|  |  |
| --- | --- |
| C:\Users\bru\Desktop\лого 2 mini.png | Автономное учреждение профессионального образования  Ханты-Мансийского автономного округа – Югры  «Нефтеюганский политехнический колледж» |

**Курсовая работа (проект)**

Написание программы «MP3 player»

09.02.07 Информационные системы и программирование

Выполнил(а) работу

обучающийся 3 курса \_\_\_\_\_\_\_\_\_\_\_\_ Марченко Г.В.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Степанов В.А.

(подпись)

г. Нефтеюганск, 2023



Автономное учреждение профессионального образования

Ханты-Мансийского автономного округа – Югры

«Нефтеюганский политехнический колледж»

**Рецензия на курсовую работу (проект)**

**Обучающийся** Марченко Георгий Вячеславович

**Группа №** ИС1.21

**Специальность:** 09.02.07 Информационные системы и программирование

**Квалификация:** Программист

**ТЕМА:** Написание программы «MP3 player»

**1. Краткое описание курсовой работы (проекта) и принятых решений:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Отрицательные стороны работы(проекта)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Положительные стороны работы (проекта)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Оценка конструкторской разработки и графического оформления**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Оценка общеобразовательной подготовки и деловых качеств студента**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оценка работы:  \_\_\_\_\_\_   (\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Руководитель курсовой работы (проекта)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Степанов В.А.)

  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.



Автономное учреждение профессионального образования

Ханты-Мансийского автономного округа – Югры

«Нефтеюганский политехнический колледж»

**З  А  Д  А  Н  И  Е**

**на курсовую работу (проект)**

Студент группы ИС 1.21 Марченко Георгий Вячеславович

Тема задания Написание программы «MP3 player»

Содержание пояснительной записки Введение, Глава 1. Теоретическая часть, 1.1. Описание предметной области, 1.2. Обзор аналогов имеющихся решений проблем, 1.3. Обоснование необходимости разработки, 1.4. Постановка задачи, 1.5. Состав ПЭВМ, 1.6. Используемое программное обеспечение, Глава 2. Практическая часть, 2.1. Выбор средства автоматизации, 2.2Дополнительные программные средства, 2.3. Требование к программе, 2.4. Проектирование программное продукта, 2.5. Текст программы с описанием, 2.6. Тестирование и отладка, Заключение, Список использованной литературы.

Содержание графической части, практической работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель профессионального цикла (руководитель работы) Степанов В. А.

г. Нефтеюганск, 2023 г..

**Содержание**

[**Введение** 5](#_Toc154568258)

[**1.Теоретическая часть** 7](#_Toc154568259)

[**1.1 Описание предметной области** 7](#_Toc154568260)

[**1.2 Обзор аналогов имеющихся решений проблемы** 9](#_Toc154568261)

[**1.3 Обоснование необходимости разработки.** Обоснование необходимости разработки аудиоплеера: 11](#_Toc154568262)

[**1.4 Постановка задачи** 12](#_Toc154568263)

[**1.5 Состав ПЭВМ** 13](#_Toc154568264)

[**1.6 Используемое программное обеспечение** 14](#_Toc154568265)

[**2.Практическая часть** 17](#_Toc154568266)

[**2.1 Выбор средства автоматизации** 17](#_Toc154568267)

[**2.2 Дополнительные программные средства** 19](#_Toc154568268)

[**2.3 Требования к программе** 21](#_Toc154568269)

[**2.4 Проектирование программного продукта** 22](#_Toc154568270)

[**2.5 Текст программы с описанием** 25](#_Toc154568271)

[**2.6 Тестирование и отладка** 27](#_Toc154568272)

[**Заключение** 29](#_Toc154568273)

[**Список использованной литературы** 30](#_Toc154568274)

# **Введение**

Аудиоплеер – это ПО или устройство, используемое для воспроизведения аудиофайлов. Он позволяет пользователям прослушивать аудиофайлы различных аудиоформатов. С его помощью можно слушать музыку, аудиокниги, подкасты и другие аудиозаписи в различных форматах, которые были загружены в него. В целом, аудиоплееры являются удобным инструментом для организации, управления и наслаждения аудиозаписями в различных ситуациях – дома, в автомобиле, во время тренировки, во время путешествий и т.д. .

В данном курсовом проекте мы проведем анализ и оценку моего приложения, а также объясню, почему мы выбрали осуществить поставленную задачу именно так.

В итоговом варианте мы хотим увидеть, приложение, созданное для прослушивания аудиофайлов в формате .mp3, написанное на языке программирования - Python, и написанное на платформах Microsoft Visual Studio. Приложение должно загружать аудиофайлы с компьютера и воспроизводить их, посредственно через опции воспроизведения аудиофайла.

**Тема -**Разработка аудиоплеера на языке Python.

**Актуальность –** приложение более актуально для обычных пользователей ПК. Оно не сложное в использовании, можно даже назвать его буквально двухкнопочным и не занимающим много памяти в сравнении c многими известными аналогами. Легкое и простое приложение, позволяющее воспроизводить заранее загруженные в него композиции, сидя за компьютером у себя дома, разбавляя немного серую рутины жизни.

**Объект исследования –** Разработка приложений на языке программирования.

**Предмет исследования –** Разработка приложения «MP-3 плеер» на языке программирования Python.

**Цель –** Разработать функциональное приложение для прослушивания аудиофайлов .mp3 формата.

**Задачи:**

* Проанализировать существующие способы решения данного запроса; Выделить актуальные и понятные методы;
* Определить минимальный функционал программы;
* Реализовать требования с помощью выбранных методов разработки; Протестировать приложение;
* Выпустить готовую программу

**Практическая значимость работы –** Практическая значимость работы аудиоплеера состоит в его способности обеспечить комфортное воспроизведение аудиофайлов, создавать персонализированные плейлисты, предлагать удобный пользовательский интерфейс и дополнительные функции, что делает его незаменимым инструментом для многих людей в их повседневной жизни.

**Методы исследования**

**Теоретические:** Анализ разновидности приложений и иных идей для заданной цели.

**Научные:** экспериментальное тестирование различных элементов функционала программы.

В **структуру** курсовой работы входит: Введение; Теоретическая часть; Практическая часть; Заключение; Список литературы и приложения

# **1.Теоретическая часть**

# **1.1 Описание предметной области**

Аудиоплеер - это устройство или программное обеспечение, предназначенное для воспроизведения аудиофайлов. Предметная область аудиоплеера включает в себя следующие основные элементы:

1.Воспроизведение аудиофайлов: аудиоплеер позволяет пользователю воспроизводить различные форматы аудиофайлов, такие как MP3, WAV, FLAC, AAC и другие.

2.Управление воспроизведением: аудиоплеер обычно предоставляет функции управления воспроизведением, такие как воспроизведение, пауза, перемотка, повторение и перемешивание треков.

3.Организация музыкальной библиотеки: аудиоплеер может иметь возможность организации музыкальной коллекции пользователя, включая создание плейлистов, категоризацию по жанрам, исполнителям и альбомам.

4.Эквалайзер и звуковые эффекты: многие аудиоплееры предоставляют пользователю возможность настройки звучания с помощью эквалайзера и применения различных звуковых эффектов.

5.Поддержка онлайн-сервисов: некоторые аудиоплееры имеют функциональность доступа к онлайн-музыкальным сервисам, таким как Spotify, Apple Music, Google Play Music и другим.

6.Совместимость с различными устройствами: аудиоплееры могут быть предназначены для работы на компьютерах, мобильных устройствах, портативных аудиоплеерах и других устройствах.

7.Пользовательский интерфейс: интерфейс аудиоплеера обычно включает в себя элементы управления воспроизведением, отображение информации о треках и удобный доступ к функциям управления музыкальной библиотекой.

Это основные элементы предметной области аудиоплеера, которые определяют его функциональность и возможности.

# **1.2 Обзор аналогов имеющихся решений проблемы**

**Цели обзора:**

1.Изучить основные функции и возможности аналогов аудиоплееров.

2.Сравнить их с требованиями к искомому решению.

3.Сформулировать выводы о актуальности выбранной темы работы и поставленных в ней задач.

**Критерии сравнения:**

1.Поддерживаемые форматы аудиофайлов: способность воспроизводить широкий спектр форматов, включая популярные и менее распространенные.

2.Функциональность: наличие возможности создания плейлистов, поддержка радио-трансляций, наличие эквалайзера и других настроек звука, возможность организации музыкальной библиотеки.

3.Интерфейс: удобство использования, наличие настраиваемых элементов интерфейса, интуитивно понятное управление.

4.Настраиваемость: возможность изменения внешнего вида, добавления дополнительных функций через плагины или расширения.

5.Потоковая передача мультимедийного контента: возможность потокового воспроизведения аудио и видео файлов.

6.Доступ к музыкальной библиотеке: возможность покупки и загрузки музыки, организация и синхронизация с устройствами.

**Обзор аналогов:**

1. Winamp:

-Поддерживаемые форматы: MP3, AAC, FLAC, WAV, WMA.

-Функциональность: создание плейлистов, поддержка радио-трансляций, эквалайзер.

-Интерфейс: настраиваемый интерфейс с возможностью изменения скинов.

-Настраиваемость: поддержка плагинов для добавления дополнительных функций.

-Потоковая передача: отсутствует.

-Доступ к музыкальной библиотеке: нету магазина и загрузки музыки

2.VLC Media Player:

-Поддерживаемые форматы: почти все аудио и видео форматы.

-Функциональность: создание плейлистов, поддержка потоковой передачи мультимедийного контента.

-Интерфейс: простой и интуитивно понятный интерфейс.

-Настраиваемость: возможность настройки параметров воспроизведения.

-Потоковая передача: поддерживается.

-Доступ к музыкальной библиотеке: отсутствует магазин для покупки и загрузки музыки.

3.iTunes:

-Поддерживаемые форматы: MP3, AAC, ALAC, WAV, AIFF.

-Функциональность: создание плейлистов, доступ к магазину для покупки и загрузки музыки.

-Интерфейс: стильный и интуитивно понятный интерфейс.

-Настраиваемость: ограниченная настройка внешнего вида.

-Потоковая передача: поддерживается.

-Доступ к музыкальной библиотеке: доступ к магазину для покупки и загрузки музыки.

4.Foobar2000:

-Поддерживаемые форматы: почти все аудио форматы благодаря поддержке плагинов.

-Функциональность: создание плейлистов, поддержка радио-трансляций, эквалайзер.

-Интерфейс: настраиваемый интерфейс с возможностью изменения внешнего вида.

-Настраиваемость: высокая настраиваемость через плагины и расширения.

-Потоковая передача: отсутствует.

-Доступ к музыкальной библиотеке: отсутствует магазин для покупки и загрузки музыки.

5.Spotify:

-Поддерживаемые форматы: потоковая передача музыки в формате Ogg Vorbis.

-Функциональность: создание персонализированных плейлистов, рекомендации на основе предпочтений пользователя.

-Интерфейс: стильный и удобный интерфейс.

-Настраиваемость: ограниченная настройка внешнего вида.

-Потоковая передача: основная функция.

-Доступ к музыкальной библиотеке: доступ к огромной библиотеке треков с возможностью создания плейлистов.

**Выводы:**

Обзор аналогов аудиоплееров показал разнообразие функций и возможностей, предлагаемых различными программами. В то время как некоторые аудиоплееры ориентированы на локальное хранение и воспроизведение музыкальных файлов, другие предлагают потоковую передачу мультимедийного контента с доступом к огромным библиотекам треков. Важно учитывать требования к новому решению при разработке аудиоплеера, чтобы обеспечить соответствие потребностям пользователей.

**1.3 Обоснование необходимости разработки.**  
Обоснование необходимости разработки аудиоплеера:

1.Удовлетворение потребностей пользователей: разработка аудиоплеера позволит предложить пользователям удобный и многофункциональный инструмент для воспроизведения аудиофайлов, соответствующий современным требованиям.

2.Автоматизация процессов: Разработка аудиоплеера позволит автоматизировать процессы воспроизведения аудиофайлов. Аудиоплеер может предложить автоматизацию процессов создания плейлистов, организации музыкальной библиотеки, что значительно упростит работу пользователя с аудиофайлами.

3.Улучшение качества звука: разработка аудиоплеера с возможностью настройки звука и использованием современных технологий обработки звука позволит пользователям наслаждаться высококачественным звуком.

4.Повышение удобства использования: разработка аудиоплеера с интуитивным интерфейсом, удобными функциями управления, возможностью настройки звука позволит пользователям комфортно использовать программу.

5.Интеграция с другими сервисами: возможность интеграции с онлайн-сервисами для потокового воспроизведения музыки, доступ к магазинам для покупки и загрузки музыки, синхронизация с мобильными устройствами.

6.Интеграция с другими устройствами: целью разработки аудиоплеера может быть обеспечение совместимости с различными устройствами, такими как смартфоны, планшеты, автомобильные системы и другие аудиоустройства

7.Освобождение пользователей от рутины: аудиоплеер может освободить пользователей от необходимости самостоятельно организовывать свою музыкальную библиотеку, предложив функции автоматической организации и каталогизации аудиофайлов.

Таким образом, разработка нового аудиоплеера будет направлена на удовлетворение потребностей пользователей, повышение удобства использования и качества звука и освобождение пользователей от рутины

# **1.4 Постановка задачи**

**Функциональное назначение аудиоплеера:**

1.Воспроизведение аудиофайлов различных форматов.

2.Поиск и организация аудиофайлов в библиотеке.

3.Поддержка сортировки и фильтрации аудиофайлов по различным критериям (название, исполнитель, жанр и т.д.).

4.Возможность управления воспроизведением с помощью горячих клавиш

Эксплуатационное назначение:

Аудиоплеер будет использоваться пользователями для прослушивания музыки, подкастов, аудиокниг и других аудиофайлов на персональных компьютерах, мобильных устройствах и других платформах. Результаты работы программы будут использоваться для воспроизведения аудиофайлов с целью развлечения, работы или обучения.

Информационная модель задачи:

Входная информация:

-Аудиофайлы различных форматов (MP3, WAV, FLAC и т.д.).

-Данные о плейлистах и метаданные аудиофайлов (название, исполнитель, альбом и т.д.).

Выходная информация:

-Звуковой сигнал, воспроизводимый на аудиоустройстве.

-Отображение информации о текущем треке (название, исполнитель, обложка альбома и т.д.).

Идентификаторы:

-Название аудиофайла.

-Исполнитель.

-Название плейлиста.

-Название альбома.

Таким образом, аудиоплеер предназначен для управления аудиофайлами, воспроизведения музыки и обеспечения удобного доступа к контенту.

# **1.5 Состав ПЭВМ**

Для создания аудиоплеера на ПЭВМ (персональном компьютере) используется следующая вычислительная техника:

1.Процессор (CPU): Основной вычислительный компонент, отвечающий за выполнение команд программы, обработку данных и управление операциями воспроизведения аудиофайлов.

2.Оперативная память (RAM): Используется для временного хранения данных, буферизации аудиофайлов и обеспечения плавного воспроизведения.

3.Жесткий диск (HDD/SSD): Хранение аудиофайлов, метаданных, плейлистов и другой информации о библиотеке аудиоконтента.

4.Звуковая карта: Отвечает за преобразование цифрового звукового сигнала в аналоговый формат для воспроизведения через акустическую систему или наушники.

5.Акустическая система или наушники: Используются для воспроизведения звукового сигнала, который обрабатывается аудиоплеером.

6.Видеокарта (опционально): В случае, если аудиоплеер имеет графический интерфейс, видеокарта отвечает за отображение элементов управления, обложек альбомов и другой визуальной информации.

7.Устройства ввода (клавиатура, мышь, сенсорный экран): Используются для управления аудиоплеером, выбора треков, регулировки громкости и других действий.

Для разработки потребовалась следующая вычислительная техника:

1. Процессор: Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz.
2. Оперативная память: 16 ГБ (доступно 15,9 ГБ).
3. Твердый накопитель SSD: 500 ГБ
4. Звуковая карта: NVIDIA High Definition Audio.
5. Акустическая система: Наушники Bloody G300.
6. Видеокарта: Nvidia GeForce GTX 1650 (4 ГБ, DDR6)
7. Устройства ввода: Клавиатура – MSI Vigor GK30, Мышь – Bloody P30 Pro

# **1.6 Используемое программное обеспечение**

Программное обеспечение (ПО) представляет собой набор инструкций, программ и данных, необходимых для работы компьютерной системы. Оно включает в себя системное программное обеспечение, которое обеспечивает базовые функции компьютера, и прикладное программное обеспечение, предназначенное для решения конкретных задач пользователей.

Структура программного обеспечения включает следующие компоненты:

1.Системное программное обеспечение:

-Операционная система (ОС): Является основным компонентом системного ПО. Она управляет ресурсами компьютера, обеспечивает выполнение программ, управляет вводом-выводом данных, обеспечивает безопасность и т.д.

-Драйверы устройств: Обеспечивают взаимодействие операционной системы с аппаратными устройствами компьютера, такими как звуковая карта, видеокарта, жесткий диск и другие.

2.Прикладное программное обеспечение:

-Аудиоплеер: Программа для воспроизведения аудиофайлов.

-Библиотеки и фреймворки: Используются для разработки аудиоплеера, обеспечивая доступ к функциям воспроизведения, управления библиотекой аудиофайлов и другим функционалом.

-Плагины и расширения: Могут использоваться для добавления дополнительных функций или поддержки различных форматов аудиофайлов.

Роль операционной системы в составе программного обеспечения заключается в обеспечении работы аппаратных устройств компьютера, управлении процессами выполнения программ, обеспечении безопасности данных и ресурсов, а также предоставлении интерфейса для взаимодействия пользователя с компьютером.

Средства системного программного обеспечения, без которых не может функционировать аудиоплеер и связанная с ним техника, включают операционную систему, драйверы звуковой карты и других устройств, а также необходимые библиотеки и фреймворки для разработки и запуска.

Аудиоплеер разрабатывался на операционной системе Windows 11 Pro

Windows 11 – весьма перспективная система, и в будущем её наверняка ждет широкое распространение. Главные преимущества ОС – новые возможности управления приложениями, функции для кастомизации интерфейса и повышенная безопасность.

# **2.Практическая часть**

# **2.1 Выбор средства автоматизации**

Для автоматизации аудиоплеера можно использовать различные инструментальные системы и средства автоматизации, включая языки программирования, библиотеки и фреймворки. Вот несколько возможных вариантов:

1.Язык программирования Python: Python - это высокоуровневый язык программирования, который обладает богатой экосистемой библиотек для работы с аудиофайлами и звуковой обработки, таких как librosa, pydub, soundfile и другие. Он также имеет библиотеки для создания графического интерфейса пользователя (GUI), что может быть полезно для создания пользовательского интерфейса аудиоплеера.

2.Библиотека LibVLC: LibVLC - это мультимедийная библиотека, которая предоставляет широкие возможности для воспроизведения аудио и видео, а также поддерживает множество форматов медиаданных. Она может быть использована для создания кастомного аудиоплеера с различными функциями воспроизведения.

3.JavaScript и Web Audio API: Если вы хотите создать веб-приложение аудиоплеера, то JavaScript совместно с Web Audio API может быть хорошим выбором. Web Audio API предоставляет возможности для работы с звуком в браузере, включая воспроизведение, обработку и синтез звука.

4.Программы для разработки музыки и звука (например, Ableton Live, FL Studio): Эти программы предназначены для создания музыки, но также могут быть использованы для автоматизации воспроизведения аудиофайлов и звуковой обработки.

При выборе инструментальной системы для автоматизации аудиоплеера необходимо учитывать следующие критерии:

-Возможности работы с аудиофайлами и звуковой обработки.

-Наличие готовых решений или библиотек для создания пользовательского интерфейса.

-Поддержка различных форматов аудиофайлов.

-Простота использования и наличие документации.

Выбор конкретной инструментальной системы будет зависеть от конкретных требований проекта, опыта разработчиков и доступных ресурсов.

Конкретно моим выбором средства автоматизации стал язык программирования Python. Python является одним из самых популярных языков программирования для автоматизации благодаря своей простоте, гибкости и мощному набору библиотек. С помощью Python можно автоматизировать различные процессы, включая задачи по обработке данных, веб-скрапингу, тестированию, работе с файловой системой, автоматизации рабочего процесса и многому другому.

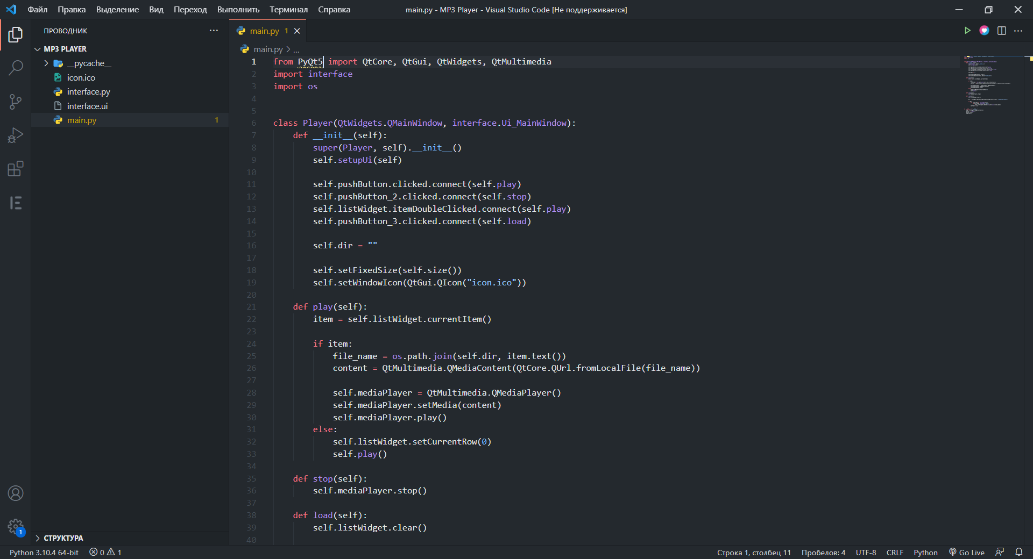
Одна из самых интересных особенностей языка Python – это его аудиомодули, позволяющие написать простой и рабочий аудиоплеер, воспроизводящий аудиофайлы, пусть даже самый простой.

Платформой же, или же редактором исходного кода послужило приложение Visual Studio Code.

VS Code – редактор исходного кода. Он имеет многоязычный интерфейс пользователя и поддерживает ряд языков програмирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности.

С его помощью можно разрабатывать кросс-платформенные приложения, используя веб-технологии. Суперсила VS Code в том, сто он не привязан к определенному языку программирования, поэтому с его помощью можно создавать сайты, мобильные приложения, работать с базами данных, тестировать сервисы и т.д. .

В общем плане VS Code предоставляет множество возможностей для разработки ПО.



*Изображение 1 – Скриншот программы VS Code*

# **2.2 Дополнительные программные средства**

При разработке программного продукта аудиоплеера могут быть использованы различные программные средства для создания интерфейса, отчетов, меню и других элементов пользовательского взаимодействия. Вот несколько возможных инструментов и средств, которые могут быть использованы:

-Генератор отчета: Для создания отчетов о проигранных треках, статистики использования плеера и другой информации можно использовать инструменты для генерации отчетов, такие как JasperReports, Crystal Reports или библиотеки для генерации PDF-файлов.

-Генератор меню: Для создания пользовательского интерфейса с меню и навигацией можно использовать инструменты для создания графического интерфейса пользователя (GUI), такие как PyQt, Tkinter, JavaFX и другие.

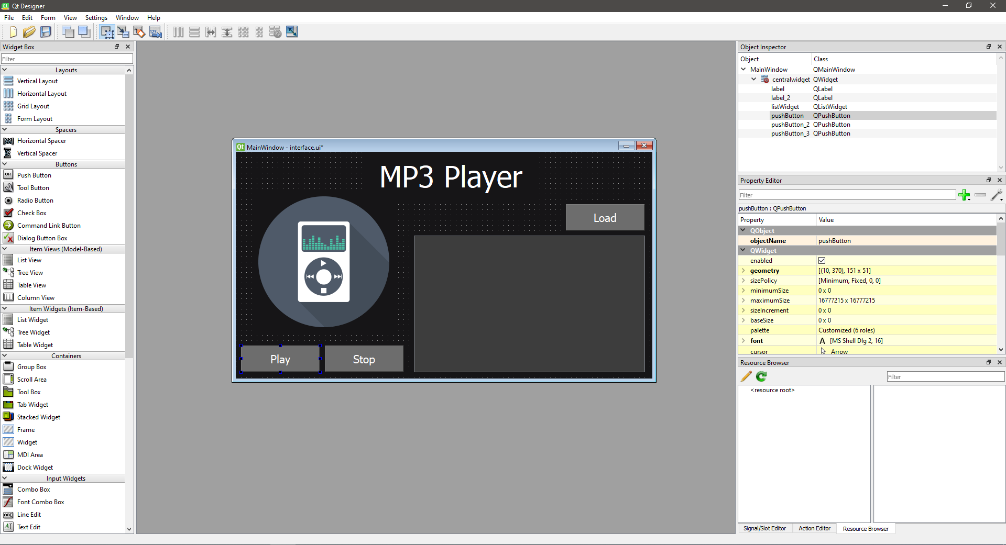
-Генератор экрана: Для создания экранов воспроизведения, списка треков, настроек и других элементов пользовательского интерфейса можно использовать инструменты для разработки пользовательского интерфейса, такие как HTML/CSS/JavaScript для веб-приложений или специализированные фреймворки для создания графического интерфейса.

-Формирование исполняемого файла (дистрибутива): Для создания исполняемого файла или дистрибутива аудиоплеера можно использовать инструменты для сборки и упаковки приложений, такие как PyInstaller, cx\_Freeze, Electron (для веб-приложений) и другие.

-Средства SQL: При работе с базой данных для хранения информации о треках, плейлистах, настроек и других данных можно использовать язык SQL и соответствующие СУБД (например, SQLite, MySQL, PostgreSQL).

-Стандартные процедуры для организации интерфейса "программа-пользователь": В процессе разработки аудиоплеера могут быть использованы стандартные процедуры для организации взаимодействия с пользователем, такие как обработка событий, валидация ввода, управление состоянием приложения и другие.

Из всего выше перечисленного, потребовалось в общем виде воспользоваться инструментами для генерации меню для создания интерфейса аудиоплеера. Использовался PyQt5 - набором библиотек, написанных на языке программирования Python для Qt — кросс-платформенного фреймворка, используемого в разработке графических интерфейсов пользователя (GUI). PyQt5 предоставляет богатый набор готовых виджетов, таких как кнопки, поля ввода, таблицы, диаграммы и другие. Благодаря этому разработчики могут быстро проектировать разнообразные элементы пользовательского интерфейса. Библиотека позволяет также создавать собственные пользовательские виджеты, адаптированные под конкретные потребности приложения. Это дает возможность создавать уникальные и индивидуальные интерфейсы. Она также включает PyQt5 Designer, конструктор графического пользовательского интерфейса. Он существенно упрощает создание GUI — разработчику достаточно расположить готовые элементы в подходящих местах макета. PyQt может генерировать код Python из Qt Designer. Также в конструктор можно добавить новые элементы управления графическим интерфейсом, написанные на Python.



*Изображение 2 Скриншот программы QtDesigner*

# **2.3 Требования к программе**

Требования к программе аудиоплеера:

1.Требования к функциональным характеристикам:

-Поддержка воспроизведения различных аудиоформатов (например, MP3, WAV, FLAC и т.д.).

-Возможность управления воспроизведением (пауза, воспроизведение, перемотка и т.д.).

-Отображение информации о текущем треке (название, исполнитель, альбом и т.д.).

-Регулировка громкости и управление звуковыми эффектами (эквалайзер, эффекты пространственного звука и т.д.).

-Поддержка функций повтора и перемешивания треков.

-Поддержка функций менеджера файлов (просмотр, поиск, сортировка и т.д.).

2.Требования к надежности:

-Контроль целостности и правильности воспроизведения аудиофайлов.

-Обработка и предотвращение возможных ошибок и сбоев в работе программы.

-Создание резервных копий промежуточных результатов для предотвращения потери данных.

-Обеспечение корректного воспроизведения при возможных сбоях в системе, таких как потеря сетевого соединения или переключение на другое устройство вывода звука.

3.Требования к составу и параметрам технических средств:

-Минимальная конфигурация: процессор с поддержкой аппаратного декодирования аудио, минимальный объем оперативной памяти (например, 1 ГБ) и достаточное количество места на жестком диске для хранения.

-Рекомендуемая конфигурация: мощный многоядерный процессор, объем оперативной памяти (например, 4 ГБ или более) и достаточное количество места на жестком диске для хранения аудиофайлов.

4.Требования к информационной и программной совместимости:

-Поддержка различных операционных систем, таких как Windows, macOS, Linux и т.д.

-Совместимость с различными версиями операционных систем, языками программирования и другими программными средствами, необходимыми для функционирования программы.

Эти требования помогут определить функциональные возможности, надежность, требования к техническим средствам и совместимость программы аудиоплеера.

# **2.4 Проектирование программного продукта**

При проектировании программного продукта аудиоплеера можно применять различные приемы проектирования. Некоторые из них включают:

1.Технология проектирования сверху вниз (Top-Down) - при этом подходе анализируются высокоуровневые требования и функциональности аудиоплеера, затем они разбиваются на более мелкие составляющие. Это позволяет вам сперва определить общую структуру и основные компоненты системы, а затем углубиться в детализацию.

2.Модульное проектирование - это подход, при котором система разбивается на отдельные модули, каждый из которых отвечает за выполнение определенной функции или задачи. Например, можно выделить модуль для управления воспроизведением, модуль для работы с файлами, модуль обработки звука и т.д. Это позволяет создать более модульный и гибкий код.

3.Объектно-ориентированное проектирование - основано на концепции объектов, которые представляют реальные сущности или абстракции. Каждая функциональность аудиоплеера может быть представлена как объект с определенными свойствами и методами. Этот подход упрощает разработку и позволяет создавать более гибкие и удобные для использования системы.

Следующим шагом после выбора подхода проектирования является создание схемы алгоритма для основной программы аудиоплеера. Эта схема может включать следующие шаги:

1.Загрузка списка аудиофайлов и создание плейлиста.

2.Выбор и воспроизведение трека.

3.Управление воспроизведением (пауза, перемотка, изменение громкости).

4.Отображение информации о треке (название, исполнитель, альбом).

5.Применение звуковых эффектов и настройка звука.

6.Взаимодействие с пользовательским интерфейсом (например, кнопки управления).

Также может быть полезно создать схемы отдельных подпрограмм и модулей для более подробного представления работы системы. Например, схема работы модуля управления воспроизведением или модуля работы с файлами.

Для организации взаимодействия данных можно использовать схему взаимодействия данных, которая показывает как данные передаются между различными модулями и компонентами системы.

Информационная модель данных представляет собой описание данных, используемых в аудиоплеере, и их взаимосвязь. Например, это может быть модель, включающая треки, аудиофайлы, информацию об исполнителях и альбомах. Описание интерфейса и основных модулей программы должно включать описание методов и функций, которые могут быть использованы для взаимодействия с системой, а также описание основных функциональных возможностей каждого модуля.

Был выбран метод модульного проектирования, так как мы посчитали его более подходящим для выполнения данных требований и исполнения заданных функций. Модульное проектирование является эффективным подходом к разработке программного обеспечения. Оно основано на разбиении приложения на отдельные компоненты, называемые модулями. Каждый модуль выполняет конкретную функцию и может быть разработан и тестирован независимо от других модулей. Такой подход облегчает сопровождение и расширение кода, а также повышает его переиспользуемость.

Некоторые из основных преимуществ модульного проектирования:

1.Масштабируемость: Модульность позволяет разрабатывать и внедрять новые функции, не затрагивая другие части приложения. Это делает процесс добавления нового функционала более гибким и удобным.

2.Повторное использование кода: Компоненты, разработанные в виде модулей, могут быть использованы в разных проектах. Повторное использование кода ускоряет разработку и снижает риск ошибок.

3.Удобство сопровождения: Каждый модуль легко понять и изменить независимо от других модулей. Это облегчает поиск и исправление ошибок, а также проведение обновлений и исправлений.

4.Тестирование: Модульное проектирование позволяет проводить тестирование отдельных модулей независимо. Это упрощает процесс обнаружения и устранения ошибок в ранних стадиях разработки.

В целом, модульное проектирование способствует созданию более гибкого, масштабируемого и надежного программного обеспечения.

# **2.5 Текст программы с описанием**

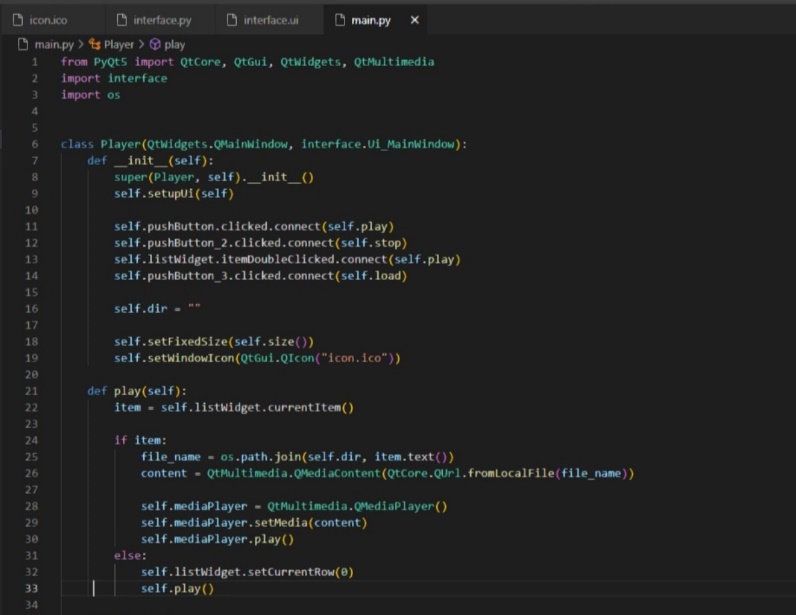
Запуск программы осуществляется непосредственно через VS Code, через корневой файл main.py.

Код для аудиоплеера писался на языке Python, в редакторе кода VS Code

В файле main.py. прописаны основные функции для кнопок интерфейса плеера.

Для написания кода использовалась библиотека PyQt5, которая загружалась через консоль редактора.

Затем импортировался интерфейс программы. После идут функции опций плеера такие как: Воспроизведение аудиофайла(Play), Остановка аудиофайла(Stop), Загрузка папки с адуиофайлами(Load).



Изображение 3 Код программы (часть 1)

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Изображение 4 Код программы (часть 2)

Интерфейс программы разрабатывался в QtDesigner.

Для его создания понадобилось использовать 2 кнопки для проигрывания и остановки композиции, а также еще 1 кнопку для загрузки корневой папки с файлами аудиоформата.

Использовались 2 виджета в виде лейблов для названия программы(MP3 Player) и для иконки МП-3 плеера. Также использовался

виджет-список, где отображаются загруженные файлы для воспроизведения.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Автоматически созданное описание

Изображение 5 Готовый вариант интерфейса

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Изображение 6 Использованные виджеты

# **2.6 Тестирование и отладка**

При тестировании программы не было обнаружено программных сбоев или иных ошибок программы. Программа без проблем запускается и воспроизводит аудиофайлы MP3 формата. При переносе программы на другой ПК не замечено файловых повреждений или повреждений кода.

Как таковых основных проблем обнаружено не было, программа спокойно исполняет свое главное назначение, а именно воспроизводит аудиофайлы, с возможностью их остановки.

Единственной погрешностью программы можно выделить только то, что она загружает файлы именно с папки в которой находятся аудиофайлы, а не по отдельности каждый аудиофайл, взятый с разных корневых расположений на жестком диске.

Сам плеер получился довольно простым во всех аспектах. Он имеет простой интерфейс с тремя лишь функциями: Воспроизведение, Остановка, Загрузка файлов. Также его код не сложный и не большой. Аудиоплеер прост в разработке. Исходя из этих фактов – он не слишком нагружает систему, то есть не требует много оперативной памяти, как и не занимает большого пространства на жёстком диске.

В целом не было задачи написать сверхсложную программу для воспроизведения аудиофайлов. Главная цель была – простота.

В ходе разработки использовались следующий видытестирования:

1. **Модульное тестирование:**

-Тестирование функций: Проверка отдельных функций аудиоплеера, таких как воспроизведение, пауза, перемотка и управление громкостью.

-Тестирование обработки ошибок: Проверка реакции аудиоплеера на различные виды ошибок, такие как отсутствие файла или неправильный формат аудио.

1. **Интеграционное тестирование:**

-Тестирование взаимодействия с внешними системами: Проверка, как аудиоплеер взаимодействует с операционной системой, библиотеками аудио и другими внешними компонентами.

1. **Системное тестирование:**

-Тестирование производительности: Оценка производительности при воспроизведении большого количества аудиофайлов или при работе с большим объемом данных.

-Тестирование совместимости: Убедиться, что аудиоплеер работает на различных операционных системах и поддерживает разные форматы аудио.

1. **Приемочное тестирование:**

-Тестирование функциональности: Проверка выполнения основных функций аудиоплеера в соответствии с требованиями.

-Тестирование пользовательского интерфейса: Убедитесь, что пользовательский интерфейс интуитивно понятен и легок в использовании.

1. **Тестирование безопасности:**

-Тестирование на уязвимости: Проверка наличия уязвимостей, которые могут быть использованы для вредоносных атак.

6. **Тестирование сценариев использования:**

-Тестирование различных сценариев воспроизведения: Проверка работы аудиоплеера в различных сценариях использования, таких как воспроизведение плейлистов, создание собственных списков воспроизведения и т. д.

7. **Тестирование с нагрузкой:**

-Тестирование устойчивости: Проверка, как аудиоплеер ведет себя при длительном воспроизведении или при работе с большим количеством одновременно воспроизводимых аудиофайлов.

# **Заключение**

В заключение проекта разработки MP-3 плеера на языке программирования Python можно отметить следующее.

В ходе выполнения проекта было проведено исследование требований и анализ готовых аналогов, что позволило определить ключевые требования к внешнему виду и функциональности программы. Простой и интуитивно понятный интерфейс с минимальным набором необходимых функций были приоритетными факторами при разработке приложения.

Выбор разработки приложения на языке Python, обоснован не только возможностью запуска приложения на различных платформах, таких как MacOS и Windows, но и относительно не сложным процессом разработки и простоты программы, не теряя при этом основной функционал программы.

Процесс постановки задачи включал определение требований к программе, описание ее основных функций и ожидаемых результатов. Также были выявлены основные цели и ограничения проекта, что способствовало более четкому определению пути разработки. Проектирование архитектуры приложения включало определение структуры, модулей и взаимодействия компонентов. Была использована клиент-серверная архитектура, модульная структура и обработка событий пользовательского взаимодействия для обеспечения функциональности и удобства использования приложения. Разработка приложения включала реализацию интерфейса, функциональности и тестирование приложения. Весь процесс разработки был сопровожден тестированием и отладкой для обеспечения качества и стабильной работы приложения.

Разработка приложения «MP-3 Player была успешной. Цели - достигнуты, а приложение отвечает требованиям и ожиданиям. Полученный результат представляет пользователям удобный и простой инструмент для воспроизведения аудиофайлов, обеспечивая простоту использования.

# **Список использованной литературы**

1. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. проф. образования / Г.Н.Федорова. — 3-е изд., стер. — М.: Издательский центр «Академия», 2019. — 288 с.
2. Федорова Г.Н. Разработка модулей программного обеспечения для ком-пьютерных систем : учебник для студ. учреждений сред.проф. образования / Г.Н.Федорова. — 3-е изд., испр. — М: Издательский центр «Академия», 2019. — 384 с.
3. Лутц, Марк. Л86 Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 832 с. : ил. — Парад, тит. Англ.
4. Доусон М. Программируем на Python <https://coollib.net/b/544517-maykl-douson-programmiruem-na-python/read>
5. Написание программы МП-3 плеера на Python. Python, PyQt5

https://ru.stackoverflow.com/questions/1370789/%D0%9C%D1%83%D0%B7%D1%8B%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D0%BF%D0%BB%D0%B5%D0%B5%D1%80-pyqt5-python