрителей. Программы лояльности: Cinepolis предлагает

программы лояльности и специальные акции для постоянных клиентов, что позволяет им получать дополнительные преимущества и скидки при бронировании мест.

2.6.3 Fandango

Fandango[10] - это еще одна популярная платформа для бронирования мест

в кинотеатрах. Она предоставляет пользователю широкий выбор фильмов, кинотеатров и сеансов, а также позволяет бронировать места и приобретать билеты онлайн. Fandango также предлагает дополнительные функции, такие как трейлеры фильмов, обзоры, рейтинги и возможность оставлять отзывы.

Преимущества: Обширная информация о фильмах: Fandango предлагает подробные обзоры фильмов, рейтинги, трейлеры и другую полезную информацию, которая помогает пользователям принять информированное решение о выборе фильма. Возможность оставить отзывы: Пользователи могут оставлять отзывы о фильмах на платформе Fandango, что позволяет другим зрителям получить представление о качестве фильма и выбрать наиболее подходящий для себя вариант.

2.6.4 Atom Tickets

Atom Tickets[11] - это известная платформа для бронирования мест в кинотеатрах. Она предлагает широкий выбор фильмов, кинотеатров и сеансов, а также позволяет пользователям бронировать места и приобретать билеты онлайн. Atom Tickets также предоставляет дополнительные функции, такие как обзоры фильмов, рейтинги и возможность покупать еду и напитки заранее.

Преимущества: Заказ еды и напитков: Atom Tickets позволяет пользователям заказывать еду и напитки заранее, чтобы избежать очередей и сделать просмотр фильма еще более удобным и комфортным. Интеграция с другими сервисами: Atom Tickets предлагает интеграцию с другими сервисами, такими как сервисы заказа такси или поиска ресторанов, что позволяет удобно планировать свой кинематографический опыт.

Каждый из упомянутых шифраторов файлов имеет свои преимущества и недостатки, и выбор зависит от конкретных потребностей пользователя, операционной системы и предпочтений в использовании.

3.ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В данном разделе описываются входные и выходные данные программы, диаграмма классов, а также приводится описание используемых классов и их методов.

3.1 Входные данные и выходные

Проект на тему "Система бронирования мест в кинотеатре" представляет собой приложение, которое обеспечивает удобное и автоматизированное бронирование мест для посетителей кинотеатра.

Структура входных и выходных данных моего данного проекта следующая:

Входные данные. Данные пользователя: Имя и фамилия пользователя для регистрации или идентификации, выбранные места в зале.

Выходные данные. Подтверждение бронирования: Успешное завершение бронирования мест: сообщение о успешном бронировании мест, сообщение об ошибке при неудачной попытке бронирования: возможные причины неудачи (недостаточное количество свободных мест, некорректные данные пользователя и т.д.), информация о свободных и занятых местах в зале после бронирования.

3.2 Разработка диаграммы классов

Диаграмма классов - это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними. Она широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

На диаграмме классы представлены в рамках, содержащих три компонента:

1. В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом;
2. В средней части перечислены атрибуты (поля) класса;
3. В нижней части перечислены методы класса.

Диаграмма классов служит для визуализации статического представления системы, представляя различные аспекты приложения. Она представляет собой графическое представление статического представления системы и представляет различные аспекты приложения и также может включать в себя классы, их атрибуты, методы и отношения между классами, такие как наследование, агрегация, ассоциация, множественность ассоциации, обобщение, зависимость, использование, реализация и композиция.

Диаграмма классов для приложения «Система бронирования мест в кинотеатре» приведена в приложении А.

3.3 Описание классов

Для создания программы системы бронирования в C++ с интерфейсом в Qt, вы можете создать следующую структуру классов:

3.3.1 Класс book:

Класс book представляет диалоговое окно в приложении.

Описание полей класса:

`Ui::book *ui` — Указатель на объект пользовательского интерфейса, с помощью которого можно взаимодействовать с элементами интерфейса данного диалогового окна.

`QColor validColor` — Цвет, используемый для обозначения допустимых значений или состояний.

`QColor invalidColor` — Цвет, используемый для обозначения недопустимых значений или состояний.

`void mysignal()` — Сигнал, который может быть отправлен для уведомления других объектов о наступлении определенного события в классе book.

`void on_btnBook_clicked()` — Слот, вызываемый при нажатии на кнопку с именем "btnBook" в диалоговом окне book.

`void on_lineEdit_textEdited(const QString &arg1)` — Слот, вызываемый при редактировании текста в объекте lineEdit в диалоговом окне book.

`void on_lineEdit_2_textEdited(const QString &arg1)` — Слот, вызываемый при редактировании текста в объекте lineEdit_2 в диалоговом окне book.

`explicit book(QWidget *parent = nullptr)` — Конструктор класса book, который принимает указатель на родительский виджет (окно) в качестве аргумента. Если аргумент не передан, то родительский виджет считается равным nullptr.

`~book()` — Деструктор класса book, который освобождает ресурсы, занятые объектом класса book.

Методы:

`void updateUI()` — Метод класса book для обновления пользовательского интерфейса.

3.3.2 Класс cancel:

Класс cancel представляет диалоговое окно в приложении.

Описание полей класса:

`Ui::cancel *ui` — Указатель на объект пользовательского интерфейса, с помощью которого можно взаимодействовать с элементами интерфейса данного диалогового окна.

`void mysignal1()` — Сигнал, который может быть отправлен для уведомления других объектов о наступлении определенного события в классе `cancel`.

`void on_btnCancel_clicked()` — Слот, вызываемый при нажатии на кнопку с именем "btnCancel" в диалоговом окне `cancel`.

`explicit cancel(QWidget *parent = nullptr)` — Конструктор класса `cancel`, который принимает указатель на родительский виджет (окно) в качестве аргумента. Если аргумент не передан, то родительский виджет считается равным `nullptr`.

`~cancel()` — Деструктор класса `cancel`, который освобождает ресурсы, занятые объектом класса `cancel`.

`void updateUI()` — Метод класса `cancel` для обновления пользовательского интерфейса.

3.3.3 Класс MyDB:

Класс `MyDB` представляет собой класс, отвечающий за управление базой данных.

Описание полей класса:

`QSqlDatabase db` — Объект класса `QSqlDatabase`, представляющий подключение к базе данных. Он используется для выполнения операций с базой данных.

`MyDB* instance` — Указатель на объект класса `MyDB`, используемый для реализации синглтона. Этот указатель позволяет иметь только один экземпляр класса `MyDB`.

`MyDB()` — Конструктор класса `MyDB`. В этом методе вызывается метод `init()`, который инициализирует подключение к базе данных.

`void init()` — Метод класса `MyDB`, инициализирующий подключение к базе данных. В этом методе устанавливается тип базы данных (в данном случае "QSQLITE") и имя файла базы данных. Если файл базы данных существует, выводится сообщение "DB file exist". Если подключение к базе данных успешно открыто, выводится сообщение "Database loaded successfull!".

`static MyDB* getInstance()` — Статический метод класса `MyDB`, возвращающий указатель на экземпляр класса `MyDB`. Если экземпляр класса `MyDB` еще не создан, то создается новый экземпляр. Этот метод реализует паттерн проектирования "Синглтон".

`QSqlDatabase getDBInstance()` — Метод класса `MyDB`, возвращающий объект `QSqlDatabase`, который представляет подключение к базе данных. Этот метод используется для получения подключения к базе данных в других частях программы.

`void ResetInstance()` — Метод класса `MyDB`, удаляющий экземпляр класса `MyDB` и сбрасывающий указатель `instance` на `nullptr`. Этот метод используется для освобождения ресурсов и перезапуска класса `MyDB`.

`~MyDB()` — Деструктор класса `MyDB`. В этом методе происходит закрытие подключения к базе данных. Выводится сообщение "closing database connection".

3.3.4 Класс `MainWindow`:

Класс `MainWindow` является основным окном приложения и имеет следующие поля:

`Ui::MainWindow *ui` — Указатель на объект пользовательского интерфейса, с помощью которого можно взаимодействовать с элементами интерфейса главного окна.

`book *ptrBook` — Указатель на объект класса `book`, который представляет диалоговое окно "Book".

`cancel *ptrCancel` — Указатель на объект класса `cancel`, который представляет диалоговое окно "Cancel".

Методы:

`MainWindow(QWidget *parent)` — Конструктор класса `MainWindow`, который принимает указатель на родительский виджет (окно) в качестве аргумента.

`~MainWindow()` — Деструктор класса `MainWindow`, который освобождает ресурсы, занятые объектом класса `MainWindow`.

`void updateUI()` — Метод класса `MainWindow` для обновления пользовательского интерфейса. В данном методе выполняется запрос к базе данных и обновляются соответствующие элементы интерфейса на основе полученных данных.

`void on_btnBook_clicked()` — Слот, вызываемый при нажатии на кнопку "btnBook" в главном окне. В данном методе вызывается метод `updateUI()` объекта класса `book` и отображается окно `book`.

`void on_btnCancel_clicked()` — Слот, вызываемый при нажатии на кнопку "btnCancel" в главном окне. В данном методе вызывается метод `updateUI()` объекта класса `cancel` и отображается окно `cancel`.

`void on_btnReset_clicked()` — Слот, вызываемый при нажатии на кнопку "btnReset" в главном окне. В данном методе выполняется запрос к базе данных для удаления записей пользователей и обновления состояния элементов интерфейса.

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

4.1 Разработка схем алгоритмов

Метод `on_btnBook_clicked()` выполняет процесс бронирования места. Схема метода показана в приложении Б.

Метод `on_btnCancel_clicked()` выполняет удаление записи пользователя из базы данных, основываясь на выбранном месте. Схема метода показана в приложении В.

4.2 Разработка алгоритмов

4.2.1 Метод `on_btnBook_clicked()` класса `book`

Шаг 1: Получаем текущий выбранный элемент из `cmbSeats`.

Шаг 2: Получаем значения из полей ввода `lineEdit` и `lineEdit_2` (имя и фамилия).

Шаг 3: Проверяем валидность введенного имени и фамилии с помощью регулярных выражений.

Шаг 4: Подсвечиваем поля ввода в зависимости от их валидности.

Шаг 5: Если имя или фамилия не валидны, выводит сообщение об ошибке и прерывает выполнение метода.

Шаг 6: Создаём объект `QSqlQuery` для выполнения запросов к базе данных, подготавливаем SQL-запрос для выбора идентификатора (`Id`).

Шаг 7: Привязываем значение выбранного места к подготовленному запросу.

Шаг 8: Выполняем запрос.

Шаг 9: Если запрос выполнен успешно и возвращает результат, то сохраняет идентификатор места (`seatId`).

Шаг 10: Создаем объект `QSqlQuery` для выполнения запросов к базе данных. Подготавливаем SQL-запрос для вставки новой записи в таблицу `users`

Шаг 11: Привязываем значения имени, фамилии и `seatId` к подготовленному запросу.

Шаг 12: Выполняем запрос на вставку новой записи в базу данных.

Шаг 13: Если запрос выполнен успешно, выводит сообщение об успешном бронировании места.

Шаг 14: Создаем объект `QSqlQuery` для выполнения запросов к базе данных. : Подготавливаем SQL-запрос для обновления статуса места на "N" (недоступно)

Шаг 15: Привязываем значение `seatId` к подготовленному запросу.

Шаг 16: Выполнение запроса на обновление статуса места.

Шаг 17: Если запрос выполнен успешно, выводит сообщение об успешном обновлении статуса места.

Шаг 18: Обновляем информацию на пользовательском интерфейсе

4.2.2 Метод `on_btnCancel_clicked()` класса `cancel`

Шаг 1: Получаем текущий выбранный элемент из `cmbSeats`.

Шаг 2: Создаем объект `QSqlQuery` для выполнения запросов к базе данных, подготавливаем SQL-запрос для выбора идентификатора (`Id`).

Шаг 3: Привязываем значение выбранного места к подготовленному запросу.

Шаг 4: Выполняем запрос.

Шаг 5: Если запрос выполнен успешно и возвращает результат, то сохраняет идентификатор места (`seatId`).

Шаг 6: Создаем объект `QSqlQuery` для выполнения запросов к базе данных. Подготавливаем SQL-запрос для удаления записи из таблицы `users`.

Шаг 7: Привязывает значение `seatId` к подготовленному запросу.

Шаг 8: Выполняет запрос на удаление записи из базы данных.

Шаг 9: Если запрос успешно выполнен и удалена хотя бы одна запись, то выводит сообщение и обновляет информацию на пользовательском интерфейсе.

Шаг 10: Если запрос успешно выполнен, но не удалена ни одна запись, то выводит сообщение и обновляет информацию на пользовательском интерфейсе.

5. РЕЗУЛЬТАТ РАБОТЫ

На рисунке 5.1 изображено начало работы программы. Интерфейс при запуске программы в начале работы состоит из главного окна приложения MainWindow. Визуально окно имеет типичный вид для приложений, разработанных с использованием Qt Framework.

Главное окно содержит несколько элементов управления, таких как кнопки и текстовые поля, которые позволяют пользователю взаимодействовать с программой. На нижней панели окна располагаются кнопки забронировать, отменить и сбросить.

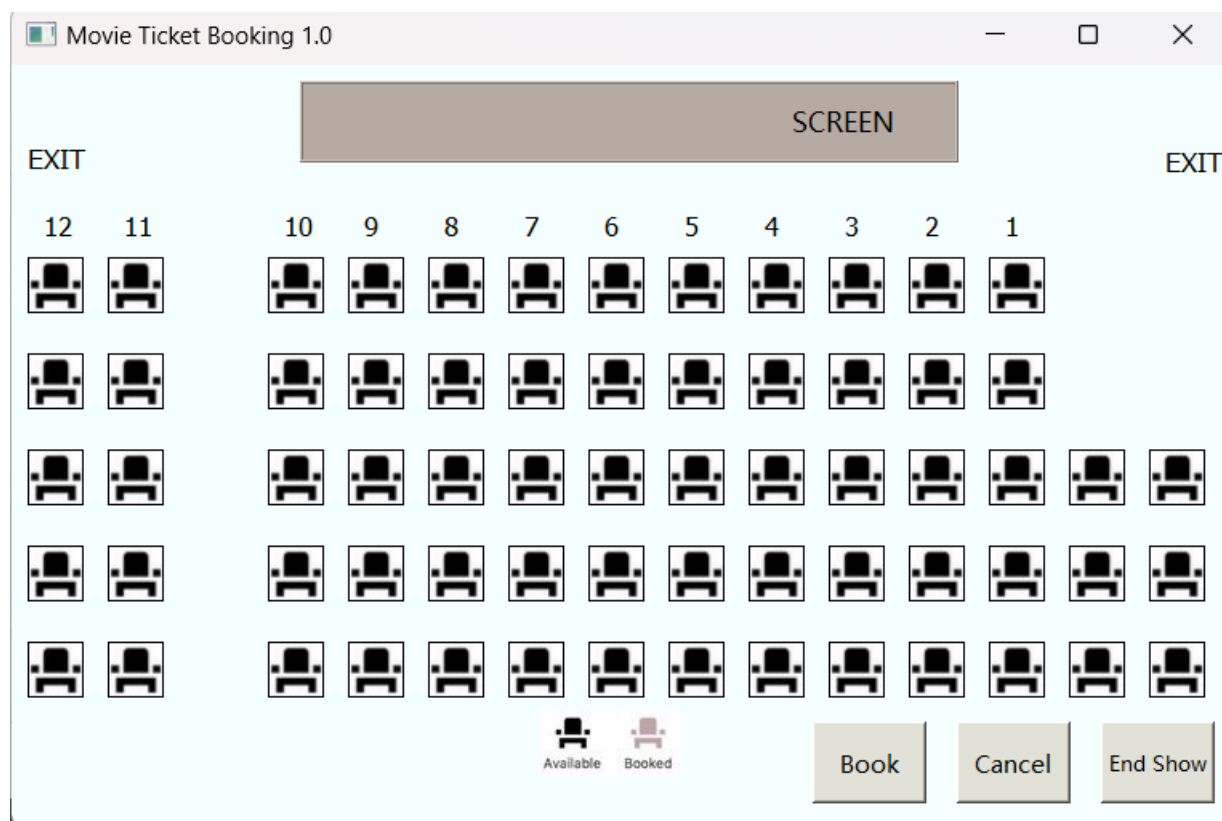


Рисунок 5.1 — Начало работы программы

На рисунке 5.2 показана работа кнопки бронирования. При нажатии кнопки, позволяет пользователю перейти к диалоговому окну бронирования билетов.

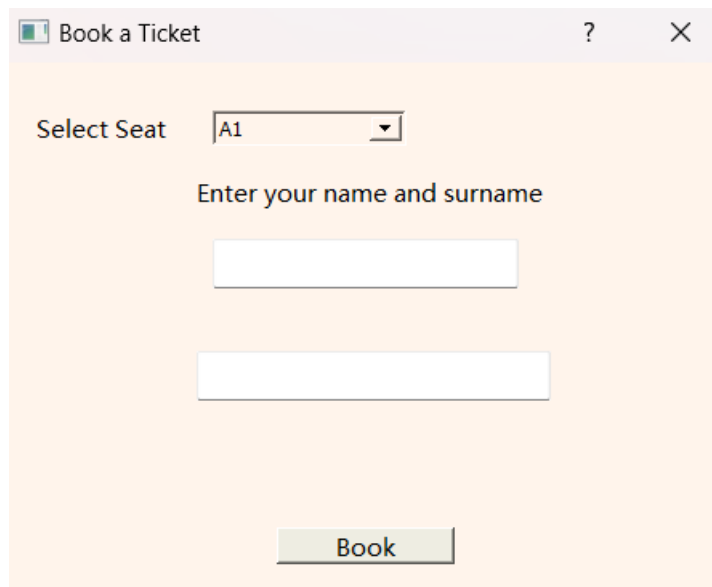


Рисунок 5.2 — Дополнительное окно кнопки бронирования

На рисунке 5.3 показано окно при неуспешном бронировании. Если при нажатии кнопки бронирования пользователь вводит неправильные данные, то программа предусматривает обработку такой ситуации и предоставляет соответствующую обратную связь пользователю.

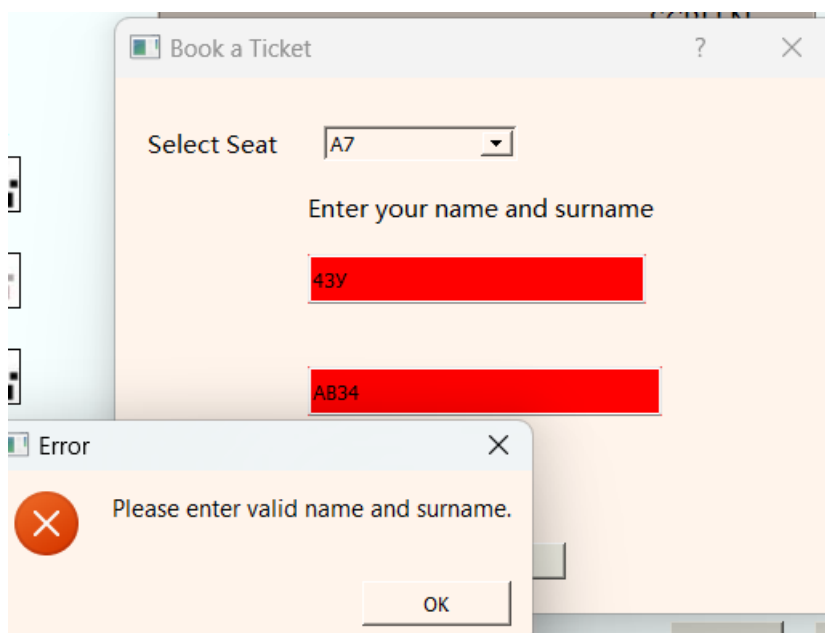


Рисунок 5.3 — Некорректный ввод данных

На рисунке 5.4 показана работа кнопки отмены. При нажатии кнопки, позволяет пользователю перейти к диалоговому окну отмены бронирования билетов.

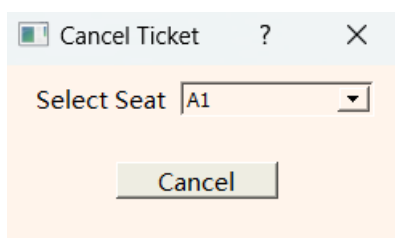


Рисунок 5.4 — Дополнительное окно кнопки отменить

На рисунке 5.5 отражена наша база данных пользователей, которая хранится в виде структурированной коллекции данных, организованной в таблице. Каждый пользователь представлен отдельной записью в таблице, где каждое поле соответствует определенной информации о пользователе.

	id	name	surname	seatId
	Фи...	Фил...	Фильтр	Фильтр
1	31	Vict...	Artish	1
2	32	Feona	Rybak	2
3	33	Stas	Chernyak	17
4	34	Maks	FRE	18
5	35	Anna	Adhamjon	65
6	36	Ikrom	Adhamjon	64
7	37	Arsen	Reshetka	38

Рисунок 5.5 — Таблица базы данных

На рисунке 5.6 отражена наша база данных мест. В базе данных у нас есть таблица с названием "Seats" для хранения информации о местах. Каждая строка в таблице представляет отдельное место и содержит следующие поля: Id - уникальный идентификатор места (целочисленное значение или строка); SeatNumber - номер места (целочисленное значение или строка); Available - флаг доступности места (например, "Y" для доступного места, "N" для забронированного или занятого места).

Таким образом, каждая строка в таблице "Seats" представляет одно место и содержит информацию о его номере и доступности.

В этом примере у нас есть 7 мест, где шесть мест (SeatNumber 1, 2, 4, 5, 6 и 7) доступны (Available равно "Y"), а третье место (SeatNumber 3) забронировано или занято (Available равно "N")

	Id	Seat	Available
	Фи...	Фи...	Фильтр
1	1	A1	Y
2	2	A2	Y
3	3	A8	N
4	4	A7	Y
5	5	A6	Y
6	6	A5	Y
7	7	A4	Y

Рисунок 5.6 — Таблица базы данных

ЗАКЛЮЧЕНИЕ

Итак, изучение объектно-ориентированного программирования (ООП) выдало себя как преобразующий этап в моем понимании и подходе к разработке программного обеспечения. Погружение в мир ООП расширило мои знания и компетенции, обогатив мой инструментарий в области программирования.

Одним из ключевых выводов, которые я сделал в процессе написания курсовой работы, является то, что ООП не только предоставляет эффективные средства организации кода, но и обеспечивает более высокий уровень абстракции, что делает разработку программ более интуитивной и гибкой. Использование классов и объектов позволяет создавать модульные, легко читаемые и поддерживаемые системы.

В наше современное время, когда требования к программному обеспечению постоянно растут, ООП становится неотъемлемой частью профессиональной подготовки разработчика. Адаптивность кода, возможность масштабирования проектов и повторного использования компонентов становятся критическими аспектами успешной разработки. ООП предоставляет технологический фреймворк для разработки сложных, но гибких систем, способных адаптироваться к изменяющимся требованиям бизнеса.

В результате выполнения курсового проекта на тему "Система бронирования мест в кинотеатре" была разработана эффективная и удобная система, которая обеспечивает управление процессом бронирования мест в кинотеатре.

Основной акцент был сделан на создании надежной и интуитивно понятной среды для пользователей, что позволяет им легко и быстро выбирать и бронировать места в кинозале.

В ходе разработки были применены современные методы программирования и использованы соответствующие технологии. Была проведена анализ требований и характеристик системы, что позволило определить необходимые функциональные возможности и реализовать их эффективно.

Система бронирования мест в кинотеатре представляет значимую ценность для кинотеатров и их посетителей. Она упрощает процесс бронирования билетов, повышает удобство для пользователей и помогает кинотеатрам эффективно управлять своими ресурсами.

В результате выполнения курсового проекта были достигнуты поставленные цели и задачи. Разработанная система успешно реализовывает функционал бронирования мест в кинотеатре и может быть использована в реальных условиях. Работа над проектом позволила закрепить и применить полученные знания о разработке программного обеспечения и методах проектирования систем.

Таким образом, разработанная система бронирования мест в кинотеатре предоставляет удобный и эффективный способ управления местами, а ее структура и код являются примером хорошей практики программирования на языке C++ с использованием фреймворка Qt.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] Рожнова, Н. Г. Вычислительные машины, системы и сети. Дипломное проектирование : учебно-метод.пособие / Н. Г. Рожнова, Н. А. Искра, И. И. Глецевич. – Минск : БГУИР, 2014. – 96 с. : ил.
- [2] Шлее М. - Qt4. Профессиональное программирование на C++/ Шлее М. - Л.:Наука, 2013. - 770 с.
- [3] Программирование на C++ [Электронный ресурс]. -Электронные данные. Режим доступа: <https://metanit.com/cpp/tutorial/> -Дата доступа: 23.11.2023.
- [9] "Алгоритмы. Построение и анализ" Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
- [10] "Структуры данных и алгоритмы в C++" Robert Lafore.
- [7] Qt for Beginners [Электронный ресурс]. –Режим доступа: https://wiki.qt.io/Qt_for_Beginners. –Дата доступа: 19.11.2023.
- [8] BookMyShow [Электронный ресурс]. -Режим доступа: <https://in.bookmyshow.com/>. -Дата доступа: 25.11.2023.
- [9] Cinapolis [Электронный ресурс]. -Режим доступа: <https://cinapolis.com/>. -Дата доступа: 25.11.2023.
- [10] Fandango [Электронный ресурс]. -Режим доступа: <https://www.fandango.com/region-error>. -Дата доступа: 25.11.2023.
- [11] Atom Tickets [Электронный ресурс]. -Режим доступа: <https://www.atomtickets.com/>. -Дата доступа: 25.11.2023.

ПРИЛОЖЕНИЕ А
(обязательное)
Диаграмма классов

ПРИЛОЖЕНИЕ Б
(обязательное)
Схема метода *on_btnBook_clicked()*

ПРИЛОЖЕНИЕ В
(обязательное)
Схема метода *on_btnBook_clicked()*

ПРИЛОЖЕНИЕ Г
(обязательное)
Код программы

ПРИЛОЖЕНИЕ Д
(обязательное)
Ведомость документов