ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**ЗАДАНИЕ**

**для самостоятельной работы по индивидуальному плану по курсу  
“ Проектный семинар “Python в науке о данных”**

Студент(ы) группы БИВ224:

* Воробьев Антон Вадимович, avvorobev\_5@edu.hse.ru, +79652039942;
* Саблин Арсений Михайлович, amsablin@edu.hse.ru, +79170284688.

Тема работы:

**Приложение для анализа музыкальных композиций из Spotify**

Руководитель:

Полякова Марина Васильевна

Оглавление

[Введение 3](#_Toc134801037)

[Основания для разработки 4](#_Toc134801038)

[Назначение разработки 4](#_Toc134801039)

[Функциональное назначение 4](#_Toc134801040)

[Эксплуатационное назначение 4](#_Toc134801041)

[Требование к программе 5](#_Toc134801042)

[Требования к функциональным характеристикам 5](#_Toc134801043)

[Требования к Frontend 5](#_Toc134801044)

[Требования к Backend 6](#_Toc134801045)

[Требования к надежности 6](#_Toc134801046)

[Условия эксплуатации 6](#_Toc134801047)

[Требования к составу и параметрам технических средств 6](#_Toc134801048)

[Требования к информационной и программной совместимости 6](#_Toc134801049)

[Требование к программной документации 7](#_Toc134801050)

[Технико-экономические показатели 7](#_Toc134801051)

[Стадии и этапы разработки 8](#_Toc134801052)

[Порядок контроля и приемки 8](#_Toc134801053)

[Критерии оценивания 8](#_Toc134801054)

# Введение

Приложение для анализа музыкальных композиций из Spotify — это приложение, разработанное для анализа музыкальных треков на основе параметров, предоставляемых платформой Spotify. Оно позволяет анализировать такие параметры, как:

* энергия (energy)
* популярность (popularity)
* танцевальность (danceability)
* громкость (loudness)
* длительность (duration)
* аккомпанемент (acousticness)
* инструментальность (instrumentalness)
* живость (liveness)
* настроение (valence)
* темп (tempo)

Приложение работает в режиме оффлайн и обеспечивает точный анализ музыкальных композиций, основанный на надежных и актуальных данных Spotify датасета. С помощью этого приложения пользователи могут получить ценные знания о своей музыкальной библиотеке, сформировать плейлисты и открыть для себя новые музыкальные жанры и артистов. Все это делает приложение для анализа музыкальных композиций из Spotify незаменимым инструментом для любителей музыки и профессионалов в индустрии развлечений.

В данном техническом задании определены требования к функциональности приложения, его интерфейсу, безопасности и сопровождению.

# Основания для разработки

Данный проект выполняется в рамках дисциплины “ Проектный семинар “Python в науке о данных” и является образовательным. По итогу курса у студентов появляются компетенции о анализе данных. Общая работа выполняется на языке программирования Python. Кроме того, курс включает в себя знакомство со следующими библиотеками:

* Pandas – для работы с таблицами
* Matplotlib – для построения математических диаграмм и графиков
* NumPy – для работы с линейной алгеброй
* Tkinter, PyQt5 (QT Framework) - для создания GUI

В процессе работы над проектом должны быть получены практические навыки со всеми вышеперечисленными библиотеками.

# Назначение разработки

## Функциональное назначение

Приложение обеспечивает точный анализ музыкальных композиций, основанный на надежных и актуальных данных Spotify, и позволяет пользователям получить ценные знания о своей музыкальной библиотеке, сформировать плейлисты и открыть для себя новые музыкальные жанры и артистов. Функциональное назначение приложения заключается в предоставлении пользователю удобного и интуитивно понятного интерфейса для анализа музыкальными данными на платформе Spotify.

## Эксплуатационное назначение

Основная цель программы - обеспечение пользователей информацией о своих музыкальных предпочтениях и помощь им в нахождении новой музыки, которая может заинтересовать их на основе предыдущих предпочтений. Кроме того, приложение может использоваться для анализа и сравнения различных характеристик композиций, таких как энергия, популярность и т. д.

# Требование к программе

## Требования к функциональным характеристикам

Программа состоит из двух основных компонент: пользовательского интерфейса (Frontend) и математических функций, обеспечивающих построение графиков (Backend). Функции, которое предоставляет приложение:

* Вывод записей (треков) из датасета, их параметров. Построение полярных графиков их параметров;
* Анализ всех исполнителей из датасета, вывод их мод и медиан (описательный анализ). Построение графиков изменения выбранного пользователем параметра по годам;
* Графический анализ данных:
  + Построение графика зависимости одного параметра от другого;
  + Построение гистограммы параметра по количеству (должна быть настройка количества столбиков);
  + Построение зависимости популярности от года (эволюция жанра);
  + Построение корреляции;
  + Построение полярного графика сравнения параметров 100 самых популярных и средних всего датасета.
* Сортировка с ограничением по годам (топ треков по заданному пользователем году). Должна быть возможность инвертировать сортировку;
* Описание всех параметров.

### Требования к Frontend

Интерфейс должен быть интуитивно понятен. Основные функции приложения должны быть разбиты по вкладкам с помощью TabWidget любой реализации (Tkinter/ Qt). Для каждого аргумента функции должно быть текстовое поле или ComboBox. Блокировка основного потока UI возможна, однако не должна приводить к ANR (Application Not Responding).

Виджеты должны быть масштабируемы, то есть разбиты по менеджерам расстановки (Layout). При изменении масштаба основного окна элементы должны подстраиваться под него, а не скрываться с экрана.

UI (user interface – пользовательский интерфейс) должен быть кроссплатформенным, для того, чтобы выполнение программы было возможно на всех нижеперечисленных операционных системах:

* Windows (начиная с Windows XP)
* Mac OS
* Linux (или любая другая UNIX-подобная с окружением рабочего стола)

### Требования к Backend

Функции должны принимать аргументы и выводить графики. У каждого графика должно быть названии. Функции должны лежать в отдельном модуле и вызываться из UI модуля.

## Требования к надежности

Недопустимо, чтобы во время работы программа не завершалась с критической ошибкой. Время, которое приложение не отвечает на запросы пользователя не должно превышать времени ошибки ANR (Application Not Responding). Обработка данных возможна в главном потоке, но не должна приводить обозначенной выше ошибке.

## Условия эксплуатации

## Требования к составу и параметрам технических средств

Приложение должно исполняться на компьютере средней мощности с 8Gb оперативной памяти и 2Gb видеопамяти не очень мощной видеокарты.

## Требования к информационной и программной совместимости

Приложение должно запускаться из командной строки запуском на счет основного модуля или командного файла. Установка приложения, в частности создание при необходимости виртуальной среды, должна быть полностью автоматизирована, т.е. осуществляться запуском на счет модуля Python или командного файла. Установка приложения и его эксплуатация не должны требовать выполнения системных операций, связанных с изменениями операционной системы, актуального дистрибутива Anaconda и прочих элементов вычислительной среды пользователя. В частности, развертывания ПО Docker и аналогичных инструментов.

Для работы приложения должна быть создана виртуальная среда исключительно средствами менеджера пакетов conda из актуального дистрибутива Anaconda. Таким образом, приложение должно работать в собственной виртуальной среде на любом компьютере, соответствующем требованиям данного документа, без необходимости добавления библиотек к актуальному дистрибутиву Anaconda. Приложение является локальным приложением с графическим интерфейсом, которое работает на изолированном компьютере. Это означает, что для работы приложения не нужны вычислительные ресурсы Интернета и не используется клиентсерверная архитектура (за исключением Web-интерфейса). В тоже время, приложение может анализировать данные из Интернета.

# Требование к программной документации

Исполнитель создает два документа: «Руководство пользователя», содержащее подробное описание решаемой задачи, инсталляции, запуска и интерфейса приложения, и «Руководство разработчика», содержащее подробное описание архитектуры приложения — состав библиотек, функций и классов, связь между ними, структура каталогов и описание параметров, управляющих работой приложения. Документация формируется в виде двух файлов формата MS Word 2016 или Adobe Acrobat.

# Технико-экономические показатели

В рамках данной работы расчёт экономической эффективности не предусмотрен.

Разрабатываемый инструмент сделает еще более удобным поиск музыки и сократит время, затрачиваемое на поиск. Что, в свою очередь, позволит меломанам посвящать больше времени именно самой музыке, а не ее поиску.

# Стадии и этапы разработки

Табл. 1

|  |  |
| --- | --- |
| № | Название стадии |
| 1 | Формулировка основных функций и выбор датасета, библиотек |
| 2 | Разработка основных функций (Backend) |
| 3 | Разработка интерфейса |
| 4 | Соединение двух частей (интерфейса и функций) |
| 5 | Тестирование и исправление ошибок |
| 6 | Написание документации |
| 7 | Защита |

# Порядок контроля и приемки

Оценивание проекта происходит в конце модуля. Необходимо прислать всю работу за 10 дней до начала сессии.

## Критерии оценивания

• Работоспособность приложения. При запуске и использовании функционала, указанного в «Руководстве пользователя», не должно возникать ошибок и не должно выводятся в консоль сообщений, связанных с некорректной работой приложения. Возможности языка Python и специально подобранные требования к разрабатываемому продукту позволяют с уверенностью утверждать, что следование указаниям задания для самостоятельной работы гарантирует работоспособность создаваемого при выполнении задания приложения

На любом компьютере, на котором установлена общая для всех студентов версия дистрибутива Python — Anaconda, при отсутствии ошибок разработчиков данного приложения. Таким образом, если разработчик корректно создал код приложения оно должно работать, как на его компьютере, так и на компьютере другого пользователя, в частности, преподавателя, при условии выполнения указанных выше требований. Отсутствие возможности приступить к эксплуатации приложение может свидетельствовать только, либо о наличии ошибок в коде, либо о нарушении требований задания, в частности, использовании специфических особенностей компьютера разработчика. Отсутствие возможности приступить к работе с приложением (критическая ошибка при запуске) или прекращение работы приложения по причине возникновения критической ошибки в ходе его эксплуатации автоматически приводит к оценке ноль баллов по десятибалльной системе. Наличие не критических ошибок приводит к адекватному снижению балла за приложение.

• Структура приложения. Полнота и качество реализации функционала и интерфейса, описанных в настоящем документе. Приложение должно соответствовать всем требованиям, указанным в настоящем документе. Отсутствие тех или иных требуемых элементов приводит к адекватному снижению балла за приложение. Полное несоответствие требованиям приводит к оценке ноль баллов по десятибалльной системе.

• Интерфейс. Качество реализации и удобство работы с интерфейсом всех частей приложения — минимально возможное наличие ручного ввода данных, интуитивная понятность и удобство размещения элементов графического интерфейса (виджетов), возможность конфигурирования интерфейса (настройки цветов и шрифтов, а также управления окнами с помощью мыши). Минимально допустимой реализацией настройки интерфейса является использование конфигурационных файлов. Приветствуется наличие графического интерфейса для выполнения настроек. Отсутствие возможности настраивать интерфейс является нарушением требований и приводит к оценке ноль баллов по десятибалльной системе.

• Реализация кода. Код должен быть разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for Python Code» и «PEP 257 -- Docstring Conventions» (см.

https://www.python.org/dev/peps/). Оценка качества кода в IDE Spyder должна быть не ниже 7 баллов. Более низкая оценка приводит к оценке ноль баллов по десятибалльной системе. Более высокий балл является аргументом для повышения оценки. Код должен содержать много комментариев, поясняющих назначение всех его частей. Количество и качество комментариев влияет на оценку.

• Документация к приложению. Оценивается качество реализации документации — «Руководства пользователя» и «Руководства разработчика». Документы должны иметь пронумерованные страницы и разбиты на разделы. В документах должно присутствовать оглавление, созданное средствами MS Word. При реализации в формате Adobe Acrobat, оглавление должно обеспечивать навигацию по документу. Технические требования: шрифт Calibri Light 12, межстрочный интервал 1.15, все таблицы и рисунки (скриншоты) пронумерованы и имеют название, каждый документ имеет титульный лист с указанием названия учебного заведения, учебной программы, учебной группы, номера и состава бригады (приведен телефон и адрес Директора).

Плагиат. За исключением кода используемых библиотек, служебного кода необходимого для их работы, а также кода, созданного библиотеками в процессе их работы, совпадение пятидесяти и более процентов строк кода с кодом, опубликованным в открытых источниках или кодом ранее выполненных индивидуальных проектов по данному курсу, считается плагиатом, и Исполнитель получает оценку ноль баллов по десятибалльной системе.