**Надоевшие комедии**

Напишите функцию get\_result(name), которая принимает на вход имя файла базы данных, по структуре идентичной «films.db», и удаляет все фильмы в жанре комедии из БД.

**У любого фильма есть длина**

Напишите функцию **get\_result(name)**, которая принимает на вход имя файла базы данных, по структуре идентичной **films.db**, и обновляет информацию о фильмах: поле с длительностью фильма, если оно изначально пустое (пустая строка), должно стать равным 42 минутам.

**Длинные фантастические фильмы**

Напишите функцию get\_result(name), которая принимает на вход имя файла базы данных, по структуре идентичной «films.db», и увеличивает длительность фантастических фильмов вдвое.

**От Я до А**

Напишите функцию get\_result(name), которая принимает на вход имя файла базы данных, по структуре идентичной «films.db», и удаляет все фильмы, название которых начинается на букву „Я“ и заканчивается на букву „а“.

# Генерация фильмов

Модифицируйте программу, рассмотренную в материалах урока, таким образом, чтобы по заданному индексу после просмотра элемент удалялся, и создавался новый по следующему правилу:

1. Название нового элемента — это перевёрнутое старое название
2. Год выпуска — соответствующий год следующего тысячелетия
3. Жанр не меняется
4. Продолжительность увеличивается в 2 раза

# Фильмотека 1.0

Напишите программу с графическим пользовательским интерфейсом на PyQt, которая позволяет просматривать и добавлять данные в таблицу Films базы данных films\_db.sqlite.

Информацию о фильмах отобразите в табличном виде, а для жанра выведите его текстовое название (подумайте, как это можно сделать с помощью SQL запроса, [вот это](https://ru.wikipedia.org/wiki/Join_(SQL)) может помочь). Добавление данных должно быть реализовано путем заполнения данных в отдельные виджеты для каждого поля таблицы films, наиболее подходящие для этого. Виджеты для добавления записи должны располагаться на отдельной форме приложения. После добавления записи необходимо обновить таблицу с отображением данных.

Предусмотрите сообщения для ошибочного пользовательского ввода, например, отрицательной длины, года в будущем и т. д.

