**Кинотека**

**Автор:** [**Савин Олег**](https://vk.com/lessoleg) **(**[**keyfawn**](https://github.com/keyfawn)**)**

**Кинотека** — это программа, в которой можно хранить свои фильмы, мультфильмы. Есть возможность добавлять кино при нажатии Ctrl + A, изменить выбранный фильм с помощью Ctrl + E и удалить выделенный мультфильм - Ctrl + D. Имеется возможность отсортировать список названий фильмов. Предназначен для users, не имеющих доступа к интернету.

Основные классы – это «Window» (главное окно), «WindowCinema» (окно фильма), «AddCinema» (окно, добавляющее фильм), «EditGenre» (окно, редактирующее список жанров), «WindowAbout» (окно, содержащее описание программы). Для удобства быстрого внесения изменений в исходный код и наилучшей читабельности классы были разбиты на отдельные файлы.

Функционал, относящийся к обслуживанию проекта, для уменьшения копирования кода вынесен в отдельный утилитарный класс Utils. В нем в основном функционал работы с БД.

**Используемые технологии**

При разработке проекта использовались следующие библиотеки:

- PyQt5 – В моём проекте используется библиотека PyQt5. С помощью её можно создавать полноценные программы. Устанавливается она с помощью простой команды[[1]](#footnote-1) в Terminal.

- Shutil – для копирования файлов баннеров и видеозаписей в отдельный каталог ресурсов.

- os – для работы с видео и графикой в каталогах проекта. Получение полного пути, удаление неиспользуемых ресурсов и т.д.

- subprocess – для запуска видео прямо из проекта.

- sqlite3 – для работы с базой данных проекта.

Кроме того, было применено несколько методологий программирования:

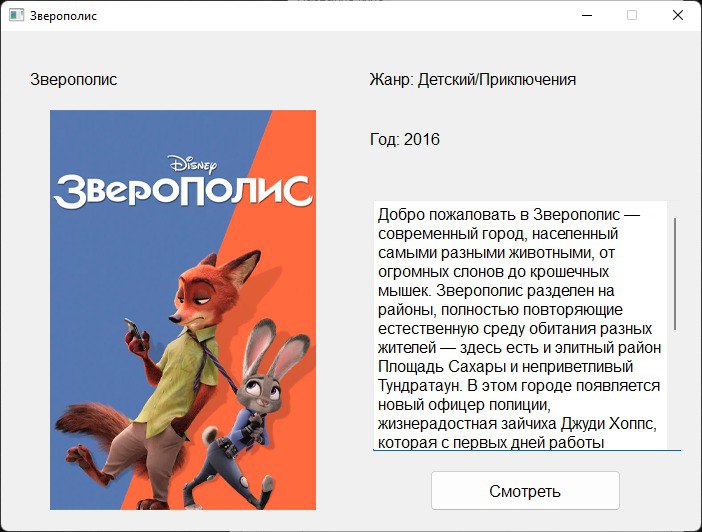
1. Весь проект в целом разработан в соответствии с ООП методологией.
2. Как уже говорилось ранее, для удобства, каждая отдельная страничка и ее функционал вынесены в отдельные классы. UI-компоненты так же существуют отдельно от функционала. Этот способ проектирования использует паттерн Page Object.

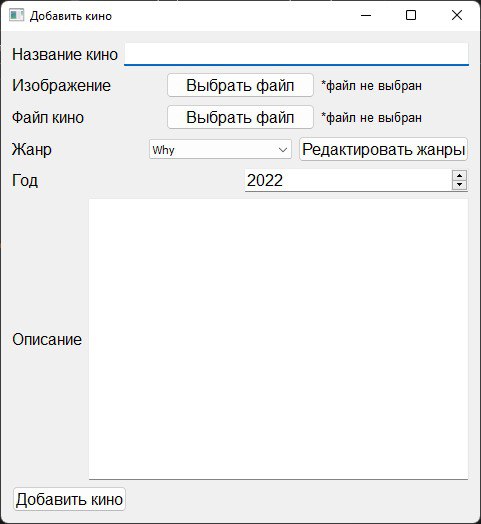
**Листинг интересного кода**

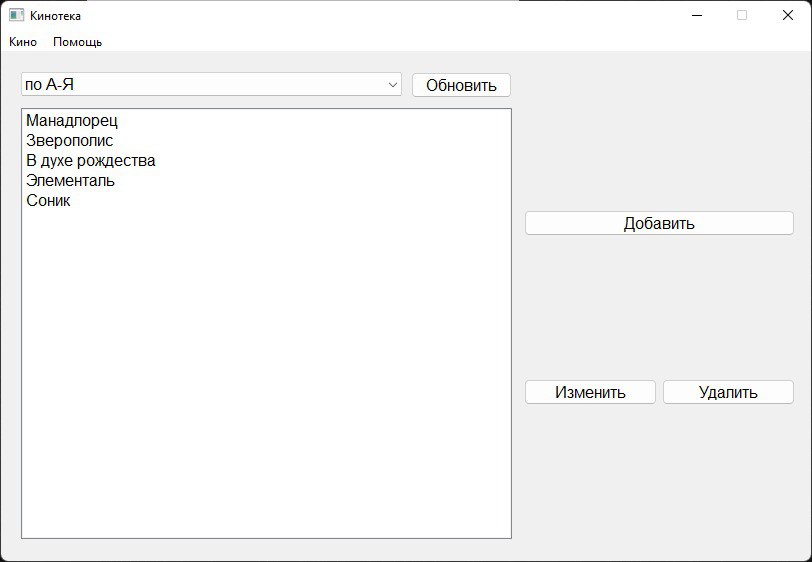
def addFileCinema(self):  
 if not self.nameCinema.text():  
 QMessageBox.critical(self**,** "Ошибка"**,** "Вы не задали название")  
 return  
 if self.db.check\_film\_name(self.nameCinema.text()):  
 QMessageBox.critical(self**,** "Ошибка"**,** "Фильм с таким названием уже есть")  
 return  
 path = QFileDialog.getOpenFileName(self**,** 'Выбрать кино'**,** ''**,** 'Видео (\*MP4);; Видео (\*WebM);; Видео (\*MOV);; Видео (\*AVI);; Видео (\*FLV);; Видео (\*WMV);;Все файлы (\*)')[0]  
 if path:  
 self.textShortFileCinema.setText('\*файл выбран')  
 self.pathCinema = f"resourses/video/vid\_{self.nameCinema.text()}\_.{Utils.get\_file\_ex(path)}"  
 shutil.copyfile(path**,** self.pathCinema)

Смысл этой функции заключается в том, что она копирует видеофайл, выбранный пользователем, в собственную директорию «resourses/video» под названием «vid\_{self.nameCinema.text()}\_.{Utils.get\_file\_ex(path)» . Если в поле для названия фильма будет пустым или название будет совпадать с существующим уже фильмом, то будет вызвана критическая ошибка.

**Скриншоты**







1. *pip install PyQt5* [↑](#footnote-ref-1)