

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Рязанский государственный радиотехнический университет
имени В.Ф. Уткина»
Кафедра «Вычислительная и прикладная математика»

Научно-исследовательская работа

Исследование методов и разработка ПО для генерации визуального
представления геометрических фигур по текстовому контенту

Выполнили:
студенты группы 143
Попов К.И.
Титов А.М.
Ярошенко А.М.
Проверил:
проф. каф. ВПМ
Пылькин А.Н.

Рязань 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналогии программного обеспечения	4
2 Ход работы	8
3 Архитектура программного обеспечения	9
4 Реализация программного обеспечения.....	13
5 Пример работы программного обеспечения.....	17
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21

ВВЕДЕНИЕ

Актуальность данной исследовательской работы определяется необходимостью в разработке эффективных инструментов для визуализации геометрических фигур на основе текстового контента. В современном мире объем данных постоянно растет, и важно иметь возможность извлекать информацию из текстовых источников и представлять её в доступной и понятной форме.

В контексте образования, разработка эффективных инструментов для визуализации геометрических фигур на основе текстового контента имеет огромное значение. Современные учебные программы все больше ориентированы на использование информационных технологий и цифровых ресурсов для обучения. Однако, многие студенты сталкиваются с трудностями в понимании абстрактных математических концепций, таких как геометрические фигуры.

Такие инструменты визуализации могут значительно облегчить процесс обучения, делая математические концепции более доступными и понятными. Студентам будет проще представить себе форму и структуру различных геометрических объектов, что способствует более глубокому усвоению материала и развитию математических навыков.

Более того, такие инструменты могут быть полезны в обучении студентов с особыми образовательными потребностями или ограниченными возможностями в визуализации абстрактных понятий. Представление геометрических фигур визуально по текстовому описанию может сделать материал более доступным и усваиваемым для этой категории студентов.

Таким образом, разработка программного обеспечения для визуализации геометрических фигур по текстовому контенту в образовании может значительно улучшить качество обучения и повысить успеваемость студентов в математике.

1 Аналоги программного обеспечения

Среди аналогов программного обеспечения для генерации визуального представления геометрических фигур по текстовому контенту были найдены несколько популярных инструментов. Вот некоторые из них:

Desmos Geometry – это бесплатный веб-инструмент, который позволяет создавать и визуализировать геометрические фигуры, а также выполнять математические операции с ними. Программа предоставляет простой в использовании интерфейс с широким набором инструментов для создания точек, линий, углов, окружностей и других геометрических объектов. Desmos Geometry также поддерживает построение графиков функций и предоставляет возможность создавать интерактивные задачи и упражнения для обучения геометрии. (<https://www.desmos.com/geometry?lang=ru>)

GeoGebra Geometry – это инструмент для создания и исследования геометрических фигур. Программа обладает обширным функционалом, включая возможность построения точек, линий, многоугольников, окружностей, а также проведение измерений, вычисление площадей и объемов, решение геометрических задач и многое другое. GeoGebra Geometry также позволяет создавать динамические документы и интерактивные приложения для обучения и исследования. (<https://www.geogebra.org/geometry>)

Mathcad – это программное обеспечение для решения инженерных и научных задач, которое сочетает в себе возможности символьных вычислений с простотой и удобством использования табличного интерфейса. Программа позволяет создавать математические документы, в которых можно проводить вычисления, вводить математические формулы, рисовать графики и визуализировать данные. Mathcad также поддерживает работу с геометрическими фигурами и предоставляет инструменты для создания графиков и рисунков, включая возможность импорта графических объектов из внешних приложений.

Сравнительный анализ аналогов приведен в таблице 1.

Таблица 1 – Сравнительная таблица аналогов

<div> <div>Наименование</div> <div>Характеристика</div> </div>	Desmos	Geogebra	Mathcad
Возможность задания фигур уравнением	Присутствует	Присутствует	Присутствует
Точность воспроизведения	Точность произведения зависит от того, правильно ли было задано уравнение фигуры	Точность произведения зависит от того, правильно ли было задано уравнение фигуры	Точность произведения зависит от того, правильно ли было задано уравнение фигуры
Возможность задания фигуры текстовым описанием	Отсутствует	Отсутствует	Отсутствует
Функциональность	Предоставляет основные функции для создания графиков и некоторых геометрических фигур.	Обладает обширным набором функций для создания и исследования геометрических объектов, включая	Специализируется на символьных вычислениях и математических операциях, поэтому его функциональность может

		поддержку различных геометрических фигур и аналитические возможности.	быть ограничена для работы с геометрическими фигурами.
Доступность и удобство использования	Обладает простым и интуитивно понятным интерфейсом, который делает его легким в использовании для широкого круга пользователей.	Имеет удобный интерфейс с множеством инструментов, но может быть слегка сложнее для новичков из-за большого количества функций.	Имеет специфический интерфейс, который может быть непривычным для пользователей, не знакомых с системами символьных вычислений.
Платформа	Предоставляет как веб-версию, так и настольные и мобильные приложения для работы на различных платформах.	Предоставляет как веб-версию, так и настольные и мобильные приложения для работы на различных платформах.	Настольное приложение, доступное для установки на компьютеры под управлением операционных систем Windows.

Использование ИИ в анализе данных	Не предоставляет специализированных инструментов для анализа данных с использованием искусственного интеллекта	Некоторые версии GeoGebra могут предоставлять некоторые возможности анализа данных, но их использование ИИ в этом контексте ограничено.	Mathcad не является специализированным инструментом для анализа данных с использованием ИИ, но может быть использован для проведения различных математических и статистических вычислений.
-----------------------------------	--	---	--

2 Ход работы

Для реализации данного проекта требуется следующий функционал:

1. Разработка интерфейса:
 - Разработка пользовательского интерфейса, который будет простым в использовании и понимании для широкого круга пользователей.
 - Включение элементов управления, инструкций и подсказок для облегчения навигации и взаимодействия пользователя с приложением.
2. Распознавание текста геометрических задач:
 - Использование алгоритмов библиотеки OpenCV для распознавания текста задач с изображений.
 - Использование алгоритмов библиотеки Speech Recognition для распознавания голоса и перевода голосовой записи в текст.
3. Анализ геометрических задач при помощи машинного обучения:
 - Использование моделей машинного обучения для анализа геометрических задач и принятия решений на основе извлеченных параметров и условий.
4. Построение геометрических фигур по описанию:
 - Разработка алгоритмов, которые преобразуют текстовое описание геометрических фигур в математические модели и геометрические объекты.
 - Использование полученных моделей и алгоритмов для создания визуального представления геометрических фигур на основе текстовых описаний.

Каждая из этих задач требует интеграции соответствующих методов и технологий, таких как алгоритмы обработки естественного языка, модели машинного обучения, инструменты для визуализации данных и создания пользовательского интерфейса.

3 Архитектура программного обеспечения

На рисунке 1 приведена предполагаемая архитектура приложения для генерации геометрического контента. Предполагается, что архитектура реализации будет клиент-серверной.

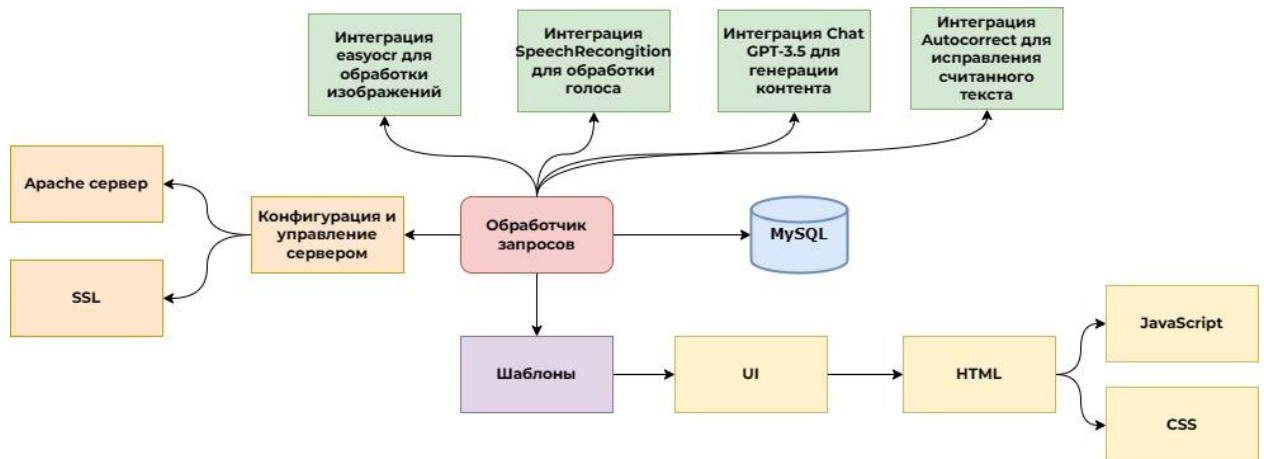


Рисунок 1 – Архитектура разрабатываемого приложения

Обработчик запросов (back-end):

- Этот компонент будет отвечать за обработку HTTP-запросов от клиентов.
- Включит в себя маршрутизацию запросов, обработку данных и взаимодействие с базой данных.

Интеграция easyocr для обработки изображений:

- Этот компонент будет ответственен за обработку изображений с текстом геометрических задач, которые могут поступать в качестве входных данных.
- Включит в себя функции и классы для работы с изображениями, предоставляемые easyocr, такие как загрузка, обработка, анализ и распознавание текста.
- Этот компонент может быть использован для извлечения признаков из изображений, а также для подготовки данных перед обучением моделей машинного обучения или после их обработки.

Интеграция SpeechRecognition:

- Этот компонент будет ответственен за обработку голосового ввода, распознавание и перевод данных в текстовый формат.
- Включит в себя функции и классы для работы с изображениями, предоставляемые SpeechRecognition, такие как загрузка, обработка, анализ и распознавание голоса.

Интеграция Chat GPT-3.5:

- Этот компонент будет ответственен за генерацию геометрического контента по извлеченной и переведенной в текст информации.
- Включит в себя все возможности искусственного интеллекта, предлагаемые данным инструментом.

Интеграция Autocorrect:

- Этот компонент будет ответственен за обработку считанного текста либо посредством обработки изображения, либо посредством обработки голосового ввода.
- Компонент будет ответственен за проверку на грамотность текстового запроса в соответствии с нормами русского языка и возможного исправления в случае, если представленные выше сервисы ИИ недостаточно корректно считали входную информацию.
- Включит в себя функции и классы для работы с изображениями, предоставляемые Autocorrect, такие как автоматическое исправление текста.

База данных (MySQL):

- Приложение будет использовать MySQL для хранения данных о геометрических фигурах.
- Будет содержать таблицы для хранения информации о фигурах, их описаниях и других связанных данных.

Шаблоны (представление):

- Этот компонент будет содержать шаблоны HTML для форматирования и представления данных, возвращаемых пользователю.

- Может быть интегрирован с обработчиком запросов для формирования HTML-страниц на лету или использовать статические шаблоны.

Apache сервер:

- Apache принимает HTTP-запросы от клиентов (браузеров) и обрабатывает их в соответствии с настройками виртуальных хостов и правилами маршрутизации.

- Доступ к статическим ресурсам, таким как HTML-страницы, изображения, CSS-файлы и JavaScript-файлы, хранящиеся на сервере.

- Виртуальные хосты и правила перенаправления (Rewrite Rules) в конфигурации Apache позволяют маршрутизировать запросы к соответствующим обработчикам приложения.

SSL (Secure Sockets Layer):

- SSL обеспечивает шифрование данных, передаваемых между клиентом и сервером, что предотвращает возможность перехвата и прослушивания конфиденциальной информации.

- SSL-сертификаты используются для аутентификации сервера, подтверждая его идентичность перед клиентами и предотвращая атаки, связанные с подменой сервера.

- SSL также обеспечивает проверку целостности данных, гарантируя, что переданные данные не были изменены в процессе передачи.

Прототип пользовательского интерфейса приложения представлен на рисунке 2.

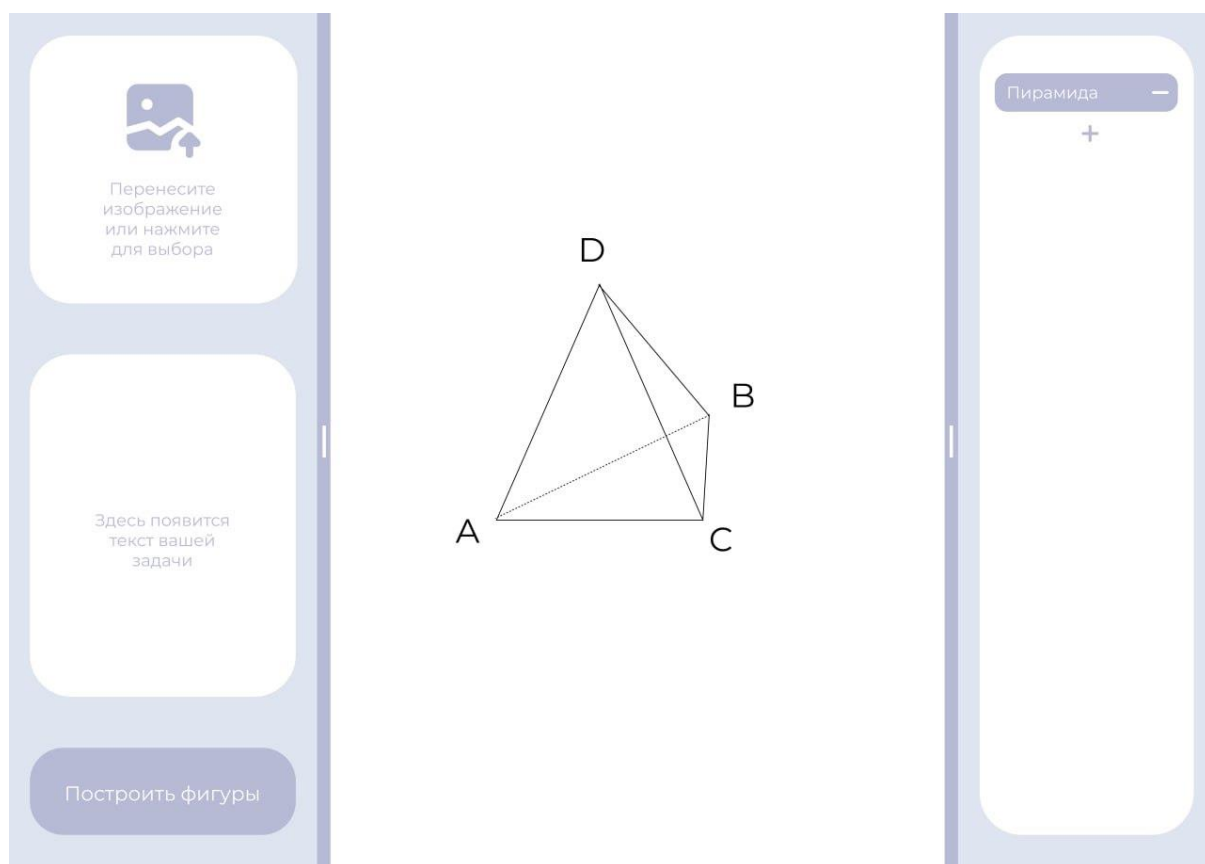


Рисунок 2 – Прототип интерфейса

4 Реализация программного обеспечения

Приведем алгоритмы взаимодействия с методами библиотек искусственного интеллекта. Языком реализации является Python, поэтому листинги кода будут приведены на нем.

Первым вариантом задания начальных данных является ввод текста вручную.

Вторым вариантом задания начальных данных является считывание изображения. Алгоритм считывания текста с изображения при помощи библиотеки `easyocr` представлен ниже:

```
def text_recognition(self):  
    file_name = self.task_path  
    reader = easyocr.Reader(["ru", "en"])  
    result = reader.readtext(file_name, detail=0, paragraph=True)  
    result = '\n'.join(result)  
    self.textEdit.setText(result)
```

Блок-схема алгоритма представлена на рисунке 3.



Рисунок 3 – Алгоритм считывания текста с изображения

Третьим вариантом задания начальных данных является голосовой ввод. Алгоритм считывания голосового ввода и перевода в текстовую информацию приведен ниже:

```
def voice_assistant(self):
```

```

self.label_voice.setText("Идет запись...")
def audio_to_text():
    with sr.Microphone() as microphone:
        recognizer = sr.Recognizer()
        recognizer.adjust_for_ambient_noise(microphone, 2)
        try:
            audio = recognizer.listen(microphone, 5, 5)
            self.label_voice.setText("Запись завершена!")
            return recognizer.recognize_google(audio, language="ru-RU")
        except:
            QMessageBox.critical(None, "Ошибка", "Не удалось распознать
текст")
self.textEdit.setText(audio_to_text())

```

Блок-схема алгоритма представлена на рисунке 4.



Рисунок 4 – Алгоритм считывания голосового ввода

После получение обработанного входного задания, необходимо сгенерировать геометрический контент. Алгоритм генерации геометрического контента при помощи Chat GPT-3.5 приведен ниже:

```

def create_request(self):
    chat_completion = self.client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": self.begin_of_request +
self.textEdit.toPlainText()}]
    )
    return chat_completion.choices[0].message.content

```

Блок-схема алгоритма приведена на рисунке 5.

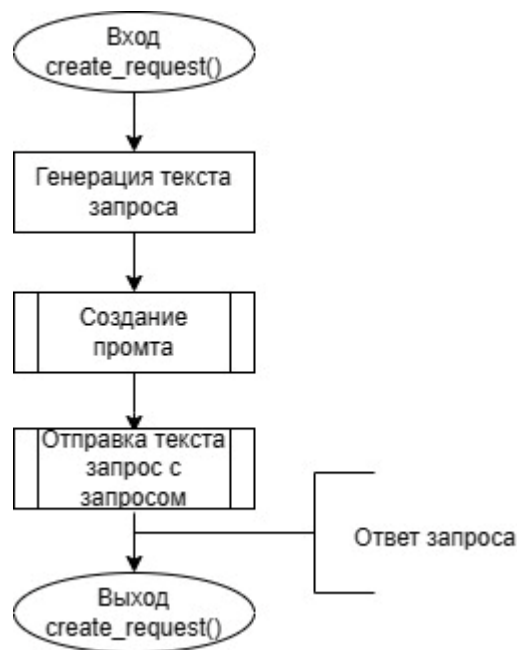


Рисунок 5 – Алгоритм генерации геометрического контента

Для представления графически полученного контента необходимо разработать парсер ответа искусственного интеллекта. В основе парсинга будут лежать возможности регулярных выражений. Алгоритм парсинга данных приведен ниже:

```
def parse_shape_and_coordinates(self, string):
    shape_match = re.search(r'Имя фигуры: (\w+)', string)
    if shape_match:
        shape = shape_match.group(1)
    else:
        shape = "Неизвестная фигура"
    coordinates_match = re.search(r'Массив координат: x = \[(.*?)\] y = \[(.*?)\]', string) or re.search(r'Массив координат: x = \[(.*?)\], y = \[(.*?)\]', string)
    if coordinates_match:
        try:
            x_coordinates = list(map(int, coordinates_match.group(1).split(',')))
            y_coordinates = list(map(int, coordinates_match.group(2).split(',')))
        except:
            QMessageBox.critical(None, "Ошибка", "Не удалось построить фигуру")
    else:
        x_coordinates = []
        y_coordinates = []
```

```
QMessageBox.critical(None, "Ошибка", "Не удалось построить фигуру")  
return shape, x_coordinates, y_coordinates
```

Блок-схема алгоритма приведена на рисунке 6.

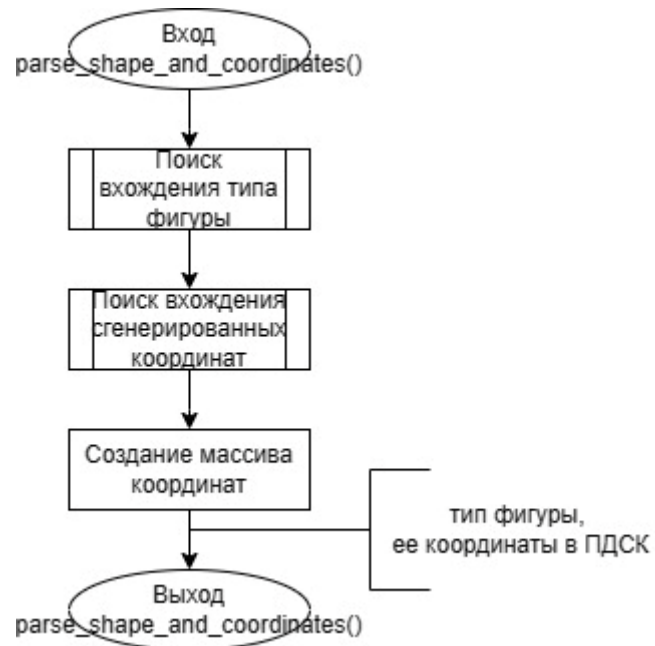


Рисунок 6 – Алгоритм парсера

5 Пример работы программного обеспечения

Разберем один из вариантов работы программного обеспечения – с выбором изображения в качестве исходных данных.

На рисунке 7 представлено начальное состояние интерфейса

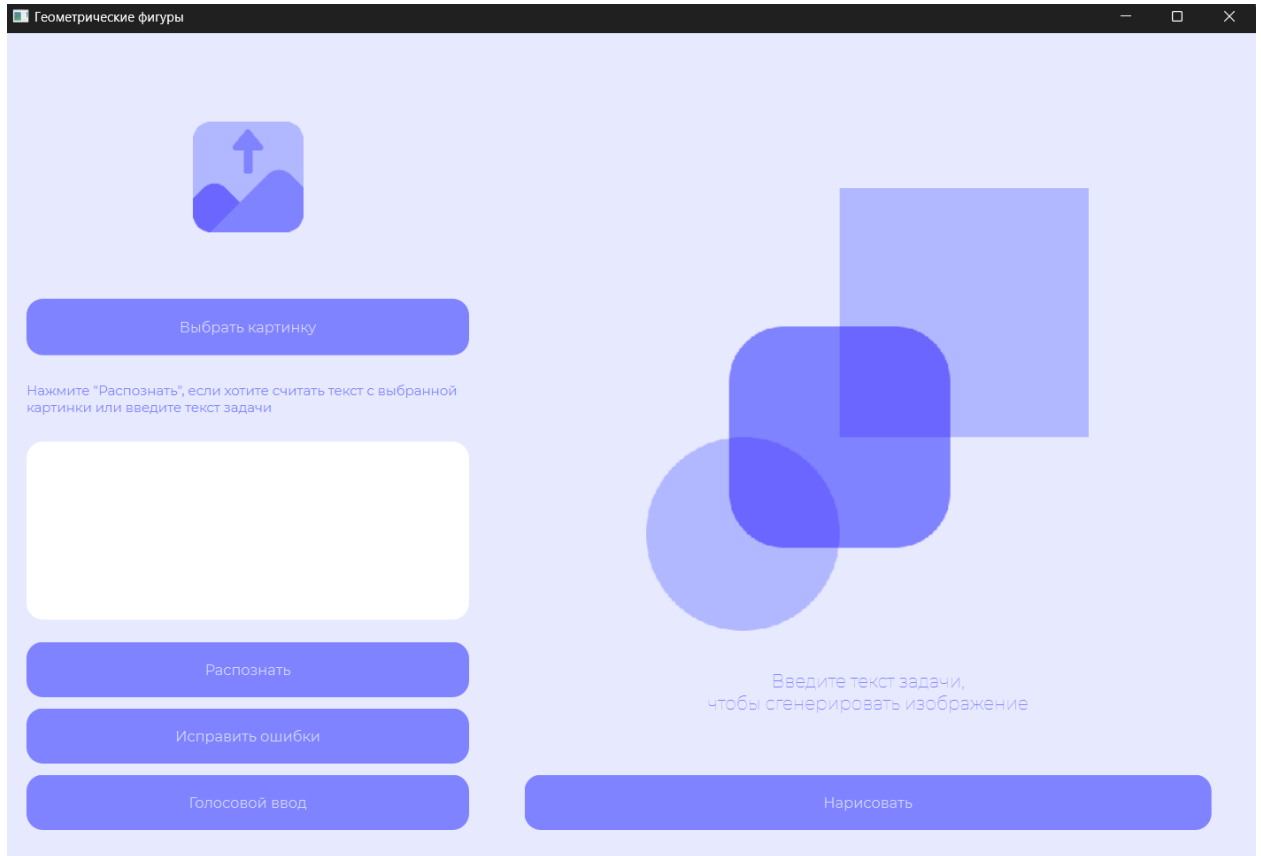


Рисунок 7 – Главное меню приложения

После выбора в диалоговом окне подходящего изображения в поле текста появится ее условие. Состояние окна представлено на рисунке 8.

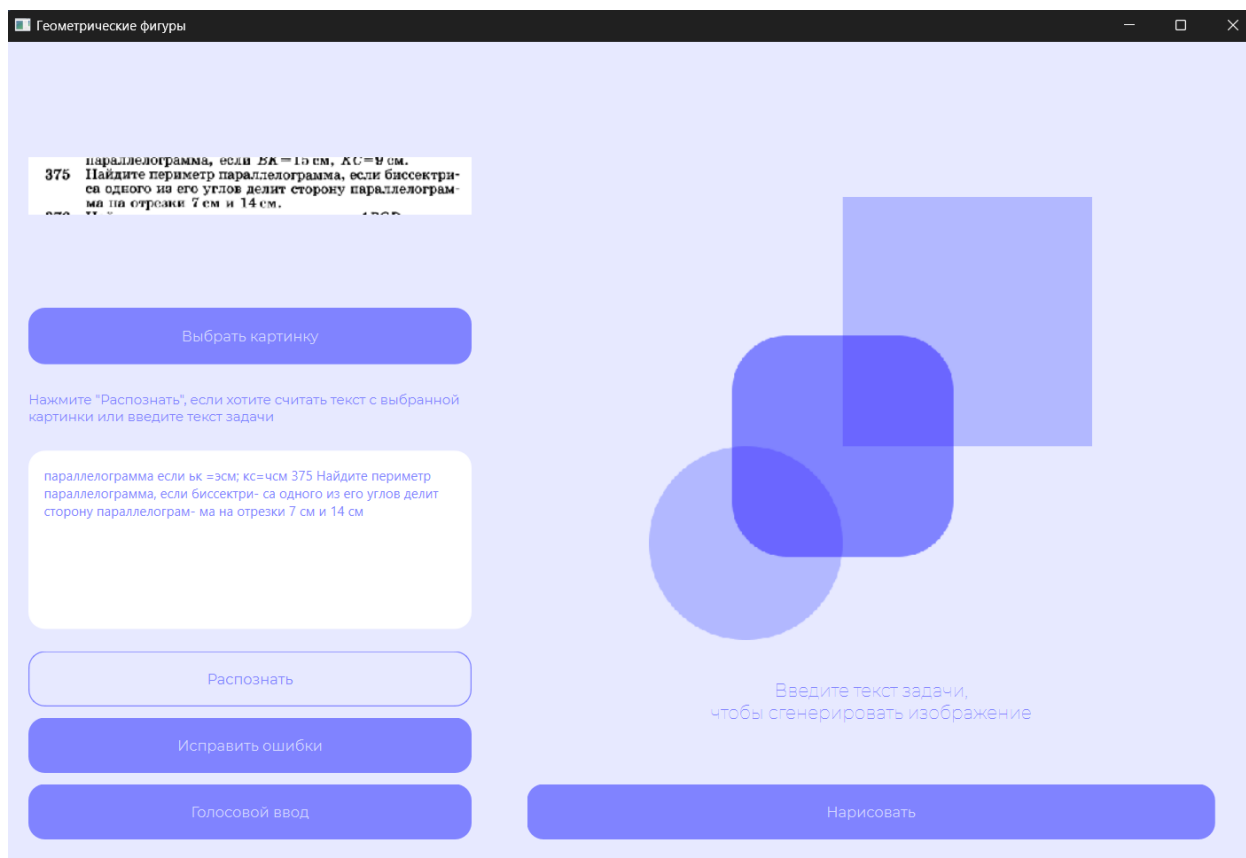


Рисунок 8 – Интерфейс окна с распознанной задачей

Пользователь может удостовериться в верности условия задачи нажатием кнопки «Исправить ошибки». После нажимается кнопка «Нарисовать» и появляется сгенерированная фигура согласно условию задачи. Итоговое состояние окна представлено на рисунке 9.

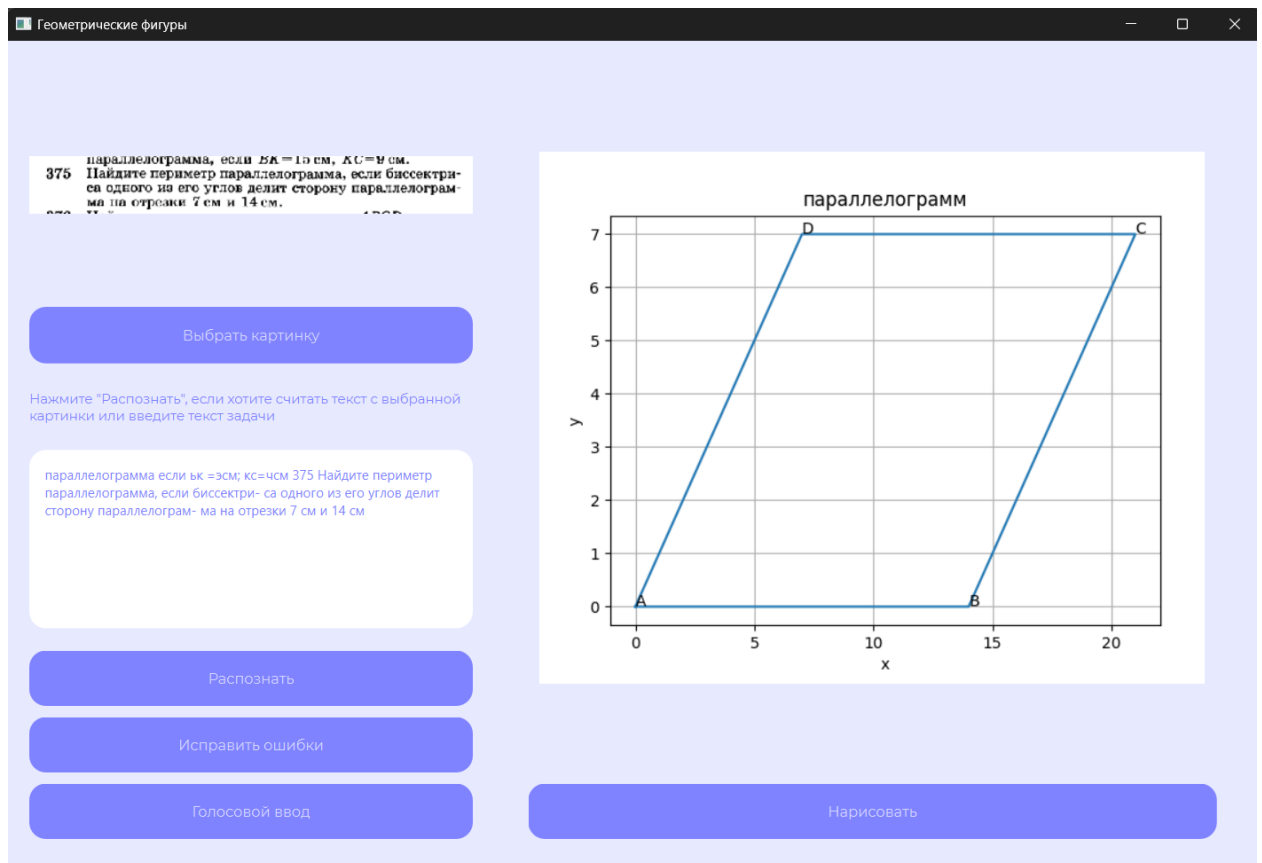


Рисунок 9 – Итоговое состояние окна с полученной фигурой

ЗАКЛЮЧЕНИЕ

В заключении данной исследовательской работы можно подчеркнуть значимость разработки эффективных инструментов визуализации геометрических фигур на основе текстового контента. В современном мире, где объем данных постоянно растет, извлечение информации из текстовых источников и ее представление в доступной форме становится все более важным.

В контексте образования, разработка таких инструментов приобретает особую значимость. Современные учебные программы все больше ориентированы на использование информационных технологий, и инструменты визуализации могут значительно облегчить процесс обучения, особенно в случае абстрактных математических концепций, таких как геометрические фигуры.

Инструменты визуализации помогают студентам лучше понять форму и структуру геометрических объектов, делая математические концепции более доступными и понятными. Они также могут быть полезны для студентов с особыми образовательными потребностями, предоставляя им возможность лучше усвоить материал.

Таким образом, разработка программного обеспечения для визуализации геометрических фигур на основе текстового контента в образовании имеет потенциал существенно улучшить качество обучения и повысить успеваемость студентов в математике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Интернет инструмент построения графиков Desmos [Электронный ресурс] <https://www.desmos.com/>
2. Интернет инструмент построения графиков GeoGebra [Электронный ресурс] <https://www.geogebra.org/>
3. Искусственный интеллект ChatGPT (OpenAI) [Электронный ресурс] <https://openai.com/chatgpt>
4. Библиотека с искусственным интеллектом EasyOCR [Электронный ресурс] <https://github.com/JaidedAI/EasyOCR>
5. Библиотека с искусственным интеллектом Autocorrect [Электронный ресурс] <https://github.com/phatpiglet/autocorrect>
6. Библиотека с искусственным интеллектом SpeechRecognition [Электронный ресурс] <https://pypi.org/project/SpeechRecognition/>