

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка приложения «Симулятор музыкального синтезатора»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 31ИС-23** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **А.О. Плюснина** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Л. Б. Гусятинер** |

**Москва 2024**



**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 31ИС-23 Иван Иванов**

**ТЕМА: Разработка приложения «Симулятор музыкального синтезатора»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

**СОДЕРЖАНИЕ**

[**Введение** 2](#_Toc184374622)

[**1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ** 5](#_Toc184374623)

[**2.** **ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИИ ПРОГРАММНОГО МОДУЛЯ** 9](#_Toc184374624)

[**Основные технологии и их использование** 9](#_Toc184374625)

[**Компоненты проекта** 10](#_Toc184374626)

[**Основные особенности и преимущества** 11](#_Toc184374627)

[**Результат** 12](#_Toc184374628)

[**3. ТЕСТИРОВАНИЕ И ОТЛАДКА РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ** 13](#_Toc184374629)

[**4. ЗАКЛЮЧЕНИЕ** 15](#_Toc184374630)

[**Список использованных источников и приложения** 18](#_Toc184374631)

[ПРИЛОЖЕНИЕ 5 Отзыв на курсовой проект 31](#_Toc184374632)

# **Введение**

Современные достижения в области цифровых технологий кардинально изменили подход к изучению и исполнению музыки. Каждый год появляются новые методы обучения, приложения и ресурсы, которые делают музыкальное образование более доступным для широкой аудитории. Музыка — это не только искусство, но и важный инструмент для самовыражения, поэтому спрос на инновационные подходы к обучению остается стабильно высоким. Сегодня большинство людей начинают осваивать музыкальные инструменты, развивать вокальные навыки или заниматься аранжировкой с помощью интернет-ресурсов, что позволяет существенно экономить время и деньги.

Использование онлайн-платформ и приложений значительно упрощает процесс получения необходимых знаний, как для профессиональных музыкантов, так и для начинающих. Эти технологии делают обучение более гибким и индивидуальным, позволяя каждому выбрать оптимальный темп и стиль освоения материала. В таком контексте создание приложения "Симулятор музыкального синтезатора" является не только актуальным, но и перспективным.

Цель разработки такого приложения заключается в предоставлении пользователям уникальной возможности исследовать мир музыкальных звуков, экспериментировать с их комбинациями, а также учиться играть на синтезаторе. Особая ценность проекта заключается в том, что пользователям больше не нужно приобретать дорогостоящие музыкальные инструменты для начала обучения. Это приложение способно превратить любой компьютер или смартфон в удобный инструмент для творчества, позволяя с легкостью комбинировать процесс обучения с развлечением.

Выбранная тема актуальна еще и потому, что она открывает новые горизонты в обучении и создании музыки. Разработка "Симулятора музыкального синтезатора" поможет не только сделать популярным музыкальное образование, но и вдохновить пользователей на новые креативные эксперименты.

Основная цель курсового проекта — разработка функционального приложения "Симулятор музыкального синтезатора", которое предоставляет пользователям возможность погрузиться в мир музыки.

Для достижения цели проекта поставлены следующие задачи:

1. Проанализировать основные принципы работы музыкальных синтезаторов.
2. Изучить существующие программные симуляторы и выявить их преимущества и недостатки.
3. Разработать архитектуру приложения, включающую интерфейс пользователя, основные модули и алгоритмы обработки звука.
4. Реализовать программное приложение пользовательский интерфейс.
5. Протестировать приложение и оценить его функциональность.

Объектом исследования является процесс создания и управления звуком с использованием музыкального синтезатора.

Предметом исследования является программная реализация функций музыкального синтезатора, включая генерацию звуков, изменение их характеристик и взаимодействие пользователя с интерфейсом приложения.

Практическая значимость разработанного приложения "Симулятор музыкального синтезатора" заключается в его широкой применимости как для обучения, так и для творчества. Программа предоставляет пользователям удобный инструмент для освоения базовых и продвинутых принципов работы с музыкальными синтезаторами. Это особенно важно для начинающих музыкантов, которые только начинают знакомство с миром цифровой музыки, а также для любителей, стремящихся создать свои первые музыкальные композиции.

В условиях активного развития цифровизации музыкальной индустрии, подобное приложение становится актуальным не только для индивидуального использования, но и как средство дополнительного образования. Оно может применяться в образовательных учреждениях, на курсах по созданию музыки, а также в онлайн-обучении, где пользователи смогут в интерактивной форме получать знания и практические навыки работы с синтезатором.

Кроме того, приложение имеет потенциал для использования в качестве инструмента для создания музыки в домашних условиях. Оно позволяет экспериментировать с различными звуковыми параметрами, такими как тембр, высота и продолжительность звука, что стимулирует развитие музыкального слуха и креативного мышления.

Реализация проекта не только удовлетворяет потребности современного музыкального образования и творчества, но и способствует популяризации музыкальной культуры. Пользователи получают возможность раскрыть свой потенциал без необходимости приобретения дорогостоящего оборудования. Таким образом, приложение гармонично вписывается в тренды развития доступных и инновационных решений для изучения и создания музыки.

# **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ**

В данном курсовом проекте тематической областью решения задач является виртуальное пианино. Программное обеспечение предназначено для реализации симулятора фортепиано с поддержкой воспроизведения звука, визуализации клавиш и хранения пользовательских настроек. Приложение можно использовать в образовательных целях, отработке музыкальных навыков и в качестве развлекательного инструмента. В целевую группу входят музыканты-любители, преподаватели музыки, студенты, обучающиеся игре на фортепиано, и пользователи, интересующиеся музыкальными приложениями. Основное назначение системы: 1. Реализация удобного интерфейса взаимодействия с виртуальным фортепиано. 2. Возможность регулировки громкости, сохранения и загрузки параметров. 3. Поддержка визуализации кнопок и воспроизведения звука в реальном времени.

Деятельность приложения заключается в выполнении следующих функций:

1. Обработка нажатий клавиш для воспроизведения соответствующих звуков.
2. Обеспечение визуальной обратной связи, выделяя активные клавиши.
3. Настройка громкости с помощью клавиш или графического интерфейса.
4. Сохранение и загрузка настроек (например, громкости) для дальнейшего использования.
5. Управление звуками с учётом назначенных клавиш и соответствующих звуковых файлов.

Необходимость компьютерной обработки данных.

1. Воспроизведение звуков, связанных с клавишами фортепиано.

2. Сохранение пользовательских параметров в файле для удобства манипулирования программой.

3. Реализация графического интерфейса для удобства взаимодействия с пользователем.

Изучение предметной области включает в себя следующее:

1. изучение существующих решений и аналогов (виртуальных музыкальных инструментов типа MIDI-симуляторов).

2. Определение функциональных требований к системе: воспроизведение звука, регулировка громкости, работа интерфейса.

3. Разработка структуры приложения с меню и окном фортепиано.

На основе анализа были определены функции, которые должна обеспечивать программа:

1. Воспроизведение звуков (белых и черных клавиш) при нажатии клавиш.

2. Регулировка громкости с отображением текущего уровня.

3. Сохраните настройки (громкость) в файле settings.json.

4. Загрузка настроек при запуске приложения.

5. Простой и удобный интерфейс, включая меню и режим главного окна.

Требования к программному обеспечению

Разрабатываемое программное обеспечение должно обеспечивать следующее:

1. Реализация виртуального фортепиано с графическим представлением клавиш.

2. обработка нажатия клавиш с воспроизведением звуков.

3. Управляйте параметрами (например, громкостью) через интерфейс и клавиатуру.

4. Сохранение и загрузка пользовательских настроек.

Визуализация данных:

1. Подсветка кнопок, соответствующих текущим действиям пользователя.

2. отображение подсказок (например, текущей громкости).

Организация входных и выходных данных:

1. Входные данные поступают посредством действий пользователя (нажатий клавиш и клавиш).

2. Выходные данные отображаются в виде звуков, визуальной обратной связи и сохраненных настроек.

Подходящей моделью жизненного цикла разрабатываемого программного обеспечения является «прототипирование», потому что:

1. Требования к программному обеспечению не были полностью известны на начальном этапе и уточнялись в ходе разработки.

2. Модель позволяет продемонстрировать готовый прототип с учетом комментариев и предложений.

Визуальные характеристики:

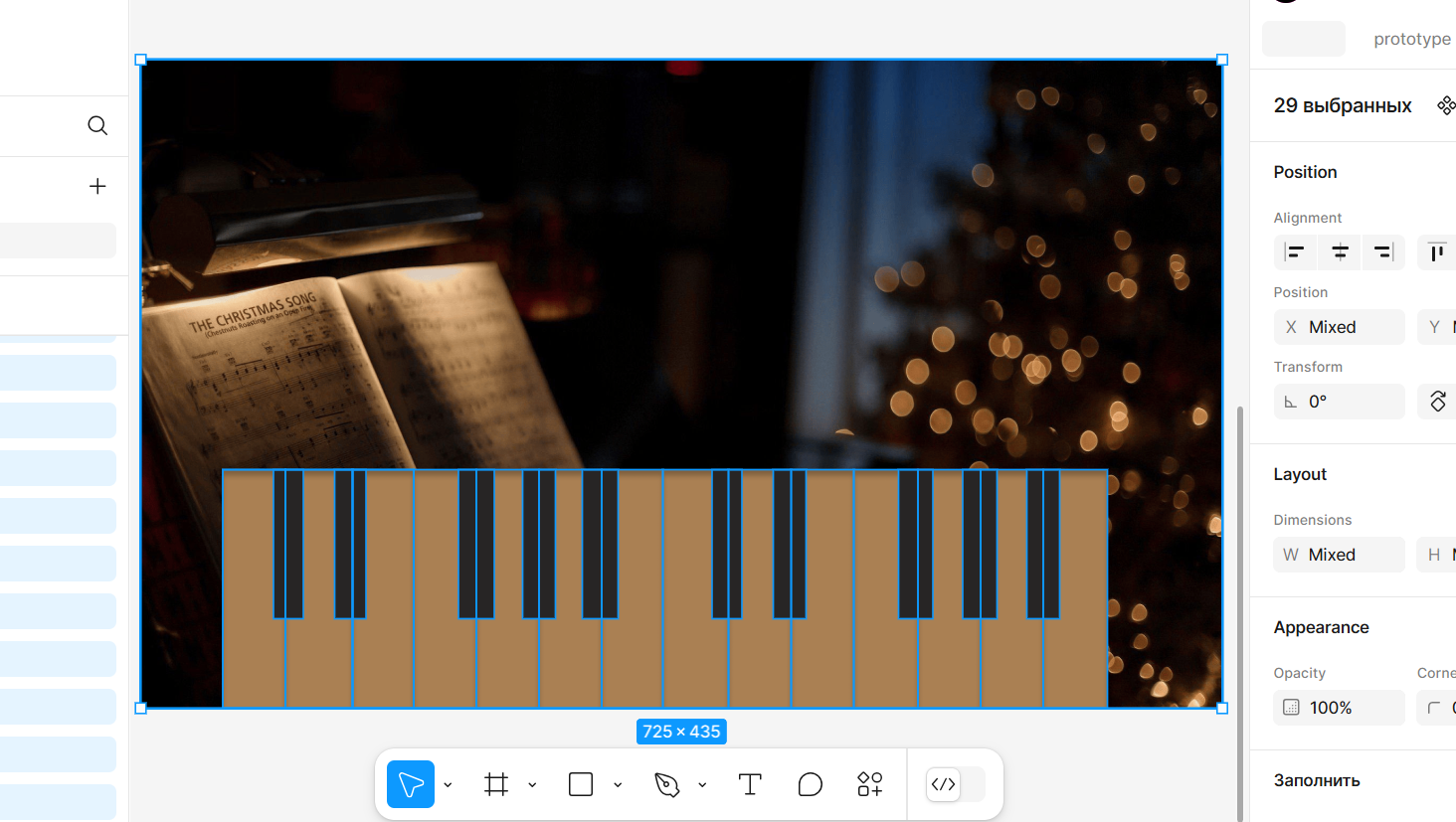
Клавиши фортепиано: отображаются как белые и черные прямоугольники, расположенные в стандартном порядке фортепианной клавиатуры.

\* Цвета клавиш: Белые клавиши отображаются светлым цветом, а черные клавиши - черным цветом.

\* Подписи клавиш: клавиши подписаны буквами, которые нажимают для их воспроизведения.

\* Размер и форма клавиш: Клавиши должны быть достаточно большими, чтобы их было удобно нажимать, но не настолько большими, чтобы они занимали слишком много места на экране. Форма клавиш также должна быть легко узнаваемой, чтобы пользователи могли быстро различать белые и черные клавиши.

\* Эффекты нажатия клавиш: при нажатии клавиши отображаться визуальный эффект изменения цвета. Это помогает пользователям подтвердить, что клавиша была нажата.



скриншот 1 прототип

# **2.** **ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИИ ПРОГРАММНОГО МОДУЛЯ**

Для реализации проекта «Виртуальное пианино с настройками» будет использована **PyCharm** — интегрированная среда разработки на языке Python. В проекте применяются библиотеки **PyQt5** для создания графического интерфейса и **Pygame** для воспроизведения звуков и обработки событий клавиатуры. Также используется модуль **JSON** для сохранения и загрузки пользовательских настроек.

**Основные технологии и их использование**

1. **PyQt5**
   1. Отвечает за разработку графического интерфейса, включая главное меню и основное окно пианино.
   2. Поддерживает работу с виджетами, такими как кнопки, текстовые поля, и элементы интерфейса для отображения клавиш пианино.
   3. Реализует события, связанные с нажатиями и отпусками клавиш, а также выводит подсказки пользователю, например, текущую громкость.
2. **Pygame**
   1. Используется для загрузки и воспроизведения звуковых файлов.
   2. Обрабатывает управление громкостью через клавиатуру, обеспечивая плавное изменение уровня звука.
   3. Работает с системой событий для воспроизведения звуков, соответствующих нажатию клавиш.
3. **JSON**
   1. Предоставляет возможность сохранения настроек (например, уровня громкости) в файл settings.json.
   2. Загружает параметры при запуске, что позволяет пользователю продолжать работу с сохранёнными настройками.

**Компоненты проекта**

**Главное меню**

Создано с помощью **PyQt5** и включает:

* **Фон**: отображается изображение, указанное в конфигурации (например, background.png).
* **Кнопки**:
  + *«Начать»*: запускает основное окно пианино.
  + *«Загрузить параметры»*: предварительно загружает настройки из файла, а затем открывает окно пианино.

Пример функции для обработки кнопки загрузки:

python

def load\_and\_start\_piano(self):

if not self.piano:

self.piano = PianoApp(self.save\_settings)

self.load\_settings()

self.menu.close()

self.piano.show()

**Окно пианино**

Графическая реализация пианино с поддержкой звуковых файлов и управления громкостью.

* **Клавиши пианино**:
  1. Белые клавиши отображаются с буквами (например, Q, W, E) и звуками.
  2. Чёрные клавиши имеют светлый текст для контраста.
* **Громкость**:  
  Пользователь может увеличивать и уменьшать громкость с помощью клавиш = и -. Громкость сохраняется в файл при нажатии кнопки *«Сохранить»*.

Пример изменения громкости:

python

def change\_volume(self, step):

self.volume = max(0.0, min(1.0, self.volume + step))

for sound in self.sounds.values():

sound.set\_volume(self.volume)

self.update()

* **Сохранение настроек**:

При нажатии кнопки *«Сохранить»* текущий уровень громкости сохраняется в файл settings.json, который затем загружается при повторном запуске.

**Основные особенности и преимущества**

1. **Удобный интерфейс**

Пользовательский интерфейс интуитивно понятен. Меню и окно пианино адаптированы для удобной работы.

1. **Гибкость настроек**

Проект позволяет сохранять и загружать пользовательские настройки. Это обеспечивает персонализацию и удобство использования.

1. **Кроссплатформенность**  
   Приложение может работать на любой операционной системе, поддерживающей Python, PyQt5 и Pygame.
2. **Расширяемость**  
   Код легко обновляется при добавлении новых функций, таких как запись воспроизводимых мелодий или добавление дополнительных звуковых эффектов.

**Результат**

Проект предоставляет пользователю функциональное приложение для игры на виртуальном пианино с возможностью настройки параметров. Благодаря использованию современных библиотек и удобных инструментов, приложение удобно в использовании и поддержке.

# **3. ТЕСТИРОВАНИЕ И ОТЛАДКА РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Тестирование черного ящика**

* Проверяет функциональность системы без учета ее внутренней структуры.
* Сосредотачивается на том, что делает система, а не на том, как она это делает.
* Использует тестовые данные, которые представляют различные сценарии использования.

**Определение тестируемых функций:**

* Обработка нажатий клавиш. (Таблица 1 тестирование)
* Управление громкостью (увеличение, уменьшение, сохранение). (Таблица 1 тестирование)
* Сохранение и загрузка настроек. (Таблица 1 тестирование)
* Отображение графического интерфейса. (Таблица 1 тестирование)
* Реакция на ошибки (например, неверные входные данные). (Таблица 1 тестирование)

В первой версии была реализована одна октава с возможностью ее смены между нижней и верхней. Проведя пользовательское тестирование было выявлено, что использование такой конструкции является неудобным, после чего последовала правка, в которой были реализованы две октавы из трех, что является достаточным для реализации нужд пользователя.

В одной из версий программы поступила жалоба от пользователей на отсутствие возможности регулировки уровня громкости, что создавало дискомфорт во время работы в программе. В последующей правке была создана возможность пользователям изменять громкость звука программы и запоминать текущую громкость, что позволило возвращаться к работе в программе с заранее настроенным звуком путем выбора варианта загрузить вместо варианта начать в стартовом окне работы программы.

# **4. ЗАКЛЮЧЕНИЕ**

В ходе разработки проекта курса «Виртуальное фортепиано» был создан интерактивный графический интерфейс, позволяющий пользователю воспроизводить звуки, соответствующие клавишам виртуального фортепиано. Реализована динамическая визуализация клавиш, состояние которых меняется при нажатии, а также добавлены функции регулировки октавы и громкости, что существенно расширяет возможности приложения.

Проект построен с использованием библиотек PyQt5 и Pygame, которые предоставляют мощный инструментарий для создания графического интерфейса и работы со звуками. Выбор этих библиотек позволил добиться высокой кроссплатформенной совместимости, а использование языка Python позволило обеспечить простоту реализации и гибкость настройки приложения.

Результатом является мультимедийное приложение с минималистичным и удобным интерфейсом, которое можно использовать в качестве учебного пособия или интерактивного музыкального инструмента.

Практические результаты

Графика и интерфейс

Разработан графический интерфейс с визуализацией клавиш фортепиано, основанной на пропорциях реального инструмента. Белые и черные клавиши представлены в интерактивном виде и меняют цвет при нажатии. Для удобства пользователя добавлены текстовые наложения, отображающие текущие настройки, включая активную октаву и громкость, а также инструкции по управлению.

Аудио

Звуки теперь можно воспроизводить для всех клавиш виртуального пианино. Каждой клавише присвоен собственный звуковой файл, соответствующий соответствующему тону. Программное обеспечение поддерживает воспроизведение звуков в нескольких октавах, что позволяет пользователю экспериментировать с диапазоном мелодий.

Элементы управления

Пользователь может взаимодействовать с приложением с помощью клавиатуры, меняя октавы и регулируя громкость. Для визуального информирования о текущем статусе в интерфейсе приложения в режиме реального времени отображается информация о выбранных настройках, что повышает удобство и наглядность работы приложения.

Рекомендации и предложения

1. Улучшение пользовательского интерфейса

Для улучшения интерактивности и эстетики можно ввести анимационные эффекты при нажатии клавиш. Например, динамические тени, плавные переходы цветов или эффекты свечения при нажатии клавиш могут сделать приложение более привлекательным и интуитивно понятным.

2. Добавьте настройки

Расширение возможностей настройки, таких как регулировка темпа и выбор типа звука (например, акустическое пианино, орган или синтезатор) делают приложение универсальным. Реализация более совершенных алгоритмов генерации звука позволит пользователям создавать уникальные мелодии и звуковые композиции.

3. Поддержка MIDI

Введение поддержки MIDI-сообщений откроет возможность подключения внешних MIDI-устройств, таких как цифровые пианино и клавиатуры. Это позволяет использовать приложение в профессиональных или образовательных целях, расширяя целевую аудиторию.

Реализация и использование

Разработанное приложение может служить инструментом для обучения игре на фортепиано или музыкальным тренажером для собственной практики. Будет полезно как начинающим музыкантам, так и учителям музыки.

В образовательном секторе приложение можно интегрировать в платформы музыкального образования. Его также можно использовать в качестве примера на курсах по программированию мультимедийных приложений и работе с такими библиотеками, как PyQt5 и Pygame.

Приложение также может быть частью более крупных проектов, таких как: Музыкальные студии или игры, требующие интеграции с инструментами виртуального исполнения музыки.

## **Список использованных источников и приложения**

Список использованных источников

1. Руководство по PyQt5 автор Vasile Buldumac <https://python-scripts.com/pyqt5?ysclid=m4broe3eh3344615423>
2. Превращаем клавиатуру в пианино с помощью Python автор Daiwery <https://habr.com/ru/articles/553032/>
3. What Does if name == "main" Do in Python? автор Martin Breuss <https://realpython.com/if-name-main-python/>
4. Пример на который опирались при создании проекта <https://www.musicca.com/ru/pianino>
5. Второй вариант реализации проекта <https://pianizator.ru/piano-tutorial/1771-izvestnye-melodii-carol-of-the-bells-kak-igrat-na-pianino.html>
6. Библиотека Pygame / Часть 1. Введение автор secretplayer <https://habr.com/ru/articles/588605/>
7. Make-Piano-With-pyQT5 автор Ankur960 <https://github.com/Ankur960/Make-Piano-With-pyQT5?ysclid=m4brtuo0ye465847256>
8. Build a GUI Calculator With PyQt and Python автор Darren Jones <https://realpython.com/courses/build-gui-calculator-pyqt/>
9. Python GUI: создаём простое приложение с PyQt и Qt Designer автор Никита Прияцелюк <https://tproger.ru/translations/python-gui-pyqt?ysclid=m4bskw6db7186473768>
10. Как научиться писать на PyQt? Автор BalaganChick <https://pikabu.ru/story/kak_nauchitsya_pisat_na_pyqt_7753163?ysclid=m4bsln5hqe761680289>
11. Руководство по PyQt5 автор Ilyaragalin <https://pythonist.ru/rukovodstvo-po-pyqt5/?ysclid=m4bsmc7f11354138317>
12. [Как пользоваться PyQt автор Иван Душенко https://pythoninfo.ru/osnovy/pyqt?ysclid=m4bsnoy1df608269152](file:///C:\Users\89523\OneDrive\Рабочий%20стол\Как пользоваться PyQt автор Иван Душенко%20https:\pythoninfo.ru\osnovy\pyqt%3fysclid=m4bsnoy1df608269152)
13. Play Musical Tablature in Python автор Bartosz Zaczyński <https://realpython.com/python-guitar-synthesizer/>
14. Python Audio Synthesis автор Mehvish Ashiq <https://www.delftstack.com/howto/python/python-audio-synthesis/>
15. Основа для записи звука нот <https://www.musicca.com/ru/poisk-not>
16. PythonMIDISynth автор Wells Santo <https://github.com/Devking/PythonMIDISynth?ysclid=m4btdxl9v4571831984>
17. Adding a Background image with PyQt5 автор не указан <https://stackoverflow.com/questions/62043038/adding-a-background-image-with-pyqt5>
18. Python Programming Guides and Tutorials [https://www.pythoncentral.io/](%20https://www.pythoncentral.io/)
19. Simple DirectMedia Layer 3.0 автор не указан <https://wiki.libsdl.org/FrontPage>

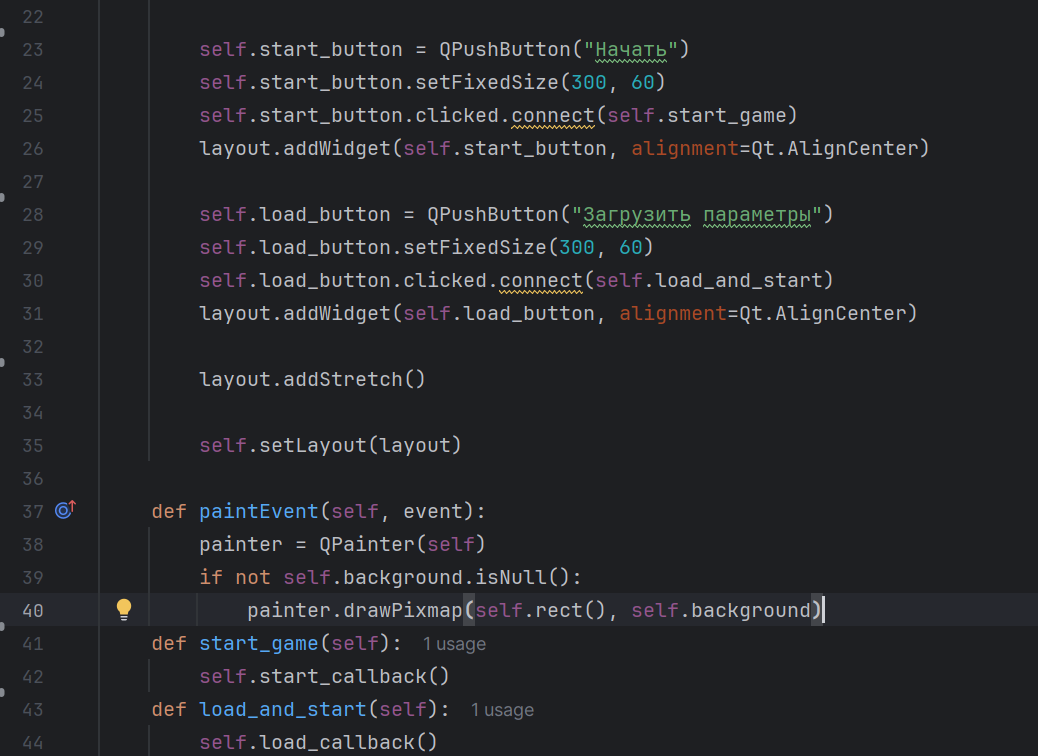
**ПРИЛОЖЕНИЕ 1** Исходный код программы.

Изображение выглядит как снимок экрана, текст, Шрифт

Автоматически созданное описание

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

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Автоматически созданное описание

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

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

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

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

**ПРИЛОЖЕНИЕ 2** Дополнительно используемые файлы

**Звуковые файлы**

[..\..\PycharmProjectsDis\pythonProjectkyrsach](C:\\Users\\89523\\PycharmProjectsDis\\pythonProjectkyrsach)

**Изображение выглядит как снимок экрана

Автоматически созданное описание**

**Графические ресурсы**

[..\..\PycharmProjectsDis\pythonProjectkyrsach\background.png](file:///C:\Users\89523\PycharmProjectsDis\pythonProjectkyrsach\background.png)

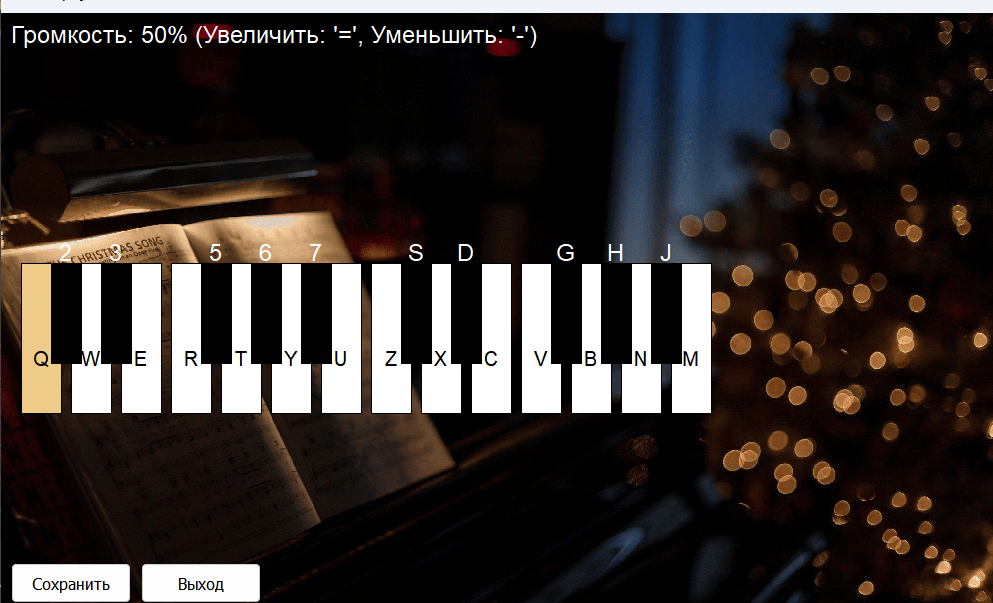
Изображение выглядит как снимок экрана, рождественская елка

Автоматически созданное описание

**ПРИЛОЖЕНИЕ 3** Тестирование

Таблица 1 тестирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | Название теста | Шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
| 1 | Тест клавиши "Q" | Нажать клавишу "Q". | Воспроизводиться звук "1.wav". | Воспроизводиться звук | Пройден |
| 2 | Тест изменения громкости  "+" | Нажать "=" 5 раз | Громкость увеличивается до 100%. | Громкость достигает 100%. | Пройден |
| 3 | Тест загрузки настройки | Сохранить громкость 80%, перезапустить. | Громкость при загрузке — 80%. | Громкость 80%. | Пройден |
| 4 | Тест нажатия неактивной клавиши | Нажать клавишу без звука (например, "Ъ"). | Звук не воспроизводится | Звук отсутствует | Пройден |



скриншот 2 Тест клавиши «Q"

Изображение выглядит как музыкальный инструмент, Музыкальная клавиатура, снимок экрана, клавиатура

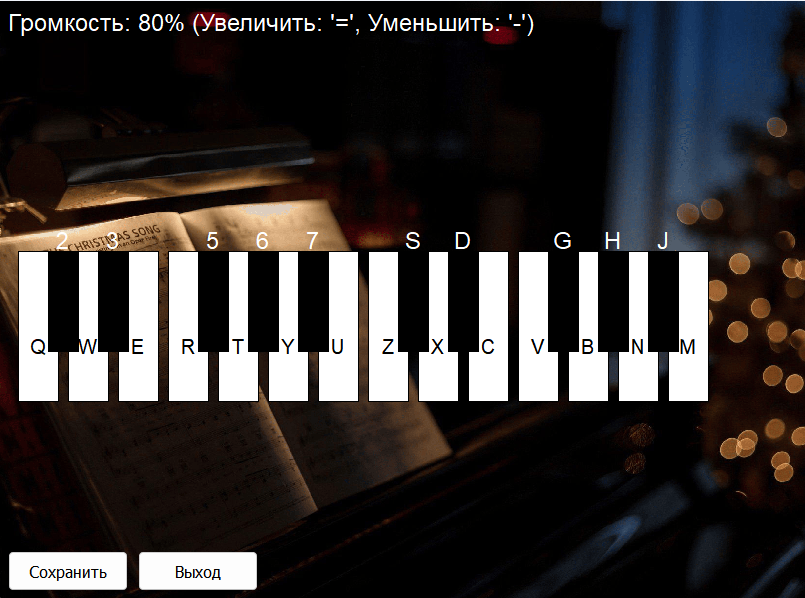
Автоматически созданное описание

скриншот 3 Тест изменения громкости

Изображение выглядит как снимок экрана, рождественская елка, рождество, в помещении

Автоматически созданное описание

скриншот 4 Тест загрузки настройки



скриншот 5 Тест загрузки настройки

Изображение выглядит как пианино, рождественская елка, снимок экрана, Музыкальная клавиатура

Автоматически созданное описание

скриншот 6 Тест нажатия неактивной клавиши

**ПРИЛОЖЕНИЕ 4** Блок-схема логической работы программы

Изображение выглядит как текст, диаграмма, Технический чертеж, линия

Автоматически созданное описание

# ПРИЛОЖЕНИЕ 5 Отзыв на курсовой проект

**ОТЗЫВ**

**научного руководителя на курсовой проект**

**Студента(ки**) Плюсниной Анастасии Олеговны

(фамилия, имя и отчество)

**группа**  302ИС-22 **курс**  3

**Дисциплина**  МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

**Тема работы** Разработка приложения «Симулятор музыкального синтезатора»

Отмечаются следующие моменты: актуальность темы исследования; степень разработанности проблемы, наиболее интересно исследованные вопросы. Оценивается степень самостоятельности и творчества студента; уровень его теоретической подготовки; умение анализировать научные материалы, делать практические выводы. Знание основных концепций, научной и специальной литературы по избранной теме.

**Научный руководитель**

(Уч. степень, уч. звание, должность, ФИО)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г

**ПРИЛОЖЕНИЕ 6** прохождение антиплагиат

