**Пояснительная записка**

**Тема работы:** создание музыкального инструмента с использованием фреймворка PyQt5.

**Заявленный функционал:**

* Плеер с возможностью изменять громкость и перематывать треки по времени.
* Составление плейлиста
* Сохранение загруженных треков в базу данных и загрузка их при новом открытии приложения
* Отделение аудиодорожки от видеофайлов
* Скачивание видео с YouTube
* Создание простых инструментальных мелодий
* Графическое оформление

**Реализованный функционал:**

* Поддержка разных форматов (wav, mp3)
* Возможность сохранения и прослушивания инструментальных мелодий
* Плейлист пользовательских композиций (инструментальных мелодий),
* а также всё, что было заявлено.

**Запуск программы.**

При запуске программы открывается окно плейлиста с загруженными треками, если они есть. Пользователь может сразу начать слушать что-то, или загрузить файл.

**Рабочее пространство**

Выражено второй вкладкой с лаконичным названием «Инструменты».

Включает в себя:

1. Небольшую часть, отделенную для разнообразной работы с видеофайлами.

Здесь есть:

Кнопка для загрузки файла в программу с компьютера.

Если в названии файла есть пробелы, то будет создана копия без пробелов, так как иначе это приведет к ошибке при попытке предварительного просмотра видео.

Поле для ввода ссылки, в которую, если файл не выбран, можно ввести ссылку. Если же файл выбран, то там будет название выбранного файла.

Кнопка для скачивания с YouTube, при нажатии на неё с помощью библиотеки pytube

Индикатор прогресса скачивания, который будет активирован при нажатии кнопки скачивания и покажет пользователю степень загрузки файла.

Кнопку для отделения аудиодорожки, при нажатии на неё будет произведена операция отделения звука от видео. Потом откроется окно сохранения аудиофайла, а затем вопрос, добавить ли звук в плейлист.

Кнопку для предварительного просмотра, при нажатии на которую будет активирован файл play\_clip.py, запустив окно с просмотром видео.

1. Список инструментальных композиций и кнопки для работы с ними.

Включает в себя:

1.Плейлист с составленными мелодиями

2.Кнопку для изменения мелодии. При нажатии на неё будет последовательно выдано три всплывающих окна: изменить ли название мелодии, изменить ли имя файла, либо изменить его расположение.

3. Кнопку для удаления. Удаляет только мелодию из плейлиста.

4.Кнопку для создания мелодии. При нажатии на неё открывается пианино(о нём речь ниже)

5.Кнопку для прослушивания мелодии. При клико по ней активируется файл play\_melody.py, который с помощью библиотеки PyGame воспроизводит выбранный MIDI-файл.

3. Всплывающее окно с пианино.

Состоит из 26 основных кнопок, 2-ух дополнительных, текстового поля и одного QComboBox для выбора октавы, а также двух слайдеров, для выбора громкости и продолжительности.

QComboBox включает в себя варианты с цифрами от 1 до 8.

Это – октавы. Они определяют высоту звучания ноты. При выборе какого-то варианта текст на клавишах меняется. При этом смена было текста была достаточно сильно проработана, чтобы допустить минимум ошибок в визуальном представлении музыкального инструмента.

При нажатии на клавишу из клавиатуры пианино, в текстовом поле(QLabel) отображается название ноты, октава, длительность и громкость.

По нажатию на кнопку «Добавить в мелодию», если пользователь выбрал ноту, эта нота добавляется в мелодию. Если нота не выбрана – будет выдано предупреждение.

При нажатии на кнопку «прослушать мелодию» с помощью файла play\_melody.py будет проиграна композиция, а затем пользователю будет задан вопрос, сохранить ли файл и добавить ли его в плейлист на вкладке «инструменты».

Также в коде есть класс Db\_helper. Он помогает ускорить рутинные операции с базами данных, чтобы код был легче для понимания.