Государственное бюджетное общеобразовательное учреждение

средняя общеобразовательная школа № 521

с углубленным изучением математики и информатики

Красногвардейского района Санкт-Петербурга

Проект

**Movielib: программное решение для поиска и хранения информации о любимых фильмах (сериалах)**

**Вид работы: программный код**

Программирование

**Выполнил:**

Воробьев Антон Николаевич,

ученик 11-А класса

**Руководитель:**

Мальцев А.М.,

учитель информатики

ГБОУ СОШ № 521

Санкт-Петербург

2024

**Содержание**

|  |  |
| --- | --- |
| I. Введение | 3 – 4 |
| II. Основная часть | 4 – 8 |
| III. Календарный план-график | 9 |
| IV. Список информационных ресурсов и приложения | 10 |
| Приложение | 11 – 12 |

1. **Введение**

На данный момент за всю историю кинематографа насчитывается порядка 2,6 миллионов кинолент, включая отдельные эпизоды сериалов, мультфильмы и короткометражки. И за последние десятилетия их количество продолжает расти. Интерес к данной сфере развлечения среди людей при этом не уменьшается. Соответственно люди продолжают смотреть фильмы притом в большом количестве. И чтобы сохранить всю эту информацию о том, что уже было просмотрено, что планируется и т.д. необходимо иметь приложение способное осуществлять эту задачу. Если брать в сравнение известные платформы, то, например, «*Кинопоиск»* предоставляет пользователю только 2 категории («буду смотреть» и «просмотрен»), что на мой взгляд реализовано куда менее удобно, чем планируется быть сделано мной.

**Объект:** язык программирования Java.

**Предмет:** прототип рабочего приложения по заданным критериям.

**Цель:** разработка программного решения для поиска и хранения информации о любимых фильмах (сериалах).

**Задачи:**

1. Изучение языка программирования Java;
2. Изучение библиотеки JavaFX и графического редактора SceneBuilder;
3. Изучение работы неофициального API сайта «*Кинопоиск*»;
4. Разработка графического дизайна приложения;
5. Проектирование программной архитектуры

**Практическая значимость проекта:** Мой проект способен создать любителям фильмов и сериалов возможность удобно сортировать все те фильмы и сериалы, которые они посмотрели, планируют посмотреть и т.д. Также код находится в открытом доступе, что позволяет энтузиастам использовать части либо весь проект, как базу для создания более усовершенствованного приложения или сайта о фильмах.

**Риски:** Из-за несовершенства навыков и опыта в создании приложения, приложение может быть не совершенным на рынке приложений, что способно привести к угасанию интереса к нему. А также есть риски вмешательства в работу кода, в случае изменения его состава иными лицами, что может привести к получению вирусов, однако учитывая, что код не подразумевает взаимодействие с другими пользователями, риск снижается.

**Положительные эффекты:** Максимальный положительный эффект от работы кода могут ощутить те, кто тесно связан с фильмами и сериалами и кто входит в группу людей для которых создавался код; куски кода можно использовать при решении иных задач, путем аналогичных действий.

**II. Основная часть**

**Графика и интерфейс**

1. **MovieListViewItem** – графический элемент, необходимый для создания мини-карты фильма в поле нахождения всех подходящих вариантов в поиске.

*Элементы:*

* **HBox** (*Style*: -fx-background-color: #0d0505)

основное поле

* **ImageView** (*fxid*: avatarThumbnail)

объект способный принимать изображение

* **VBox** (*Style:* -fx-background-color: #0d0505)

поле для добавления строк в столбец

* **Label** (*Style*: -fx-background-color: #595151, fxid:nameTitle)

строка для названия фильма

* **Label** (*Style*: -fx-background-color: #595151, fxid:Description)

строка для типа

1. **MovieCardViewer** –графический элемент, необходимый для получения списка найденных фильмов и для хранения карты с информацией о фильме.

*Элементы:*

* **HBox**

поле для хранения других объектов

* **ListView** (*Style*: -fx-background-color: #3b0d0d, *fxid*:movieListView)

поле, в которое будут передаваться все найденные MovieListViewItem

* **HBox** (*Style*: -fx-background-color: #362b2b, *fxid*:movieCardContainer)

поле, которое будет отображать карту с фильмом

1. **cardOfFilms** – карта с главной информацией о фильме.

*Элементы:*

* **HBox** – основное поле для других объектов
* **BorderPane** (*Style*: -fx-background-color: #0d0505)

поле, в котором будем размещать объекты с определенной ориентацией и привязкой к границе

* **ImageView** (*fxid*:Poster)

поле для изображения постера к фильму

* **TilePane**

поле, в которое будут загружаться строки

* **Label** (*Style*: -fx-background-color: #595151, *fxid*:nameTitle)

строка названия

* **Label** (*Style*: -fx-background-color: #595151, *fxid*:years)

строка года

* **Label** (*Style*: -fx-background-color: #595151, *fxid*:Genres)

строка жанров

* **Label** (*Style*: -fx-background-color: #595151, *fxid*:Description)

строка оригинального названия

* **GridPane**

таблица 2 на 3, в которую будут помещены кнопки

* **Button** (*Style*: -fx-background-color: #540808, *fxid*:btnWatched)

кнопка просмотрено

* **Button** (*Style*: -fx-background-color: #540808, *fxid*: btnFavourites)

кнопка избранное

* **Button** (*Style*: -fx-background-color: #540808, *fxid*:btnPlanned)

кнопка в планах

* **Button** (*Style*: -fx-background-color: #540808,*fxid*:btnPostponed)

кнопка отложено

* **Button** (*Style*: -fx-background-color: #540808, *fxid*:btnAbandoned)

кнопка брошено

1. **project** – графическая часть, являющаяся основой для всего проекта и принимающая в себя графические элементы 1 – 3.

*Элементы:*

* **BorderPane** (*Style*: -fx-background-color: #0d0505)

поле, в котором будем размещать объекты с определенной ориентацией и привязкой к границе

* **TabPane** (*Style*: -fx-background-color: #0d0505)

встроенный объект, который позволит создать ячейки

* **Tab** (*Style*: -fx-background-color: #8a8383)

переход на главную страницу

* **AnchorPane** (*Style*: -fx-background-color: #0d0505)

основное поле каждой ячейки

* **Button** (*Style*: -fx-background-color: #595151, *fxid*:searchBtn)

кнопка поиска фильма

* **TextField** (*Style*: -fx-background-color: #595151, *fxid*:searchTextField)

строка для ввода названия фильма

* **VBox**

Остальные пять ячеек этого элемента строятся на том же принципе, в связи с чем их описание не будет задействовано.

Для более подробного изучения графических элементов вы можете открыть исходный код и по пути:

**project\_now-master\src\main\resources\com\example\project**

найти все графические элементы.

**Основные классы, предназначения, свойства и методы кода**

|  |  |  |
| --- | --- | --- |
| **Название класса** | **Предназначение класса** | **Свойства и методы** |
| **MovieApplication** | создание окна для приложения | Start - запуск программы  MovieService - подключение сервера |
| **Семейство классов Controller** | переход от сцены к сцене с использованием кнопок и др. | Графические элементы  initialize - обработка касаний кнопок  changed - отображение сцены при нажатии на раздел |
| **MovieService** | передача информации из карточки | getMovies (для каждого отдела) - возвращение листа кино каждого отдела  FindMovies - получение List содержащий фильмы, по слову  saveMovie - сохранение фильма в репозитории |
| **MovieRepository** | хранение информации из карточек | findBy…IsTrue( для каждого раздела) - получение информации о фильмах сохраненных в данном разделе |
| **Movie** | содержание всех необходимых свойств и методов о фильме | флажки для разделов  переменные данных  is…( для каждого раздела) - получение статуса флажка  set… - установление статуса флажка  set..( для информации) - установление информации о фильме  get… - получение информации о фильме |
| **MovieCard** | Загрузка информации в карточку и обработка касаний кнопок (В планах,избранное и т.д.) | граф. элементы  MovieCard - прогрузка карточки в сцене  SetMovie - установление информации о фильме  initialize - обработка касаний кнопок |
| **MovieCardViewer** | Загрузка специального граф. элемента (ListView + Hbox) и заполнение его необходимой информации | граф. элементы  MovieCardViewer - прогрузка в сцене  SetMovies - добавление фильмов  movieClicked - обработка касания на карточку  initialize - получение всей необходимой информации |
| **MovieListViewItem** | Заполнение карточки с краткой информацией о фильме | граф. элементы  MovieListViewItem - прогрузка элемента в сцене  SetMovie - загрузка информации в карточку  initialize - оптимизация загрузки изображения |

**Вывод:** Мне удалось добиться результата, который был заложен в целях и задачах моего проекта. Код работает исправно и выполняет свои задачи. Код выложен в открытый доступ. Каждый кто захочет изучить код подробнее может сделать это перейдя по ссылке:

<https://github.com/AntonVorob/project_now>

К текстовой работе прилагается видео файл с демонстрацией работы кода.

**III. Календарный план-график**

|  |  |  |  |
| --- | --- | --- | --- |
| **Этап работы над проектом** | **Содержание деятельности** | **Результаты работы и выводы** | **Сроки выполнения** |
| Организационно-подготовительный этап | 1. Выбор направления проекта, учителя – куратора, формулирование темы продукта. 2. Формулирование целей и задач проекта, его объекта и предмета. 3. Обоснование актуальности проекта. | Направление и куратор выбраны, написана вводная часть. | Сентябрь-октябрь 2023 |
| Деятельностный этап | 1. Изучение языка программирования 2. Знакомство со средой разработки 3. Решение задач с целью более подробного изучения языка 4. Изучение ООП 5. Изучение графического редактора 6. Создание приложения | Изучено все необходимое для создания рабочего приложения. Создание версии приложения, максимально приближенной к первоначальной задумке. | Сентябрь 2021- октябрь 2023 |
| Оформление проекта | 1. Написать введение, содержание, календарный план-таблицу. 2. Написать основную информацию о коде. 3. Записать ход работы над кодом. 4. Сделать ссылки на источники. | Письменная часть полностью написана. | Октябрь 2023-декабрь 2023 |
| Презентация проекта | 1. Подготовка к презентации перед классом 2. Презентация проекта | Презентация для защиты перед классом подготовлена | Февраль 2024 |

**IV. Список информационных источников**

1. <https://ru.wikipedia.org/wiki/Java> − Википедия. Java
2. [https://ru.wikipedia.org/wiki/Объектно-ориентированное\_программирование](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) − Википедия. ООП
3. [https://ru.wikipedia.org/wiki/Класс\_(программирование)](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) −

Википедия. Класс(программирование)

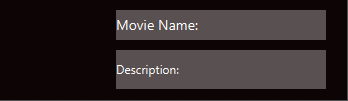
1. [https://ru.wikipedia.org/wiki/Объект\_(программирование)](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) −

Википедия. Объект (программирование)

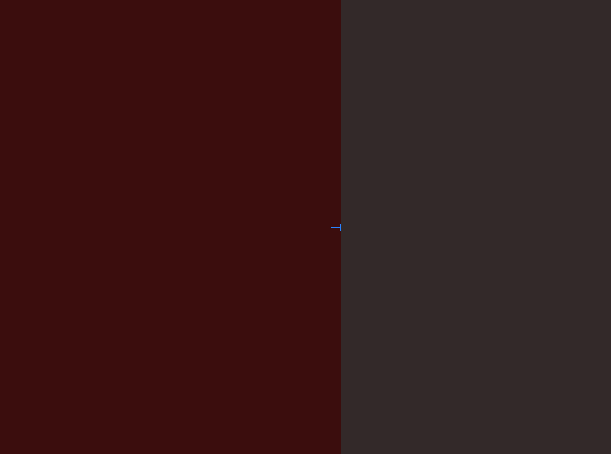
1. [https://ru.wikipedia.org/wiki/Поле\_класса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) − Википедия. Поле класса
2. [https://ru.wikipedia.org/wiki/Репозиторий](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9) − Википедия. Репозиторий
3. <https://ru.wikipedia.org/wiki/Spring_Framework> − Википедия. Spring
4. <https://ru.wikipedia.org/wiki/Фреймворк> − Википедия. Фреймворк
5. <https://ru.wikipedia.org/wiki/API> − Википедия. API
6. <https://ru.wikipedia.org/wiki/JavaFX> − Википедия. JavaFX
7. <https://habr.com/ru/articles/> − Хабр
8. <https://github.com/> − GitHub
9. <https://stackoverflow.com/> − Stack Overflow
10. Курс: «Проектное программирование Java», АЦТ, 2021-2022. Руководитель: Семенов Георгий Витальевич

**Приложение**

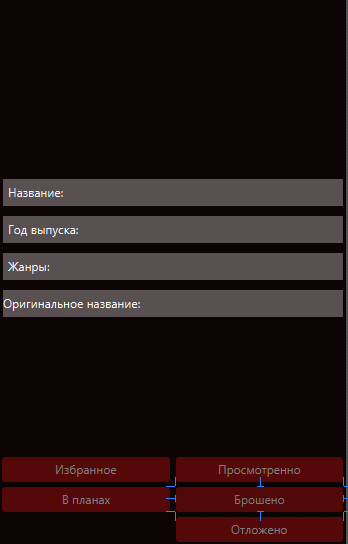
1. **MovieListViewItem**



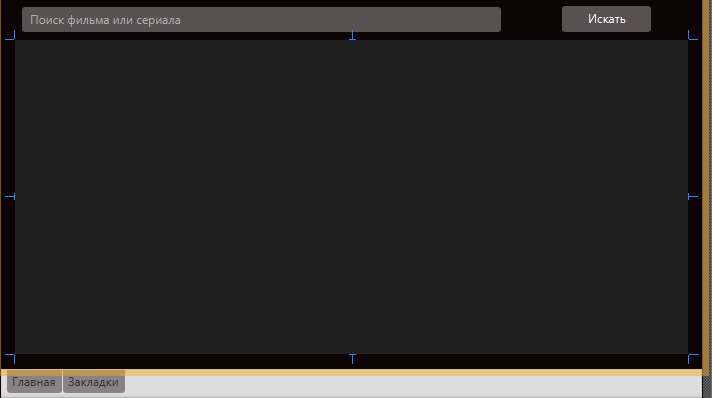
1. **MovieCardViewer**

****

1. **cardOfFilms**

****

1. **project**

****