**Государственное бюджетное общеобразовательное учреждение города Москвы «Школа № 192»**

**Телеграм-бот   
“ТОТ САМЫЙ ПЛЕЙЛИСТ”.**

Ученица 10в класса “Школы № 192”

Царёва Фиона Сергеевна

руководитель: учитель информатики

Москва “Школа № 192”

Бонвеч Елена Алексеевна

**Москва, 2024**

# Оглавление

[**Оглавление 2**](#_7zsln4l7nvd4)

[**Введение 3**](#_yk7qymakc0td)

[**Цель и задачи 5**](#_jfyhdbkhbwwm)

[**Методика выполнения работы 6**](#_arj8r96xlxgw)

[Сбор и изучение информации 6](#_gp2heg3pyx4)

[➔ Существуют ли проекты-аналоги? 6](#_hu0kwfcgr9p6)

[➔ Проводились ли какие-либо исследования о потреблении музыкального контента пользователями? 6](#_wovr02pagjqx)

[Выбор инструментов 7](#_tz0jzfurfyhf)

[➔ Какую среду разработки выбрать? 7](#_7054vgqofqts)

[➔ На каком языке программирования писать код? 7](#_j7uamvm3uzpu)

[➔ Что использовать для создания базы данных? 7](#_xmnfi647am09)

[➔ Какие дополнительные библиотеки нужно использовать при работе над проектом и необходимо изучить? 8](#_gskisaq1byx1)

[Создание базы данных 8](#_3z4ykcwje838)

[Написание кода 9](#_yxfnzrfgyauo)

[Тестирование и доработка 9](#_r835x3gbp3va)

[**Перспективы развития проекта 10**](#_pkoolnvdtj1e)

[**Описание итогового результата 11**](#_kmxs1c8dw00k)

[**Список использованной литературы 12**](#_gfxps8pnzn13)

# Введение

Люди, которые часто слушают музыку во время работы, отдыха или поездки, в один момент оказываются в небольшом кризисе по причине того, что прослушивание любимой музыки больше не приносит такого удовольствия как раньше, а порою даже раздражает. Очевидно, что нужно найти другие треки, в других жанрах или от других исполнителей, однако ни они, ни алгоритмы, собирающие для них музыкальные сборники (плейлисты) на основе их нынешних предпочтений, не знают что именно заинтересует пользователя. Можно попытаться изучить все существующие жанры, просмотреть весь Интернет и все музыкальные площадки в поиске того самого “спокойного плейлиста для работы”, но это займёт слишком много времени…

Эта на первый взгляд незначительная проблема на самом деле актуальна, так как в настоящее время огромное количество людей активно пользуется сервисами для прослушивания музыки, но имеют неоднозначное мнение об алгоритмах, подбирающих рекомендации для пользователей. *“Тогда как 37% опрошенных ими не пользуются или почти не пользуются, <...>, 31% пользуются, но склонны думать, что набор подсказок ограничен”* – пишут в результатах своего исследования музыкального потребления россиян студенты НИУ ВШЭ Иван Напреенко и Артём Рондарев.[1]

Для решения этой проблемы нужно создать инструмент, помогающий активным слушателем находить новую музыку, новые плейлисты, которые по своей тематике идеально подходят пользователю. И чтобы пользоваться им было так же просто и удобно, как и музыкальными платформами, и не нужно было бы устанавливать какие-либо дополнительные файлы, инструмент будет встроен в чат-бот приложения Телеграм. Телеграм – второй в мире мессенджер по популярности после WhatsApp [2], который легко позволяет создать внутри себя бота с любым алгоритмом работы, поэтому идея создать наш инструмент внутри именно этого мессенджера логична и справедлива.

# Цель и задачи

Главной целью моей работы над проектом является создание инструмента, который будет обладать большой базой данных с разнообразными плейлистами и который сможет правильно понять рекомендации пользователя о нужном ему плейлисте.

Весь мой процесс создания чат-бота состоит из следующих этапов:

* Сбор и изучение информации
  + Существуют ли проекты-аналоги?
  + Проводились ли какие-либо исследования о потреблении музыкального контента пользователями?
* Выбор инструментов для создания проекта
  + Какую среду разработки выбрать?
  + На каком языке программирования писать код?
  + Что использовать для создания базы данных?
  + Какие дополнительные библиотеки необходимо изучить для работы над проектом?
* Создание базы данных
* Написание кода
* Тестирование и доработка

# Методика выполнения работы

## Сбор и изучение информации

### Существуют ли проекты-аналоги?

Да, в приложении Telegram был найден чат-бот, также реализующего поиск плейлистов по рекомендациям пользователя [11]. При запуске бот предлагает пользователю выбрать сервис, на котором он ищет плейлист, а затем жанр. Бот простой и понятный и в использовании, однако все же имеет два небольших недостатка. Во-первых, большая часть плейлистов, которые может предложить пользователю чат-бот, расположена на заблокированном в РФ сервисе Spotify, а во-вторых, чат-бот имеет ограниченный список жанров и тем.

Поэтому моя задача – расширить список жанров и тем плейлистов, а также сделать больший выбор сервисов для прослушивания

### Проводились ли какие-либо исследования о потреблении музыкального контента пользователями?

Да, одно из них уже цитировалось выше. Помимо него, существуют и другие исследования музыкального потребления в настоящее время, такие как:

* *Д. В. Апанович. Курсовая работа “Музыкальное потребление современной молодежи Кубани”*.[2] Студент из Краснодара изучает общую характеристику и особенности музыкального потребления среди молодежи в настоящее время, а также влияние прослушиваемой музыки на формирование нравственных ценностей человека.
* *М. Г. Снежинская. Автореферат “Музыкальная индустрия как социокультурный феномен”.* [9]В данном исследовании анализируется характеристики музыки как социально-культурного феномена и её специфики в конце XX – начале XXI века.

## Выбор инструментов

### Какую среду разработки выбрать?

Весь код я писала в онлайн-блокноте Deepnote. Несмотря на то, что платформа рассчитана на разработчиков в области Big Data и и создана в основном для совместной работы в реальном времени, она не требовала тщательной настройки виртуального окружения, что позволило быстро приступить к работе.

### На каком языке программирования писать код?

Весь алгоритм работы бота написан на Python 3. Это современный и универсальный язык программирования, при изучении которого не возникает трудностей. Для реализации проекта и работы с библиотеками было достаточно знаний основ языка.

### Что использовать для создания базы данных?

Я рассматривала три варианта как мне стоит создавать таблицу с данными о плейлистах и работать с ними:

* Библиотека Pandas для создания и обработки больших данных
* Библиотека PySQL, которая позволяет работать с SQL-файлами в Python
* Excel-таблица, с которой можно работать на Python с помощью библиотеки openpyxl

В конце концов, я решила работать с Excel-таблицами, так как за короткий срок этот метод работы с данными был легче всего освоить

**Примечание:** В общей сложности работа над проектом длилась одну-две недели до дня окончания подачи заявок на конференцию

### Какие дополнительные библиотеки нужно использовать при работе над проектом и необходимо изучить?

Общий список библиотек, которые мне пришлось дополнительно установить и изучить для работы над проектом

* **PyTelegramBotAPI.[4][8]** Эта библиотека создана для описания всех функций для Телеграм-бота на языке Python
* **openpyxl.[6][10]** Позволяет взаимодействовать с Excel-таблицей: добавлять, менять и удалять данные внутри неё

## Создание базы данных

Excel-таблица с данными о плейлистах в общем виде выглядит следующим образом, см Рис. 1.

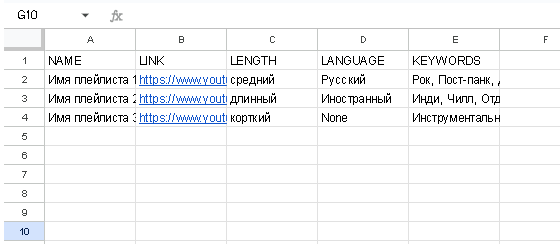


Рис. 1: Шаблон таблицы с данными о плейлистах

Каждый столбец относится к одной из характеристик плейлистов, каждая строка относится к одному плейлисту, в каждом листе содержатся плейлисты из одного музыкального сервиса.

## Написание кода

Практическая часть создания проекта, в ходе которой весь алгоритм работы бота прописывается в виде кода. Этот этап я считаю самым важным, так как здесь создается основная часть всего проекта – его главные функции и внешний вид.[5]

## Тестирование и доработка

Дополнительный этап, цель которого – выявить все недочеты, упущенные во время практической части, и исправить их прежде чем их обнаружит пользователь.

# Перспективы развития проекта

В дальнейшем развитии проекта планируется следующие нововведения:

* Возможность для пользователей сохранить понравившиеся подборки внутри бота и всегда иметь к ним доступ; добавлять собственные музыкальные подборки в базу данных бота; скачать композиции из сборника на устройство.
* Расширение базы данных, добавление новых плейлистов из других сервисов для прослушивания музыки

# Описание итогового результата

Данный проект представляет собой Телеграм-бота с собственной базой данных и создан для того, чтобы упростить поиск музыкальных плейлистов по настроению и тематике. Ниже описан принцип работы бота.

После запуска (ввода команды “/start”) бот приветствует пользователя и предлагает ему либо пройти тест (ввести команду “/test”), либо найти плейлист по ключевым словам (ввести команду “/search”).

Если пользователь решил пройти тест, бот начинает задавать пользователю вопросы. Список с вариантами ответа представляет собой inline-клавиатуру под сообщением с вопросом. Помимо ответов на вопросы, в inline-клавиатуре есть кнопки со стрелками для перехода между вариантами ответа (если вариантов ответа слишком много и на одной “странице” не умещаются), кнопка перехода на следующий и предыдущий вопросы, а также кнопка сброса ответа и кнопка завершения теста.

Список вопросов, которые бот задает пользователю:

* На какой музыкальной платформе пользователь ищет плейлист?
* Какой максимальной длины должен быть плейлист? (имеет ли значение для пользователя длина плейлиста?)
* Имеет ли значение для пользователя, на каком языке исполняются музыкальные композиции, или, может, композиции в плейлисте должны быть вовсе без слов?

До перехода к последнему вопросу пользователь может сбросить ответ на любой из трех вопросов и ввести ответ на них ещё раз. После перехода к последнему вопросу пользователь не сможет изменить ответы.

Последним вопросом теста является выбор нескольких ключевых слов. Пользователю показывают большой список из названий жанров и тематик. Пользователю нужно выбрать несколько ключевых слов, по которым бот и будет искать плейлист. В списке, очевидно, не все существующие в базе данных ключевые слова, а только те, которые есть в плейлистах, удовлетворяющих ответам пользователя на предыдущие три вопроса. После каждого выбора список сокращается, чтобы не получилась такая комбинация ключевых слов, по которой не удастся найти плейлист. Пока пользователь не завершил тест, он может добавить в свой ответ ключевые слова до тех пор, пока не останется больше вариантов в списке, убрать из ответа последний вопрос или сбросить весь ответ.

Когда пользователь наконец-то завершает тест, бот, проанализировав его ответы с данными в таблице, присылает сообщение со списком плейлистов, которые, вероятнее всего, подойдут пользователю. Каждый номер списка состоит из названия плейлиста и ссылки на него на музыкальной платформе. Пользователь может перейти по ссылке и прослушать плейлист.

Если пользователь ввел команду “/search”, то он может найти плейлист в базе данных бота по ключевым словам в названии. Прежде чем он сможет ввести поисковый запрос, пользователь должен выбрать музыкальный сервис, на котором ищет плейлист (под сообщением – inline-клавиатура с вариантами ответа). Пользователь может пропустить этот вопрос, если это для него не имеет значения. Теперь пользователь может ввести поисковый запрос. Результаты поиска выводится аналогично результатам прохождения теста: сообщение со списком, в каждом пункте списка название плейлиста и ссылка на него на музыкальной платформе.

# Список использованной литературы

1. Практики музыкального потребления россиян. Основные особенности и тренды / Иван Напреенко, Артем Рондарев // Институт исследований культуры НИУ ВШЭ — 2022 // URL: <https://publications.hse.ru/pubs/share/direct/560621998.pdf> (дата обращения: 18.02.2024)
2. Курсовая работа “Музыкальное потребление соверемнной молодежи Кубани” / Д. В. Апанович // ФГБОУ ВО “КубГУ” – 2018 // URL: <https://www.kubsu.ru/sites/default/files/users/26406/portfolio/kursach_daniil.pdf> (дата обращения: 18.02.2024)
3. Telegram – Wikipedia // URL: <https://ru.wikipedia.org/wiki/Telegram> (дата обращения: 19.02.2024)
4. PyTelegramBotAPI на примере проекта сбора обратной связи #3. Клавиатура // URL: <https://habr.com/ru/articles/775630/> (дата обращения: 16.02.2024)
5. Как создать телеграм-бота на Python: инструкция // URL: <https://timeweb.cloud/tutorials/python/kak-sozdat-telegram-bota-na-python> (дата обращения: 16.02.2024)
6. Документация по библиотеке openpyxl версии 3.1.2 // URL: <https://openpyxl.readthedocs.io/en/stable/> (дата обращения: 16.02.2024)
7. Документация по языку программирования Python версии 3.12.2 // URL: <https://docs.python.org/3/index.html> (дата обращения: 15.02.2024).
8. Документация по библиотеке pyTelegramBotAPI версии 4.15.4 // URL: <https://pytba.readthedocs.io/en/latest/index.html> (дата обращения: 15.02.2024).
9. Музыкальная индустрия как социокультурный феномен / М.Г. Снежинская // URL: <http://www2.rsuh.ru/binary/2639735_10.1569831551.48089.pdf> (дата обращения: 18.02.2024)
10. Работа с файлами XLXS при помощи модуля openpyxl // URL: <https://docs-python.ru/packages/modul-openpyxl/> (дата обращения: 18.02.2024)
11. Телеграм-бот по поиску музыки “Плейлисты | Бот из TikTok” // URL: <https://t.me/RealMusicRobot> (дата обращения: 17.02.2024)