МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине: «*Технология программирования»*

на тему: «АРМ кассира кинотеатра»

Выполнил:
Студент гр. *АВТ-710, АВТФ*
Покалюк Евгений Александрович
«___» _____ 2019г.
_____
(подпись)

Проверил:
к.т.н., доцент каф. ВТ
Васюткина Ирина Александровна
«___» _____ 2019г.
_____
(подпись)

Новосибирск
2019

Оглавление

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Введение (описание предметной области)

Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена Sun Microsystems в 1995 г. Существует множество приложений и веб-сайтов, которые не работают при отсутствии установленной Java, и с каждым днем число таких веб-сайтов и приложений увеличивается. Java отличается быстротой, высоким уровнем защиты и надежностью. От портативных компьютеров до центров данных, от игровых консолей до суперкомпьютеров, используемых для научных разработок, от сотовых телефонов до сети Интернет — Java повсюду! Программы на Java транслируются в байт-код, который затем выполняется виртуальной машиной Java (JVM). JVM — это программа, которая обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор. Достоинством подобной реализации является независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует JVM.

Назначение разработки, ее актуальность

Данное приложение развивает одно из направлений научно-технического прогресса, такого как автоматизация. С целью освобождения человека от участия в процессах получения, передачи информации. Автоматицазия позволяет повысить уровень производительности труда, улучшить качество продукции, оптимизировать процессы управления.

Требования к программе

В приложении реализован следующий функционал:

- Работа с окнами
  - Окно авторизации, в котором требуется ввести логин и путь к базе данных;
  - Окно кассира кинотеатра, с возможностью просмотра, покупки или бронирования мест в кинотеатре;
  - Окно клиента кинотеатра, в котором хранится описание фильма и кинозала, время сеанса и расположение свободных мест в выбраном кассиром кинозале;
- Работа с базой данных
  - Хранение актуальных фильмов для просмотра, хранение сеансов, хранение кинозалов.
  - Покупка или бронирование мест в кинотеатре.
- Реализация программы
  - Возможность работы с одним и более клиентом одновременно.
  - Синхронизация баззы данных к неограниченному количеству клиентов.

- Для покупки или бронирования места в кинотеатре у кассира есть возможность выбора фильма из списка доступных, хранящихся в базе данных. После выбора фильма, кассир должен выбрать сеанс и кинозал, а также тип билета.

Постановка задачи (формулирование задач, подлежащих решению)

Задание: Разработать программу «АРМ кассира кинотеатра».

Разработать клиент-серверное приложение, автоматизирующий продажу билетов на несколько фильмов и сеансов в кинотеатре. Иметь схемы нескольких залов. Реализовать графический интерфейс выбора мест в зале. Данные продаж сохранять в базе данных.

Написанная программа должна использовать только встроенный функционал языка Java. Графический интерфейс необходимо разработать на встроенной библиотеке Swing. Программа должна записывать данные в БД.

Задачи:

- Выбор версии Java Development Kit (далее JDK)
- Разработка графического интерфейса
- Разработка архитектуры программы
- Создание программы
  - Создание БД

ПРОЕКТИРОВАНИЕ

Для разработки приложения была выбрана JDK версии 12, поскольку на сегодняшний день является актуальной и подходит исключительно под x64 архитектуру.

Для создания программного кода данного ПО было спроектирована структура, представленная на рисунке 1.

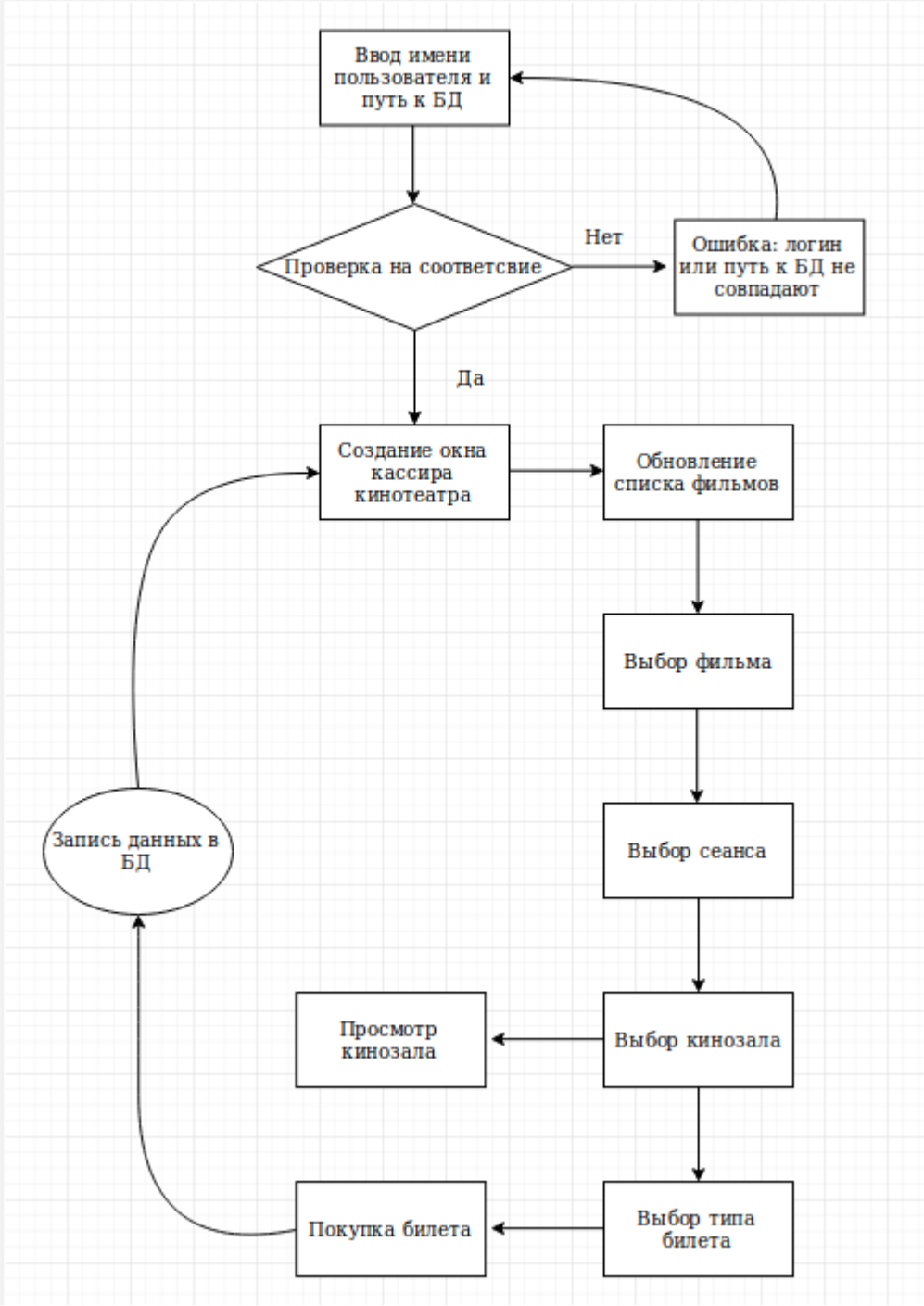

Рисунок 1 Схема последовательности действий

Данный подход разделяет графический интерфейс от работы с данными. В результате чего при изменении логики работы не придётся изменять классы, связанные с отображением содержимого. Внутри модуля графической части классы имеют тип полноценного окна (формы). Работу с данными можно разделить на тип:

- Авторизация — здесь храниться все, для входа в приложение.
- Кассир — в данном модуле хранятся все классы для работы с логикой кассира кинотеатра.

- Клиент — в данном модуле хранятся все классы для работы с логикой клиента.

СТРУКТУРНОЕ ОПИСАНИЕ РАЗРАБОТКИ

Описание структуры программы, ее модулей

Программа разделена на 4 основных модулей: cinema, components, frames, lib.

Модуль cinema содержит внутри себя всю основную логику программы.

components предоставляет из себя набор компонентов, необходимых для отображения графического интерфейса кинозала выбранного кассиром. Графический интерфейс написан на библиотеке Swing.

frames предоставляет из себя набор компонентов, необходимых для отображения графического интерфейса окон для работы внутри программы. Графический интерфейс написан на библиотеке Swing и состоит из графического окна кассира кинотатра, клиента, авторизации.

lib предоставляет из себя набор компонентов, необходимых для отображения графического интерфейса окон для работы с программой. Графический интерфейс написан на библиотеке Swing.

Описание классов: спецификации данных, методов

Исходный код программы состоит из 13 публичных классов и 8 интерфейсов. Рассмотрим каждый класс и для чего он предназначен.

В модуле lib содержаться следующие классы:

- Main – класс, в котором отображается основное графическое окно, с помощью которого пользователь может:
  - Запустить окно авторизиции
  - Ввод имени пользователя, путь к БД
  - При соответсвии логина и пути к БД запустить окно кассира кинотеатра
- Cheked – класс, который не имеет подклассов. Служит для обработки исключений.

```
public static <T, R> Function<T, R> Checked(IChecked<T, R> e) {
      return (T i) -> {
          try {
              return e.apply(i);
          } catch (Exception ex) {
              ex.printStackTrace();
              throw new RuntimeException(ex);
          }
      };
  }

  public static ActionListener CheckedActionListener(CheckedActionListener l)
```

```
{
        return e -> {
            try {
                l.actionPerformed(e);
            } catch (Exception ex) {
                ex.printStackTrace();
                throw new RuntimeException(ex);
            }
        };
    }

    public static ListSelectionListener
CheckedListSelectionListener(CheckedListSelectionListener l) {
        return e -> {
            try {
                l.valueChanged(e);
            } catch (Exception ex) {
                ex.printStackTrace();
                throw new RuntimeException(ex);
            }
        };
    }
}
```

- IAuth – интерфейс. Отвечает за вызов метода login.

```
public interface IAuth {
    void login(String url, String username, String password) throws Exception;
}
```

В модуле frames содержаться следующие классы:

- Auth – класс наследуемый публичным классом Jframe из библиотеки Swing, в котором отображается основное графическое окно авторизации, с помощью которого пользователь может ввести имя пользователя и путь к БД. Также, класс отвечает за отправление данных для подтвержения входа в приложение.

```
            try {
                    // 127.0.0.1:3306/cinema – путь к БД
                    auth.login("jdbc:mysql://" + localhostField.getText(),
loginField.getText(), passwordField.getText());
                    lMistake.setVisible(false);
                    lAccept.setVisible(true);
                    setVisible(false);

                    synchronized (this) {
                        notify();
```

```
        }
    }
```

- Cashier – класс наследуемый публичным классом Jframe из библиотеки Swing, в котором отображается основное графическое окно кассира кинотеатра и диалоговое окно,а также логику и обработку событий:
    - Обновление списка фильмов, находящихся в БД.
    - Возможность выбора фильма, сеанса, кинозала, типа билета, находящихся в БД.
    - Возможность просмотра кинозала
    - Возможность покупки, бронирования места в кинотеатре
    - Создание окна клиента

Исполнение событий:

По нажатию кнопки, обновляет список фильмов:

```
btnCurrentMovies.addActionListener(CheckedActionListener(e -> {
    movieList.setListData(cinema.getMovies());
    movieList.updateUI();
}));
```

По нажатию на один из элементов списка, обновляет список доступных сеансов для выбранного фильма:

```
movieList.addListSelectionListener(CheckedListSelectionListener(e -> {
    sesssList.setListData(cinema.getShowtimes((IMovie) movieList.getSelectedValue()));
    sesssList.updateUI();
}));
```

По нажатию на один из элементов списка, обновляет список доступных кинозалов для выбранного фильма в соответсвии с его сеансом:

```
sesssList.addListSelectionListener(CheckedListSelectionListener(e -> {
    roomsList.setListData(cinema.getRooms((IShowTime) sesssList.getSelectedValue()));
    roomsList.updateUI();
}));
```

По нажатию на один из элементов списка, пользователь может выбрать кинозал доступный исходя из прошлых запросов:

```
roomsList.addListSelectionListener(CheckedListSelectionListener(e ->{
    viewroomButton.setEnabled(true);
}));
```

Выбор типа билета:

```
priceList.addListSelectionListener(e -> {
    if (!movieList.isSelectionEmpty() && !sesssList.isSelectionEmpty() && !
roomsList.isSelectionEmpty())
    seatButton.setEnabled(true);
});
```

- Customer – класс наследуемый публичным классом Jframe из библиотеки Swing, в котором отображается основное графическое окно, с помощью которого пользователь может:
    - Получить информацию о названии фильма, начале сеанса, выбранном кинозале, а также пользователь увидит кинозал с уже купленными и свободными местами.

В модуле components содержаться следующий класс:

- Room – класс наследуемый публичным классом JPanel из библиотеки Swing, в котором выолняется отрисовка мест в кинозале.

Используется коллекция ArrayList:

```
ArrayList<RoomListener> listeners = new ArrayList<>();
...
public void addActionListener(RoomListener l) {
    listeners.add(l);
}
```

Отрисовка мест в кинозале:

```
seatsPanel.setLayout(new GridLayout(room.getHeight(), room.getWidth()));
for (int y = 1; y <= room.getHeight(); ++y) {
    for (int x = 1; x <= room.getWidth(); ++x) {
        final int column = x, row = y;

        buttons[y][x] = new JButton(String.format("%d:%d", y, x));
        buttons[y][x].addActionListener(e -> {
            for (RoomListener l : listeners) {
                l.actionPerformed(new RoomEvent() {
                    final int Column = column;
                    final int Row = row;

                    public int getColumn() {
                        return Column;
                    }

                    public int getRow() {
                        return Row;
                    }
                });
            }
        });

        seatsPanel.add(buttons[y][x]);
    }
}
```

В модуле cinema содержаться следующие классы:

Cinema – класс реализуемый интерфейсом Icinema, IAuth, в котором выполняется основная логика программы и соединения с БД.

Movie – класс реализуемый интерфесом IMovie, в котором хранится информация о ID фильма, название фильма, описание фильма, процент стоимости билета.

```
public Movie(int id, String name, String description, int percent) {
    Id = id;
    Name = name;
    Description = description;
    Percent = percent;
}
```

Offer – класс реализуемый интерфесом IOffer, в котором хранится информация о ID заказа, название фильма, сеансе, кинозале.

```
public Offer(int id, IMovie movie, IShowTime showtime, IRoom room) {
    Id = id;
    Movie = movie;
    ShowTime = showtime;
    Room = room;
}
```

Order – класс реализуемый интерфесом IOrder, в котором хранится информация о ID, название фильма, сеансе, кинозале, ценна, ряд и строка который содержит в себе места кинозала.

```
public Order(int id, IMovie movie, IShowTime showtime, IRoom room, IPrice
price, int row, int column) {
    Id = id;
    Movie = movie;
    Showtime = showtime;
    Room = room;
    Price = price;
    Row = row;
    Column = column;
}
```

Price – класс реализуемый интерфесом IPrice, в котором хранится информация о ID типа билета, название типа билета, процент.

```
public Price(int id, String name, int percent) {
    Id = id;
    Name = name;
    Percent = percent;
}
```

Room – класс реализуемый интерфесом IRoom , в котором хранится информация о ID кинозала, название кинозала, описание кинозала, ширина и высота кинозала.

```
public Room(int id, String name, String description, int width, int height) {
    Id = id;
```

```
            Name = name;
            Description = description;
            Width = width;
            Height = height;
        }
```

ShowTime – класс реализуемый интерфесом IShowTime, в котором хранится информация о ID сеанса, времени: час и минута, процент.

```
public ShowTime(int id, int hours, int minutes, int price) {
    Id = id;
    Hours = hours;
    Minutes = minutes;
    Price = price;
}
```

ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ РАЗРАБОТКИ

Описание алгоритмов и методов решения.

Рассмотрим создание мест в кинозале. Мы запрашиваем ширину и высоту объекта room, принимаем значения x,y которые лежат в БД в выбранного кинозала. Формат %d, принимает значения типа int, запрашиваем количество столбцов column и строк row:

```
seatsPanel.setLayout(new GridLayout(room.getHeight(), room.getWidth()));
for (int y = 1; y <= room.getHeight(); ++y) {
    for (int x = 1; x <= room.getWidth(); ++x) {
        final int column = x, row = y;

        buttons[y][x] = new JButton(String.format("%d:%d", y, x));
        buttons[y][x].addActionListener(e -> {
            for (RoomListener l : listeners) {
                l.actionPerformed(new RoomEvent() {
                    final int Column = column;
                    final int Row = row;

                    public int getColumn() {
                        return Column;
                    }

                    public int getRow() {
                        return Row;
                    }
                });
            }
        });

        seatsPanel.add(buttons[y][x]);
    }
```

Базы данных

Рассмотрим подключение пользователя к АРМ кассира кинотеатра. Программа использует DriverManager, который устанавливает соединение между БД и драйвером, которое такэе принимает такие значения как имя пользователя, пароль и путь к БД

```
<T> ArrayList<T> performQuery(String sql, Function<ResultSet, T> create) throws SQLException
{
    ArrayList<T> values = new ArrayList<>();

    try(var connection = DriverManager.getConnection(Url, User, Password)) {
        ResultSet rs = connection.createStatement().executeQuery(sql);
        while (rs.next()) {
            values.add(create.apply(rs));
        }
    }

    return values;
}
```

Рассмотрим получение списка фильмов с БД. Список фильмов получет модель абстрактного списка запрашиваю размер и длину списка вызовом метода setListData обращаясь к БД cinema интерфейса IMovie → getMovies и обновляет список:

```
btnCurrentMovies.addActionListener(CheckedActionListener(e -> {
    movieList.setListData(cinema.getMovies());
    movieList.updateUI();
}));
```

Параллельные потоки

У данной программы имеются параллельные потоки, отвечающие за различные события. При запуске более одного клиента программа создаёт ещё один потом отвечающий за клиент кассира каинотеатра:

```
private static Cinema cinema = new Cinema();
private static Cinema cinema2 = new Cinema();
private static Thread cinemaThread = new Thread(cinema);
private static Thread cinemaThread2 = new Thread(cinema);

 try {
        cinemaThread.run();
        Auth auth = new Auth(cinema);
        auth.setMinimumSize(new Dimension(350, 200));

        // wait for user login
        synchronized(auth) {
            auth.wait();
        }

        Cashier cashierFrame = new Cashier(cinema);
        cashierFrame.setVisible(true);
        cashierFrame.setTitle("Окно кассира кинотеатра");

    }
```

```
    catch (Exception e) {
        e.printStackTrace();
    }

    try {
        cinemaThread2.run();
        Auth auth = new Auth(cinema2);
        auth.setMinimumSize(new Dimension(350, 200));

        // wait for user login
        synchronized(auth) {
            auth.wait();
        }

        Cashier cashierFrame = new Cashier(cinema2);
        cashierFrame.setVisible(true);
        cashierFrame.setTitle("Окно кассира кинотеатра");

    }

    catch (Exception e) {
        e.printStackTrace();
    }
}
```

ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

На рисунке 2 предоставлено окно авторизации. Оно содержит JTextField для ввода имени пользователя, JTextField для ввода пароля, JTextField для ввода пути в БД, JButton «Войти» для входа в окно кассира кинотеатра, JButton «Выйти» для выхода из программы.

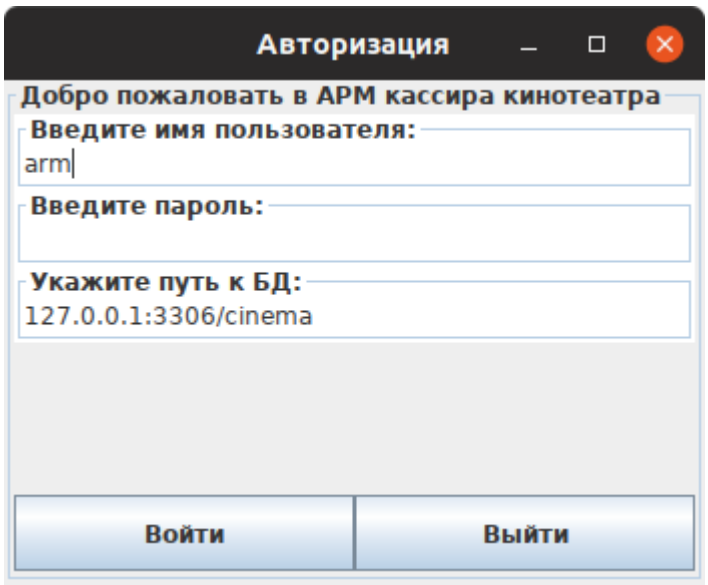

Рисунок 2 — Окно авторизии программы

На рисунке 3 содержаться следующие элементы:

1. Панель JPanel названа mainPanel которая является основной панелью для размещения компонентов.
2. Панель JPanel названа ctrlPanel которпя является левой панелью для размещения компонентов с которыми пользователь может взаимодействовать.
3. Панель JPanel которая названа roomPanel которпя является правой панелью для размещения выбранного кассиром кинозала.
4. Список JList movieList который хранит в себе список фильмов
5. Список JList priceList который хранит в себе типы билетов
6. Список JList roomsList который хранит в себе список кинозалов
7. Список JList sessList который хранит в себе список сеансов
8. Кнопка JButton seatButton отвечает за покупку места
9. Кнопка JButton viewroomButton отвечает за просмотр кинозала
10. Кнопка JButton exitButton отвечает за выход из программы
11. Кнопка JButton btnCurrentMovies отвечает обновление списка фильмов
12. Строка меню JmenuBar menuBar в который входило:
    a. Меню JMenu fileMenu «Файл» в который входит:
        i. Элемент меню JMenuItem «Инстукция по использованию» отвечает за открытие инструкции для использования
        ii. Элемент меню JMenuItem «Выход» отвечает за выход из программы
    b. Меню JMenu helpMenu «Помощь»
        i. Элемент меню JMenuItem LearnItem «Изучение Java» отвечает за открытие pdf файла с учебно-методическим пособием.
        ii. Элемент меню JMenuItem helpItem «Как написать АРМ» отвечает за открытие pdf файла с методическим пособием по написанию курсовой работы.
    c. Меню JMenu aboutMenu «О программе» вызывает диалоговое окно с информацией о программе.

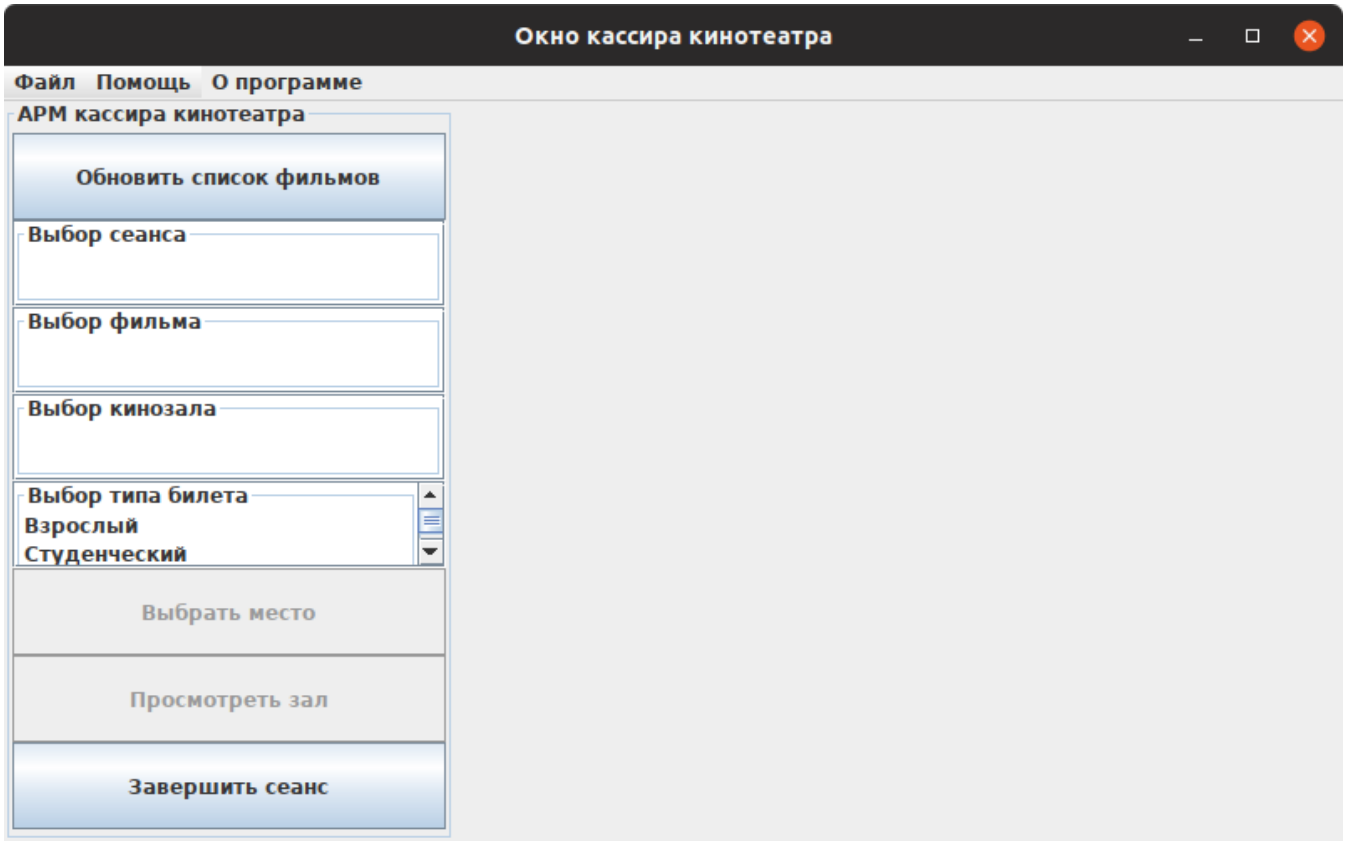

Рисунок 3 — Окно кассира кинотеатра

При нажатии на кнопку выбора места пользователь увидит выбранный им зал с купленными и свободными местами (рисунок 4).

Рисунок 4 — окно кассира кинотеатра

На рисунке 5 содержаться следующие элементы:

Данное окно содержит следующие элементы:

1. Панель JPanel названа mainPanel которая является основной панелью для размещения компонентов.
2. Панель JPanel названа ctrlPanel которпя является левой панелью для размещения компонентов с которыми пользователь может взаимодействовать.
3. Панель JPanel которая названа roomPanel которпя является правой панелью для размещения выбранного кассиром кинозала.
4. Надписи Jlabel movLabel, roomLabel, shwLabel которые отвечают за наименование заголовка названия фильма, кинозала, сеанса.
5. Текстовые поля JTextArea descField, roomLabelName, durtField которые отвечают за описание фильма, кинозала, а также время сеанса.

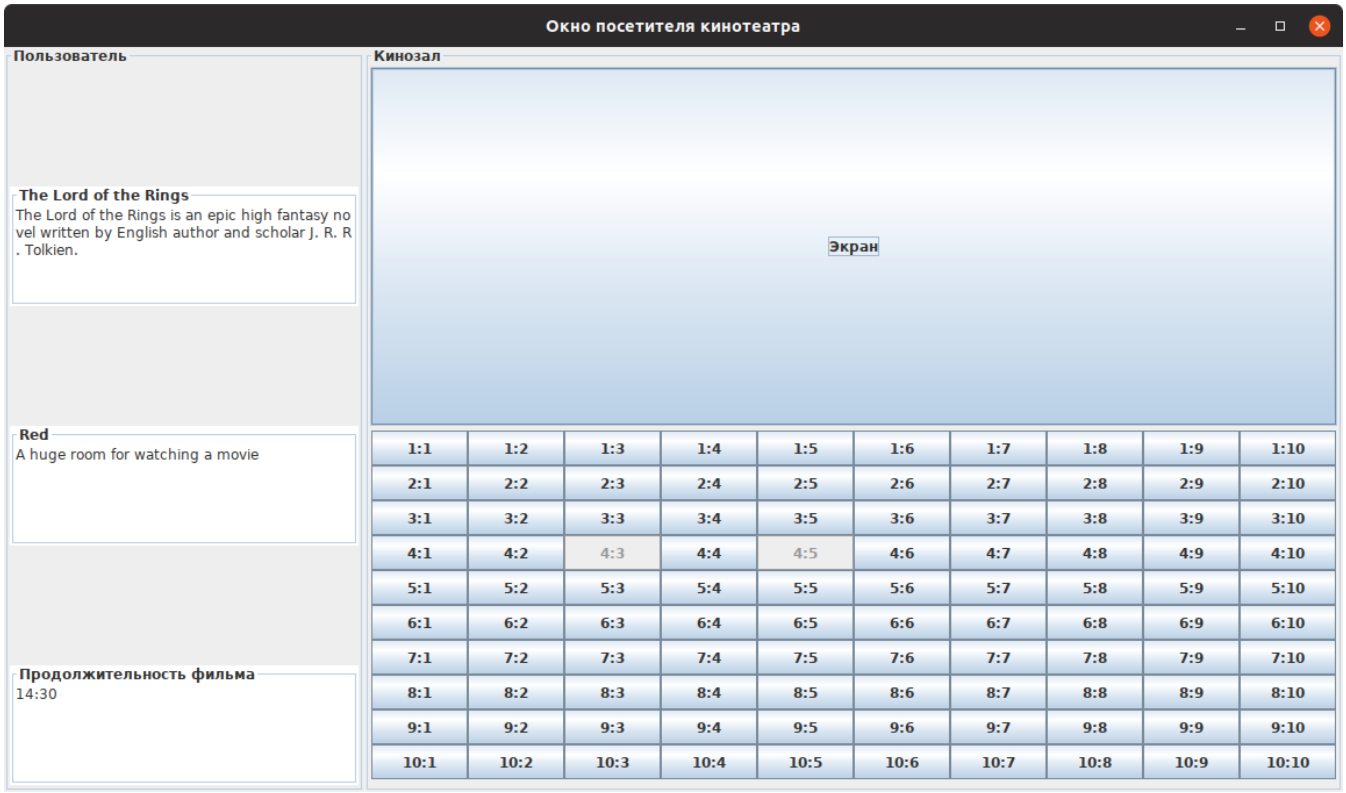

Рисунок 5 — окно клиента кинотеатра

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

После запуска программы вас встречает окно авторизации. На нем находятся текстовые поля для ввода данных и кнопки управления. Перед началом работы авторизируйтесь используя имя пользователя, путь к БД. В случае ввода неправильных данных появиться ошибка, инче вы попадёте в окно кассира кинотеатра:

1. Нажмите нужную кнопу «Обновить список фильмов» . После нажатия кнопки, список «Выбор фильма», расположенный ниже, покажет вам доступные фильмы.
2. Нажмите на один из фильмов, который вам требуется. После нажатия на один из элементов списока, «Выбор сеанса», расположенный ниже, покажет вам доступные сеансы для выбранного вам фильма.
3. Нажмите на один из сеансов, который вам требуется. После нажатия на один из элементов списока, «Выбор кинозала», расположенный ниже, покажет вам доступные кинозалы для выбранного вам фильма. Для вас доступна кнопка «Просмотреть зал», которая позволит вам просмотреть кинозал с купленными и свободными местами выбранного фильма во время выбранного сеанса.
4. Нажмите на один из типов билета, который вам подходит. После нажатия на один из типов билетов, для вас будет доступна кнопка «Выбрать место», где мышкой требуется выбрать одно или несколько мест которые необходимо купить.

Для выхода из программы используйте кнопку «Завершить сеанс» или в строке меню находиться кнопка «Выход».

РУКОВОДСТВО ПРОГРАММИСТА

Программа не требует установки, но для обеспечения его работоспособности необходимо наличие JDK 12, а также соблюдение минимальных технических требований для стабильной работоспособности данного компонента.

ЗАКЛЮЧЕНИЕ

Была разработана программа, которая позволяет покупать, бронировать места в кинотеатре. Были написаны таблицы, которые хранятся в базе данных на официально зарезервированном доменном имени для частных IP-адресов. Разработаны потоки, обрабатывающие авотризацию нескольких клиентов, обрабатывающие информацию класса cinima. Был разработан графический интерфейс пользователя. Программа является армтоматизировнным рабочим местом кассира кинотеатра.

Список литературы

1. **Эккель Б.** Философия Java. Библиотека программиста [Книга]. - [б.м.] : СПб: Питер, 2013. - 4-е издание : стр. 640.
2. **Васюткина И.А.**, канд. Иехн. Наук, доцент. ТЕХНОЛОГИЯ РАЗРАБОТКИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММ НА JAVA [Книга]. - [б.м.]: Нск: Новосибирск, 2012. - стр. 165.

3. **Эндрюс Г. Р.** Основы многопоточного, параллельного и распределенного программирования [Книга]. - [б.м.] : Издательский дом "Вильямс", 2003. - стр. 512.
4. **Шилдт Г.** Java 8. Полное руководство [Книга]. - [б.м.] : Oracle Press, 2017. - 9-е изд. : стр. 1355