

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ «МИСИС»

Институт информационных технологий и компьютерных наук
Кафедра инженерной кибернетики

Курсовая работа

по дисциплине «Технологии программирования»

на тему:

«Приложение для аренды автомобилей»

Выполнил:

студент 2-го курса,
гр. БПМ-21-3 Кочян Л. В.

Проверил:

доцент, к.т.н. Полевой Д. В.

Москва, 2022

Оглавление

1. Описание технического задания	3
1.1.Основные требования к приложению	3
1.2.Пользовательский интерфейс	3
2. Пользовательская документация	4
2.1.Инструкция по использованию приложения	4
2.1.1.Регистрация	4
2.1.2. Авторизация	5
2.1.3.Главное меню	5
2.1.4.Примечание	9
3. Техническая документация.....	9
3.1.Требования для установки и тестирования программы.....	9
3.2.Инструкция по установке программы.....	9
3.3.Инструкция по тестированию программы.....	9
3.4.Модули.....	10
3.4.1.Пространства имен.....	10
3.4.2.Классы.....	10
3.4.3.Класс AppWindow	10
3.4.4.Класс authorization.....	13
3.4.5.Класс Registration	14
3.5.Структура базы данных	15

Описание технического задания

В целях ознакомления с процессом создания кроссплатформенных приложений с графическим интерфейсом и использованием баз данных реализовать полноценную пользовательскую программу для аренды автомобилей на определённый срок.

Основные требования к приложению

1. Наличие графического интерфейса для пользователя.
2. Приложение должно иметь базу данных с информацией о данных пользователей, машин и состоянии аренды.
3. Приложение должно поддерживать внесение изменений о состоянии аренды и наличии доступных машин.
4. Приложение должно уметь регистрировать новых пользователей и авторизировать существующих.

Пользовательский интерфейс

1. Окна регистрации, авторизации и главное меню, в котором содержится информация о наличии свободных машин, состоянии аренды и её стоимости.
2. Текстовые поля для ввода информации о пользователе, а также для выбора автомобиля.
3. Кнопка «Регистрация» переводит пользователя в меню регистрации, где требуется ввести информацию: имя, фамилия, почта, номер телефона, никнейм, пароль и пр.
4. Кнопка «Войти» производит авторизацию пользователя и при корректности введённых данных переводит его в главное меню приложения.
5. Кнопка «Профиль» открывает страницу с информацией о пользователе.
6. Кнопка «Аренда» открывает страницу с информацией о наличии доступных машин.
7. Кнопка «Арендовать» добавляет автомобиль в аренду.
8. Кнопка «Возврат» открывает страницу с информацией о текущих автомобилях в аренде у пользователя.

9. Кнопка «Вернуть» заканчивает аренду автомобиля.
10. Кнопка «Цена» открывает страницу с информацией о рассчитанной стоимости аренды.
11. Кнопка «Выйти» закрывает главное меню и возвращает пользователя к окну авторизации.

Пользовательская документация

Инструкция по использованию приложения

Регистрация

1. Нажать на кнопку «Регистрация».



Рисунок 1 — Страница авторизации в приложении

2. Ввести необходимые данные (поскольку приложение тестовое, можно вводить ненастоящие данные).



Рисунок 2 — Страница регистрации в приложении

3. После введения данных нажать на кнопку «Создать». Данные внесутся в базу данных, и появится возможность авторизоваться.

Авторизация

1. Ввести никнейм и пароль от созданного аккаунта.
2. Нажать кнопку «Войти». При корректности введённых данных откроется главное меню приложения.

Главное меню

1. При нажатии на кнопку «Профиль» откроется страница с информацией о пользователе.

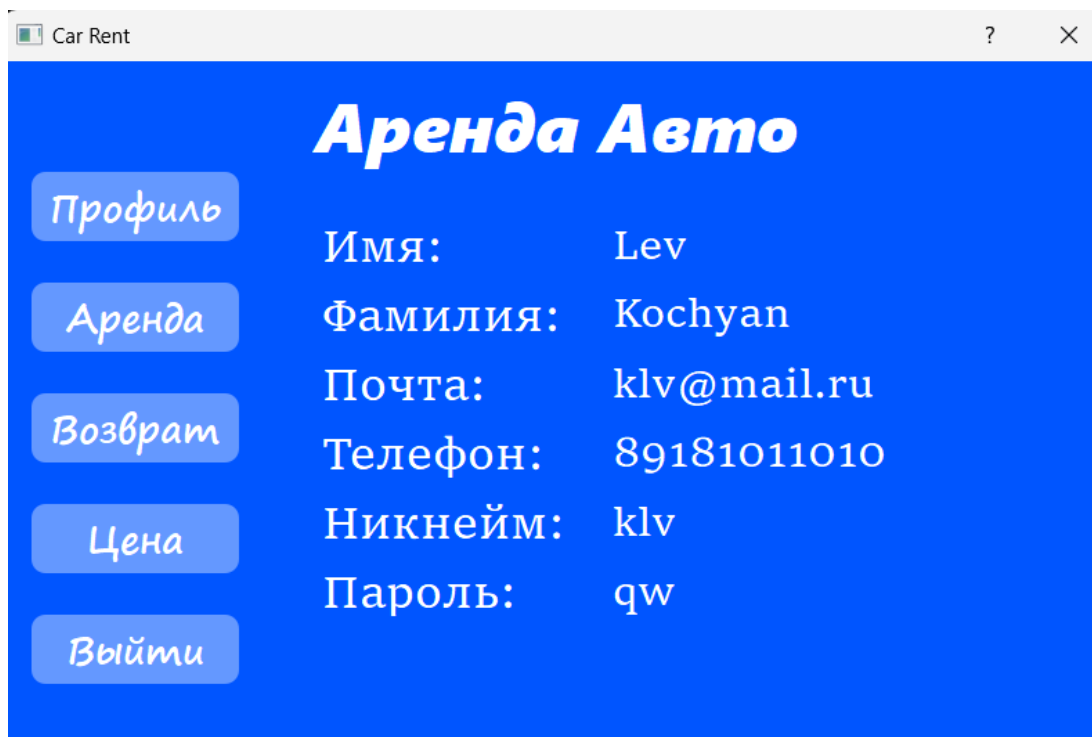


Рисунок 3 — Страница с информацией о профиле пользователя

2. Для аренды нажать на кнопку «Аренда».



Рисунок 4 — Страница аренды автомобилей

3. Выбрать Id желаемого автомобиля, ввести Id и нажать кнопку «Арендовать».



Рисунок 5 — Сообщение об аренде автомобиля

4. Для просмотра состояния аренды нажать на кнопку «Возврат».

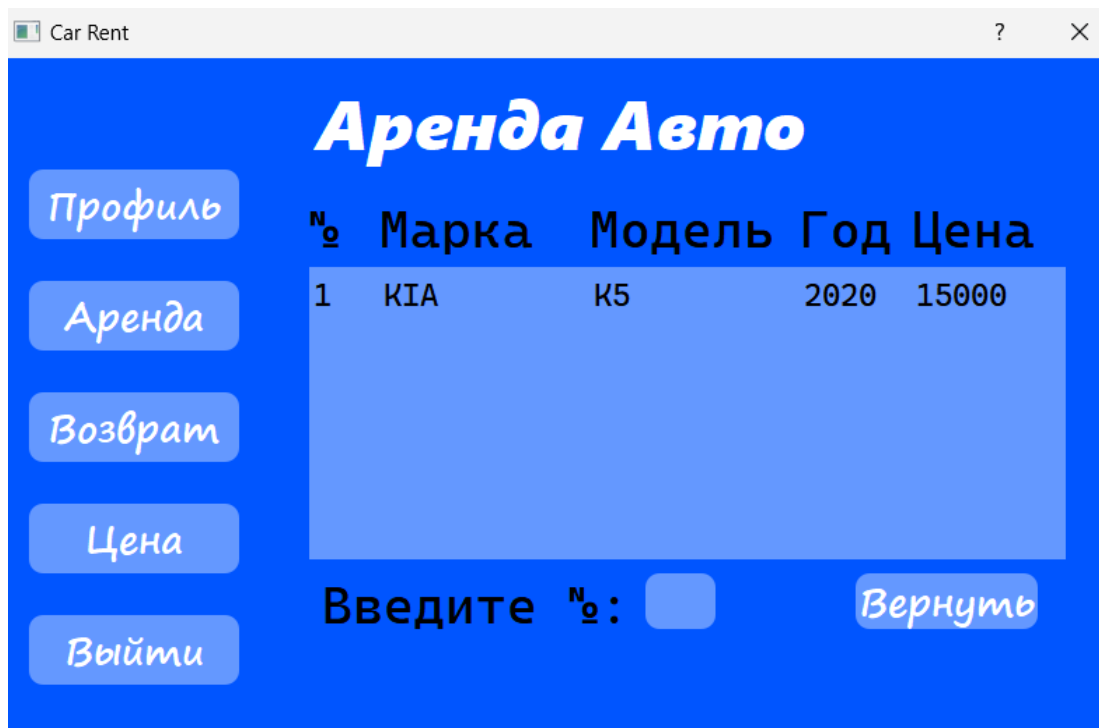


Рисунок 6 — Страница возврата автомобиля

5. Для окончания аренды нажать на кнопку «Вернуть» и ввести Id автомобиля.



Рисунок 7 — Сообщение о возврате автомобиля

6. Для просмотра рассчитанной стоимости аренды нажать на кнопку «Цена».

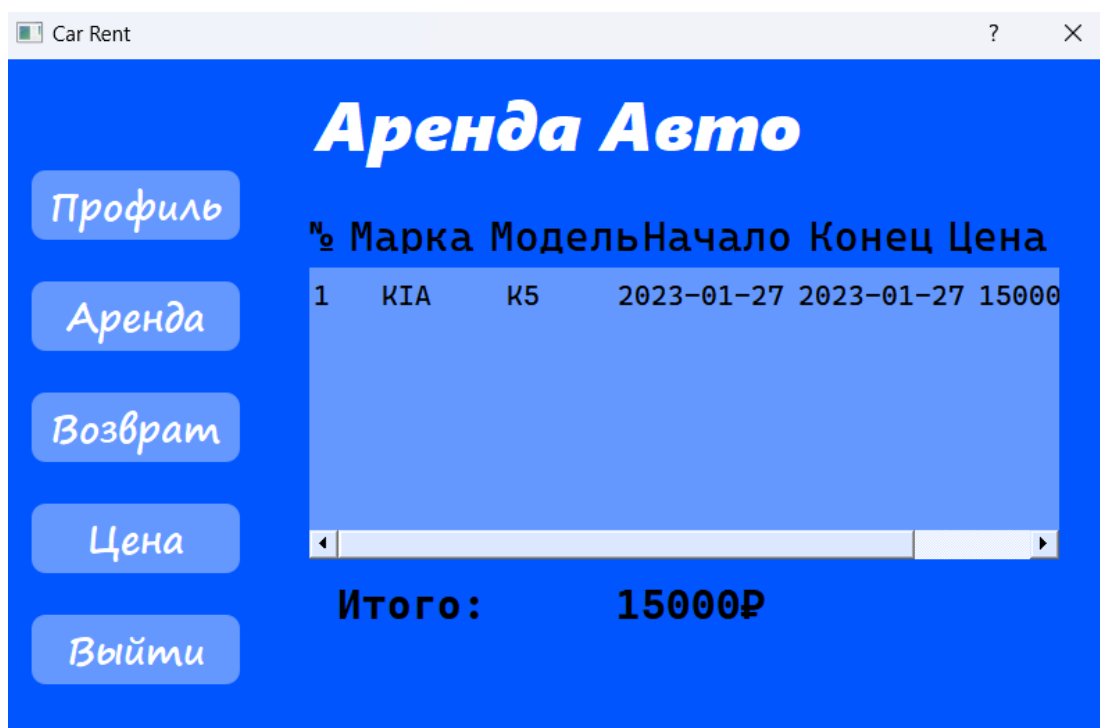


Рисунок 8 — Страница с информацией о стоимости аренды

7. Для выхода из приложения нажать на кнопку «Выйти»

Примечание

Чтобы внести изменения в имеющуюся базу данных (например, добавить новые автомобили), нужно открыть файл carrent.db в директории с файлом autorent.exe с помощью приложений для работы с базами данных (например, HeidiSQL). Далее с помощью SQL-запросов либо вручную можно произвести обновление, изменение, очистку таблиц.

Техническая документация

Требования для установки и тестирования программы

1. Cmake минимальной версии 3.5.
2. Язык C++ стандарта 17.
3. Qt минимальной версии 5.15.2.
4. Компилятор MSVC 2019 (MinGW не всегда компилируется).
5. SQLite и модуль QSQLITE в Qt.

Инструкция по установке программы

1. Выгрузить репозиторий по ссылке: <https://github.com/KochyanLV/car-rental-app>
2. Сконфигурировать проект в CMake.
3. Указать в переменной CMAKE_INSTALL_PREFIX путь к папке, куда установить программу. По умолчанию программа устанавливается в директорию сборки в поддиректорию INSTALL.
4. Сгенерировать проект.
5. Открыть проект (можно через “open project” в CMake).
6. Собрать проект под названием INSTALL.
7. В папке INSTALL запустить autorent.exe.

Инструкция по тестированию программы

1. Запустить autorent.exe
2. Нажать на кнопку «Регистрация» и зарегистрироваться.
3. Авторизироваться в приложении.

4. Проверить открытие страниц главного меню приложения путём нажатия на кнопки «Профиль», «Аренда», «Возврат», «Цена».
5. Арендовать автомобиль по Id автомобиля.
6. Вернуть автомобиль.
7. Проверить корректность стоимости аренды на странице «Цена».
8. Выйти из приложения.
9. Проверить корректность всех действий, просмотрев таблицы базы данных.

Модули

Пространства имен

Пространство имен Ui

Пространство имён **Ui**: используется для группировки автоматически сгенерированных окон в одном пространстве имен. Это помогает различать класс пользовательского интерфейса, созданный из файла пользовательского интерфейса дизайнера, и класс, реализующий функциональность.

Классы

Класс AppWindow

Открытые члены

- **AppWindow (QWidget *parent=nullptr)**
- **~AppWindow ()**

Закрытые слоты

- **void on_ProfileBtn_clicked ()**
- **void on_RentCarBtn_clicked ()**
- **void on_ReturnCarBtn_clicked ()**
- **void on_PriceBtn_clicked ()**
- **void on_LogoutBtn_clicked ()**
- **void on_RentBtn_clicked ()**
- **void on_ReturnBtn_clicked ()**

- **void refreshTableViews ()**

Закрытые данные

- **Ui::AppWindow * ui**

Подробное описание

Главный класс, отображающий графический интерфейс (главное меню нашего приложения) и содержащий в себе обработчики пользовательского ввода.

Конструктор(ы)

AppWindow::AppWindow (QWidget * parent = nullptr)[explicit]

Конструктор класса Appwindow. В аргументы конструктора можно передать указатель на родительский класс, что передаст управление памятью экземпляру родительского класса. В конструкторе происходит поиск и открытие доступных динамических библиотек, создаётся экземпляр потока-обработчика.

AppWindow::~~AppWindow ()

Деструктор класса Appwindow. Удаляет динамически выделенную память, высвобождает динамические библиотеки, останавливает и удаляет потокообработчик.

Методы

void AppWindow::on_LogoutBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Выйти", осуществляющий выход из профиля и переход на страницу входа в приложение.

void AppWindow::on_PriceBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Цена", осуществляющий переход на страницу приложения, где содержится информация об итоговой стоимости аренды.

void AppWindow::on_ProfileBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Профиль", осуществляющий переход на страницу приложения, где содержится основная информация о пользователе.

void AppWindow::on_RentBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Арендовать", осуществляющий аренду автомобиля, обновление базы данных и внесение изменений в неё.

void AppWindow::on_RentCarBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Аренда", осуществляющий переход на страницу приложения, где содержится информация о доступных для аренды автомобилях.

void AppWindow::on_ReturnBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Вернуть", осуществляющий возврат автомобиля из аренды, обновление базы данных и внесение изменений в неё.

void AppWindow::on_ReturnCarBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Возврат", осуществляющий переход на страницу приложения, где содержится информация о находящихся в аренде у пользователя автомобилях.

void AppWindow::refreshTableViews ()[private], [slot]

Метод обновления базы данных.

Данные класса

Ui::AppWindow* AppWindow::ui[private]

Поле, хранящее указатель на экземпляр, отвечающий за графический интерфейс и его элементы.

Класс authorization

Открытые члены

- **authorization (QWidget *parent=nullptr)**
- **~authorization ()**

Закрытые слоты

- **void on_CreateBtn_clicked ()**
- **void on_SingInBtn_clicked ()**

Закрытые данные

- **Ui::authorization * ui**

Подробное описание

Класс, который предоставляет функционал для авторизации существующих пользователей и осуществляет переход на главную страницу приложения.

Конструктор(ы)

authorization::authorization (QWidget * *parent* = nullptr)[explicit]

Конструктор класса authorization. В аргументы конструктора можно передать указатель на родительский класс, что передаст управление памятью экземпляру родительского класса. В конструкторе происходит поиск и открытие доступных динамических библиотек, создаётся экземпляр потока-обработчика.

authorization::~~authorization ()

Деструктор класса authorization. Удаляет динамически выделенную память, высвобождает динамические библиотеки, останавливает и удаляет потокобработчик.

Методы

void authorization::on_CreateBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Регистрация", осуществляющий переход на страницу регистрации новых пользователей.

void authorization::on_SingInBtn_clicked ()[private], [slot]

Слот-обработчик нажатия на кнопку "Войти", осуществляющий вход пользователя в приложения и проверяющий корректность введённых данных.

Данные класса

Ui::authorization* authorization::ui[private]

Поле, хранящее указатель на экземпляр, отвечающий за графический интерфейс и его элементы.

Класс Registration

Открытые члены

- **Registration (QWidget *parent=nullptr)**
- **~Registration ()**

Закрытые слоты

- **void on_RegistrBtn_clicked ()**

Закрытые данные

- **Ui::Registration * ui**

Подробное описание

Класс, который предоставляет функционал для регистрации новых пользователей и добавления их данных в таблицу базы данных.

Конструктор(ы)

Registration::Registration (QWidget * *parent* = nullptr)[explicit]

Конструктор класса `Registration`. В аргументы конструктора можно передать указатель на родительский класс, что передаст управление памятью экземпляру родительского класса. В конструкторе происходит поиск и открытие доступных динамических библиотек, создаётся экземпляр потока-обработчика.

`Registration::~~Registration ()`

Деструктор класса `Registration`. Удаляет динамически выделенную память, высвобождает динамические библиотеки, останавливает и удаляет потокобработчик.

Методы

`void Registration::on_RegistrBtn_clicked ()[private], [slot]`

Слот-обработчик нажатия на кнопку "Создать", осуществляющий регистрацию новых пользователей, переход на страницу авторизации и добавления данных в таблицу базы данных.

Данные класса

`Ui::Registration* Registration::ui[private]`

Поле, хранящее указатель на экземпляр, отвечающий за графический интерфейс и его элементы.

Структура базы данных

1. Имеется 3 таблицы: “cars” (машины), “usersdata” (данные пользователей) и “rentdatedb” (даты аренд).
2. В таблице “cars” (машины) содержится информация о наличии машин, возможности аренды, модели и марки автомобилей, год выпуска.
3. В таблице “usersdata” (данные пользователей) содержится информация о данных пользователей: имя, фамилия, почта, номер телефона, никнейм, пароль.

4. В таблице “rentdatedb” (даты аренд) содержится информации о состоянии аренды, начале и конца аренды, Id машины и стоимости аренды.