

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

Дипломный проект (работа)

На тему: Разработка приложения для генерации текстовых отчётов на основе анализа скриншотов с помощью компьютерного зрения

ФЕДОРКОВА ФЁДОРА ГЕННАДЬЕВИЧА

(ФИО студента полностью в родительном падеже, прописными (большими) буквами)

Студент 3 курса группы ИС50-11/3-23

по специальности 09.02.07 «Информационные системы и программирование»

для присвоения квалификации: Специалист по информационным системам

Форма обучения: очная

Руководитель: _____ /Прищеп Василика Станиславовна/

(подпись)

« _____ » _____ 2026 г.

Консультант: _____ /Прищеп Михаил Сергеевич/

(подпись)

« _____ » _____ 2026 г.

Студент: _____ /Федорков Фёдор Геннадьевич/

(подпись)

« _____ » _____ 2026 г.

Допущен к защите

Приказ от « _____ » _____ 2026 г. № _____

Содержание

ВВЕДЕНИЕ.....	4
1. ОБЩАЯ ЧАСТЬ	9
1.1. Цели разработки.....	9
1.2. Средства разработки	10
1.2.1. Технические средства	10
1.2.2. Программные средства	11
2. СПЕЦИАЛЬНАЯ ЧАСТЬ.....	14
2.1. Постановка задачи	14
2.1.1. Описание задачи.....	14
2.1.2. Входные и выходные данные	19
2.2. Внешняя спецификация.....	22
2.2.1. Метод.....	22
2.2.2. Тесты	24
2.3. Проектирование.....	26
2.3.1. Схема данных	26
2.3.2. Функциональная схема.....	30
2.3.3. Структурная схема	33
2.3.4. Схема пользовательского интерфейса	34
2.3.5. Укрупнённый алгоритм	35
2.3.6. Блок-схема	36
2.4. Результаты работы программы.....	39
3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	45
3.1. Инструментальные средства разработки.....	45

3.2. Отладка программы	47
3.3. Защитное программирование	49
3.4. Характеристики программы.....	51
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ....	58
ПРИЛОЖЕНИЯ.....	61

ВВЕДЕНИЕ

В последние годы наблюдается стремительная цифровизация процессов подготовки технической и учебной документации. Развитие методов компьютерного зрения и появление мультимодальных моделей машинного обучения коренным образом изменили подходы к созданию пояснительных записок, инструкций, методических материалов и руководств пользователя. Современные программные средства позволяют автоматически выделять значимые области на снимках экрана, формировать связные текстовые описания произвольных изображений и преобразовывать визуальный контент в структурированные документы.

Подготовка отчётной документации сопровождается большим количеством иллюстраций - снимков пользовательского интерфейса с поясняющими подписями. Каждый такой снимок необходимо снабдить связным описанием на русском языке и оформить по требованиям ГОСТ: вставить в документ с центрированием, подписать в формате «Рисунок N - наименование», поддерживать сквозную нумерацию рисунков. Развитие мультимодальных языковых моделей и средств автоматизации Microsoft Word через интерфейс COM делает возможной полную автоматизацию этой рутинной операции - от захвата снимка экрана до получения готового абзаца с описанием и подписью внутри открытого документа Word.

Актуальность выбранной темы обусловлена возрастающим спросом на инструменты автоматизированной подготовки технической и учебной документации, а также появлением мультимодальных языковых моделей, делающих технически реализуемым автоматическое описание содержимого снимков экрана с учётом контекста. Разрабатываемое в

рамках настоящей выпускной квалификационной работы приложение позволяет на практике отработать перечисленные задачи и получить полнофункциональное решение для повседневной подготовки отчётов с большим количеством иллюстраций.

Современный рынок программных средств захвата экрана представлен рядом коммерческих и свободных решений. Анализ существующих продуктов позволяет выявить их сильные и слабые стороны и обосновать проектные решения, принятые в рамках настоящей работы.

Первым решением является штатная утилита Microsoft Snipping Tool, входящая в состав операционной системы Windows и вызываемая комбинацией клавиш Win + Shift + S. К сильным сторонам относятся бесплатное распространение в составе системы, мгновенный запуск и поддержка четырёх режимов выделения области экрана. Слабыми сторонами являются полное отсутствие средств автоматического описания снимков экрана, невозможность прямой вставки изображения в открытый документ Microsoft Word с подписью «Рисунок N - наименование» и отсутствие сквозной нумерации иллюстраций - все перечисленные операции пользователю приходится выполнять вручную.

Вторым конкурентом выступает свободное настольное приложение Greenshot, распространяемое по лицензии GNU GPL. Преимуществами являются простой и интуитивно понятный интерфейс, низкое потребление системных ресурсов и поддержка экспорта в распространённые графические форматы. Однако функциональные возможности программы ограничены работой со статичными изображениями: отсутствуют средства автоматического описания снимков на основе языковых моделей и подсистема прямой вставки

иллюстраций в документ Microsoft Word с автоматической нумерацией и оформлением по требованиям ГОСТ.

Третьим решением является свободная утилита Lightshot, ориентированная на быстрое создание снимков экрана с возможностью облачной публикации полученных изображений. К сильным сторонам относятся минимальный размер дистрибутива, поддержка горячей клавиши и базовый редактор аннотаций. Ограничения заключаются в отсутствии средств автоматического описания снимков, невозможности интеграции с Microsoft Word и обязательной зависимости функций обмена от облачной инфраструктуры разработчика.

Четвёртым рассмотренным решением является облачный сервис Loom, ориентированный преимущественно на запись видеосообщений с экрана. К сильным сторонам относятся встроенная нейросетевая транскрипция звуковой дорожки и удобный интерфейс публикации. Ограничения заключаются в полной зависимости от облачной инфраструктуры, отсутствии режима офлайн-работы и невозможности прямой работы с документами Microsoft Word при подготовке технической документации.

Сравнительный анализ показал, что все рассмотренные продукты успешно решают подзадачу захвата изображения с экрана, однако ни одно из готовых решений не обеспечивает целостного процесса - от захвата снимка экрана до получения внутри открытого документа Microsoft Word абзаца с пояснительным текстом, изображения с центрированием и подписи «Рисунок N - наименование», сформированных автоматически на основе мультимодальной языковой модели и оформленных по требованиям ГОСТ. В этой связи разработка собственного компактного программного продукта представляет практический интерес.

Целью данной работы является разработка десктопного приложения для операционной системы Windows, обеспечивающего быстрый захват снимков экрана, автоматическое формирование текстовых описаний полученных изображений на основе мультимодальной языковой модели и автоматическую вставку результата в открытый документ Microsoft Word с подписью «Рисунок N - наименование» и оформлением по требованиям ГОСТ. Дополнительно приложение содержит встроенный редактор аннотаций и ряд вспомогательных подсистем, расширяющих сценарии его повседневного использования.

Для достижения поставленной цели в ходе работы были решены следующие задачи:

- анализ предметной области и формирование функциональных требований к приложению;
- проектирование модульной архитектуры программного комплекса на платформе .NET Framework;
- реализация модуля быстрого захвата снимков экрана с интеллектуальным выделением областей на основе оператора Собеля и алгоритма заливки по цветовому сходству;
- интеграция с мультимодальной языковой моделью Google Gemini для автоматического формирования текстовых описаний снимков экрана с учётом контекста;
- разработка подсистемы автоматической вставки полученного описания, изображения и подписи «Рисунок N - наименование» в открытый документ Microsoft Word через интерфейс COM-автоматизации с оформлением по требованиям ГОСТ;

- реализация встроенного графического редактора аннотаций со стрелками Безье, подсветками, маркерами, текстовыми блоками и эффектами размытия в качестве дополнительного функционала;
- реализация вспомогательных подсистем (запись экранного видео, автоматическая коррекция русскоязычного текста, поиск файлов на локальном носителе и управление буфером обмена) в качестве расширения базового сценария;
- тестирование корректности работы приложения и анализ результатов.

Практическая значимость работы заключается в том, что разработанное программное обеспечение может быть непосредственно использовано техническими писателями, преподавателями, аналитиками и инженерами-разработчиками для повседневной подготовки отчётной документации с большим количеством иллюстраций. Применение приложения позволяет существенно сократить время оформления пояснительных записок, повысить качество и единообразие подписей к иллюстрациям и снизить количество ошибок ручного ввода описаний к снимкам экрана.

1. ОБЩАЯ ЧАСТЬ

1.1. Цели разработки

Целью разработки настольного приложения для генерации текстовых описаний снимков экрана и их автоматической вставки в открытый документ Microsoft Word является создание программного инструмента, ускоряющего повседневную подготовку технической и учебной документации с большим количеством иллюстраций. Разработка данного приложения позволяет упростить взаимодействие пользователя с собранным визуальным материалом: пользователь делает снимок экрана глобальной горячей клавишей, мультимодальная языковая модель Google Gemini автоматически формирует связное описание изображения, а полученное описание вместе с самим снимком и подписью «Рисунок N - наименование» автоматически вставляется в открытый документ Microsoft Word с центрированием иллюстрации и оформлением по требованиям ГОСТ. Дополнительно приложение содержит встроенный редактор аннотаций и ряд вспомогательных подсистем, расширяющих сценарии его повседневного использования.

Актуальность работы заключается в устойчивом росте объёмов технической и учебной документации и повышении требований к качеству и единообразию оформления иллюстраций. На данный момент существует ограниченное количество решений, обеспечивающих сквозной процесс - от захвата снимка экрана до получения внутри открытого документа Microsoft Word абзаца с описанием, изображения и подписи «Рисунок N - наименование», сформированных автоматически на основе мультимодальной языковой модели, - что создаёт необходимость в разработке современного, удобного и функционально полноценного программного продукта.

1.2. Средства разработки

1.2.1. Технические средства

Технические средства проектирования и тестирования приложения - это аппаратное обеспечение, на котором выполняется работа над программным продуктом. От характеристик технических средств зависят возможности отладки, корректность работы при произвольных параметрах разрешения экрана и удобство проверки приложения при разных коэффициентах масштабирования системы.

В таблице 1 указаны технические характеристики портативного компьютера.

Таблица 1 - Технические характеристики

Наименование	Описание
Процессор	Intel Core i5-12450H, 8 ядер, 12 потоков, базовая частота 2,0 ГГц
Видеокарта	NVIDIA GeForce RTX 3050, 4 ГБ GDDR6
Оперативная память	DDR4 16 ГБ, 3200 МГц
Накопитель	SSD NVMe 512 ГБ
Дисплей	15,6 дюйма, разрешение 1920 × 1080 пикселей, частота 144 Гц
Сетевой адаптер	Wi-Fi 6 (802.11ax), Bluetooth 5.2
Операционная система	Microsoft Windows 10 Pro x64, версия 22H2

1.2.2. Программные средства

Для проектирования приложения определены языки программирования и программные средства, на основе которых ведётся работа над продуктом. Используемые языки программирования представлены в таблице 2.

Таблица 2 - Языки программирования

Название	Описание	Назначение
C#	Объектно-ориентированный язык программирования общего назначения, разработанный компанией Microsoft. Предоставляет строгую типизацию, безопасное управление памятью и поддержку асинхронного программирования.	При разработке данного проекта используется в качестве основного языка реализации настольного приложения - модуля захвата экрана, графического редактора аннотаций, подсистем интеграции с Microsoft Word и Google Gemini, модуля автоматической коррекции текста и подсистемы экспорта документов.
JavaScript	Кроссплатформенный язык программирования с динамической типизацией, разработанный компанией Netscape. Является стандартным языком сценариев браузера, поддерживает работу с HTML5-элементами video и audio.	При разработке данного проекта используется во встроенных HTML-страницах подсистемы воспроизведения медиа: реализует управление видео- и аудиоэлементами и обмен данными с управляющим C#-кодом через объектную модель элемента WebBrowser.
Python	Высокоуровневый интерпретируемый язык программирования общего назначения с динамической типизацией. Обладает богатой стандартной библиотекой и широкой экосистемой пакетов для обработки естественного языка.	При разработке данного проекта используется в утилитарных сценариях подготовки данных: генерации словаря словоформ для модуля автоматической коррекции, обработке морфологических ресурсов проекта OpenCorpora и формировании сериализованных файлов фильтра Блума.

PowerShell	Объектно-ориентированный командный интерпретатор и язык сценариев компании Microsoft, тесно интегрированный с платформой .NET. Поддерживает обработку структурированных данных и управление процессами Windows.	При разработке данного проекта используется для автоматизации сборки приложения - запуска компилятора csc.exe с необходимыми ключами, управления каталогами вывода и контроля кода завершения сборки.
HTML	Стандартизированный язык разметки гипертекстовых документов, описывающий структуру и оформление содержимого веб-страниц. Поддерживает встроенные мультимедийные элементы стандарта HTML5.	При разработке данного проекта используется для описания внутренних страниц встроенного медиаплеера, отображаемого через элемент WebBrowser операционной системы Windows.

Конец таблицы 2 - Языки программирования

Помимо языков программирования при проектировании приложения применяется ряд программных средств: целевая платформа выполнения, среда редактирования исходного кода, штатный компилятор, библиотека работы с документами Microsoft Word, облачный программный интерфейс мультимодальной языковой модели и система контроля версий. Перечень используемых программных средств приведён в таблице 3.

Таблица 3 - Программные средства

Название	Описание	Назначение
.NET Framework 4.0	Программная платформа компании Microsoft для разработки и выполнения управляемых приложений в Windows. Включает общезыковую исполняющую среду CLR, библиотеку базовых классов и подсистему Windows Forms.	При разработке данного проекта используется как целевая платформа выполнения. На базе подсистемы Windows Forms построен пользовательский интерфейс, классы пространств имён обеспечивают работу с графикой и сериализацию настроек.

Visual Studio Code	Свободный кроссплатформенный редактор исходного кода компании Microsoft с поддержкой подсветки синтаксиса, автодополнения, отладки и интеграции с системами контроля версий. Расширяется через систему модулей.	При разработке данного проекта используется в качестве основной среды разработки исходного кода: обеспечивает редактирование C#-, JavaScript-, Python- и PowerShell-файлов и запуск сборок через интегрированный терминал.
csc.exe	Штатный командный компилятор языка C#, входящий в состав .NET Framework. Преобразует исходные файлы C# в управляемые сборки формата Portable Executable.	При разработке данного проекта используется для непосредственной сборки исполняемого файла приложения с указанием пакета исходных файлов, ссылок на стандартные сборки .NET и режима unsafe.
Microsoft Word	Текстовый процессор пакета Microsoft Office. Предоставляет программный интерфейс автоматизации на основе технологии COM, позволяющий внешним приложениям управлять документами формата DOCX.	При разработке данного проекта используется в качестве целевого приложения для подсистемы интеграции - через COM приложение вставляет в открытый документ снимки экрана с описанием и подписью «Рисунок N - наименование».
Google Gemini API	Облачный программный интерфейс компании Google для обращения к семейству мультимодальных языковых моделей Gemini. Поддерживает приём изображений и текстовых запросов в формате REST.	При разработке данного проекта используется как удалённая языковая модель для автоматического формирования текстовых описаний снимков экрана: приложение отправляет в API изображение и получает связанное описание.
Open XML SDK	Программный комплект разработки компании Microsoft, обеспечивающий программный доступ к структуре документов Open XML, включая файлы DOCX. Позволяет работать с документами без установленного пакета Office.	При разработке данного проекта используется в подсистеме автономного формирования отчёта: формирует абзацы, вставляет иллюстрации с подписями и сохраняет результат в DOCX по шаблону ГОСТ.

Git	Распределённая система контроля версий с открытым исходным кодом. Поддерживает ветвление, слияние, локальные и удалённые репозитории, фиксацию истории изменений с возможностью отката к произвольной ревизии.	При разработке данного проекта используется для контроля версий исходного кода - каждое значимое изменение фиксируется отдельным коммитом, что обеспечивает воспроизводимую историю разработки.
-----	--	---

Конец таблицы 3 - Программные средства

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

2.1.1. Описание задачи

В процессе проектирования настольного приложения для генерации текстовых описаний снимков экрана и их автоматической вставки в открытый документ Microsoft Word необходимо предусмотреть следующие возможности:

Захват снимков экрана. Данная функция обеспечивает быстрое получение растрового образа произвольной области экрана по нажатию глобальной горячей клавиши. После активации захвата на экране отображается замороженный оверлей с текущим изображением рабочего стола, на котором пользователь выделяет требуемую область по цвету в режиме по умолчанию (оператор Собея и заливка по цветовому сходству), а при зажатии правой кнопки мыши - произвольной прямоугольной рамкой. Координаты выделения и сам растровый снимок передаются в графический редактор приложения. Для расширения выделенной области до целого окна, строки или экрана пользователь использует прокрутку колеса мыши.

Аннотирование снимков экрана. Данная функция предоставляет встроенный графический редактор для нанесения на снимок поясняющих элементов: стрелок на основе кубических кривых Безье,

прямоугольных и эллиптических подсветок, нумерованных маркеров, текстовых блоков и комментариев в выносках. Все элементы являются перетаскиваемыми, поддерживают изменение размеров и удаление через стек операций отмены и повтора.

Применение эффектов обработки. Данная функция позволяет применять к выделенным областям снимков эффекты размытия по Гауссу и размытия в движении, а также скрывать конфиденциальную информацию средствами неразрушающего редактирования. Все операции выполняются с возможностью отмены в любой момент через стек операций.

Автоматическое описание снимков. Данная функция обеспечивает интеграцию с облачной мультимодальной моделью Google Gemini. Полученный снимок экрана кодируется в формат Base64 и передаётся в API облачного сервиса вместе с управляющим запросом, после чего возвращённый ответ преобразуется в связное текстовое описание, готовое для вставки в документ Microsoft Word. При недоступности основной модели подсистема автоматически переключается на резервные модели семейства Gemini.

Автоматическая вставка в Microsoft Word. Данная функция предоставляет возможность напрямую вставлять подготовленный снимок экрана в активный документ Microsoft Word посредством технологии СОМ-автоматизации. Перед изображением размещается описательный абзац, само изображение центрируется на странице, а под ним формируется подпись «Рисунок N - наименование» с автоматической нумерацией рисунков и оформлением по требованиям ГОСТ.

Формирование отчёта в открытом документе Microsoft Word. Данная функция обеспечивает автоматическое формирование итогового документа формата DOCX непосредственно в активном экземпляре

Microsoft Word. На основе шаблона титульного листа подсистема автоматически заполняет поля сведений об авторе, теме работы, специальности и руководителе, после чего формирует разделы пояснительной записки с оформлением по ГОСТ. Для работы функции документ Microsoft Word должен быть открыт; экспорт в закрытый файл не предусмотрен.

Автоматическая коррекция русскоязычного текста. Данная функция позволяет в реальном времени проверять и исправлять опечатки в наборе подписей и текстовых блоков. Подсистема построена на ансамбле из фильтра Блума, морфологического анализатора OpenCorpora, словаря объёмом порядка полутора миллионов словоформ и нейросетевой модели уровня символов; срабатывает при наборе текста и предлагает варианты исправления.

Сопровождающие подсистемы. Дополнительно приложение реализует чтение контекста элемента пользовательского интерфейса под текущим положением курсора через интерфейсы Microsoft UI Automation, отслеживание истории системного буфера обмена с возможностью повторной вставки произвольного фрагмента, а также полнотекстовый поиск файлов на томах NTFS на основе прямого чтения таблицы Master File Table - указанные возможности расширяют сценарии повседневного применения приложения.

Запись экранного видео. Данная функция обеспечивает захват видеопотока выбранного монитора с одновременным сведением системного звука и сигнала микрофона через интерфейс WASAPI. Сохранение результата производится в форматах MP4, WMV или AVI; полученные ролики могут использоваться для иллюстрирования сложных пользовательских сценариев в дополнение к статичным снимкам экрана.

Настройка приложения. Данная функция предоставляет пользователю интерфейс для управления параметрами работы программы: выбора глобальной горячей клавиши, цветовой схемы редактора, шрифтов и кеглей подписей, пути к шаблону документа, API-ключа облачной языковой модели и иных предпочтений. Параметры сохраняются между сеансами в каталоге AppData пользователя в формате INI и JSON.

Приложение является однопользовательским и не предусматривает разграничения доступа на основе ролей. Однако функциональность приложения зависит от технических условий среды эксплуатации - наличия установленного Microsoft Word, действующего API-ключа Google Gemini и подключения к сети Интернет. Соответствующие условия доступности функций представлены в таблице 4.

Таблица 4 - Условия доступности функций приложения

Категория функций	Условия использования	Описание
Базовые	Без дополнительных требований	Захват снимков экрана, аннотирование, применение эффектов размытия, отмена и повтор операций, сохранение в графические форматы, настройка приложения, история буфера обмена, поиск файлов по NTFS Master File Table.
Интеграция с Word	Установлен Microsoft Word и открыт документ	Автоматическая вставка снимка экрана в текущую позицию курсора с подписью «Рисунок N - наименование», вставка титульного листа по шаблону, формирование сквозной нумерации страниц.
Облачное описание	Указан API-ключ Google Gemini, есть подключение к сети Интернет	Автоматическое формирование текстовых описаний снимков экрана средствами мультимодальной языковой модели, перефразирование произвольных абзацев пояснительной записки, проверка стилистики.

Категория функций	Условия использования	Описание
Запись видео и звука	Поддержка WASAPI на устройстве	Захват видеопотока экрана с одновременным сведением системного звука и сигнала микрофона, экспорт в форматы MP4, WMV и AVI.
Автономный экспорт	Без дополнительных требований	Сборка готового документа DOCX через библиотеку Open XML SDK по шаблону ГОСТ без установленного пакета Microsoft Office.

Конец таблицы 4 - Условия доступности функций приложения

2.1.2. Входные и выходные данные

Входные данные представляют собой потоки информации, поступающие в приложение из различных источников - графической подсистемы операционной системы, аудиоподсистемы WASAPI, указательного устройства пользователя, клавиатуры, файлов настроек и облачного программного интерфейса. Перечень входных данных приведён в таблицах 5-7.

Таблица 5 - Входные данные модуля захвата контента

Наименование	Тип	Описание
Снимок экрана	Растровое изображение BGRA	Растровый образ видимой области одного или нескольких мониторов, получаемый в момент срабатывания глобальной горячей клавиши. Размер соответствует разрешению экрана с учётом коэффициента DPI.
Координаты выделения	Структура Rectangle	Координаты прямоугольной либо произвольной области, выделенной пользователем при помощи указательного устройства на оверлее замороженного экрана.
Видеопоток экрана	Последовательность кадров	Поток растровых кадров рабочего стола, получаемый через интерфейсы Desktop Duplication API при работе подсистемы записи видео.
Системный звук	PCM-поток 32 бит float	Цифровой аудиопоток, получаемый через интерфейс WASAPI в режиме loopback, используется для записи системного звука одновременно с экраным видеопотоком.
Сигнал микрофона	PCM-поток 32 бит float	Цифровой аудиопоток с устройства захвата по умолчанию, получаемый через интерфейс WASAPI в режиме capture.

Таблица 6 - Входные данные графического редактора

Наименование	Тип	Описание
Текст аннотаций	Строка Unicode	Произвольные текстовые блоки, комментарии в выносках и подписи, набираемые пользователем во встроенном графическом редакторе аннотаций.
Параметры стилей	Структуры Settings	Цвет, толщина линий, шрифт, размер кегля, форма стрелок, степень размытия и иные параметры стилей, задаваемые пользователем через панели управления редактора.
Координаты элементов	Список PointF	Контрольные точки кривых Безье для стрелок, координаты выделенных областей подсветки, позиции маркеров и текстовых блоков, перетаскиваемых пользователем при помощи указательного устройства.
Действия пользователя	Поток событий	Последовательность событий клавиатуры и указательного устройства, обрабатываемая графическим редактором: нажатия клавиш быстрого доступа, перемещения курсора, изменения выделения.

Таблица 7 - Входные данные подсистем интеграции

Наименование	Тип	Описание
API-ключ Gemini	Строка длиной до 256 символов	Конфиденциальная строка, обеспечивающая аутентификацию запросов к облачному программному интерфейсу Google Gemini. Хранится в защищённом виде в каталоге AppData пользователя.
Шаблон титульного листа	Документ DOCX	Файл шаблона титульного листа выпускной квалификационной работы с подстановочными полями, используемый подсистемой формирования отчёта.
Параметры пользователя	INI / JSON	Сохраняемые между сеансами параметры: глобальная горячая клавиша, цветовая схема редактора, путь к шаблону, фамилия и инициалы автора, наименование группы, цветовые предпочтения.
Контекст элемента UI	Дерево UI Automation	Атрибуты элемента пользовательского интерфейса под текущим положением курсора, получаемые через интерфейсы Microsoft UI Automation для подсистемы автоматической коррекции текста.
История буфера обмена	Список Object	Текстовые и графические записи системного буфера обмена, отслеживаемые через сообщения WM_CLIPBOARDUPDATE.

Выходные данные формируются приложением в результате обработки входных потоков и представляют собой готовые к использованию артефакты - графические файлы, видеоролики, фрагменты документов Microsoft Word, итоговые отчёты в формате DOCX и сохранённые пользовательские параметры. Перечень выходных данных приведён в таблицах 8-9.

Таблица 8 - Выходные данные графической подсистемы

Наименование	Тип	Описание
Аннотированный снимок	PNG / JPEG	Растровый образ выделенной области экрана с нанесёнными аннотациями, эффектами размытия и иными модификациями, готовый к сохранению в файл или передаче через буфер обмена.
Видеоролик с аудио	MP4 / WMV / AVI	Видеозапись экранного содержимого с одновременной звуковой дорожкой, сведённой из системного звука и сигнала микрофона. Используется для иллюстрирования сложных сценариев использования.
Текстовое описание	Строка Unicode	Связное текстовое описание снимка экрана, сформированное на основе ответа облачной мультимодальной модели Google Gemini.
Подпись рисунка	Строка Unicode	Сформированная по требованиям ГОСТ подпись изображения вида «Рисунок N - наименование» с автоматической сквозной нумерацией.

Таблица 9 - Выходные данные подсистем экспорта

Наименование	Тип	Описание
Документ Microsoft Word	Изменения в открытом DOCX	Изменения, вносимые в активный документ Microsoft Word через интерфейс COM-автоматизации: вставка описательного абзаца, изображения и подписи рисунка.
Автономный отчёт DOCX	Файл DOCX	Готовый файл документа в формате Open XML, собранный библиотекой Open XML SDK без установленного пакета Microsoft Office. Содержит титульный лист, абзацы пояснительного текста, иллюстрации с подписями.

Наименование	Тип	Описание
Файл настроек	INI	Текстовый файл с параметрами приложения в каталоге AppData пользователя, сохраняемый при изменении настроек.
Файл данных	JSON	Файл с пользовательскими списками значений (тем, дисциплин, групп) и параметрами интеграции с Gemini, сохраняемый в каталоге AppData пользователя.
Уведомления	Всплывающие окна	Короткие всплывающие сообщения о завершении операций экспорта, успешном получении описания снимка от облачной модели или ошибках сетевых запросов.

Конец таблицы 9 - Выходные данные подсистем экспорта

2.2. Внешняя спецификация

2.2.1. Метод

В процессе создания программного продукта использовалась спиральная модель жизненного цикла. Выбор данной методологии обусловлен возможностью повторного обращения к предыдущим этапам разработки, оперативной демонстрацией промежуточных результатов и поэтапной проверкой ключевых рисков, связанных с интеграцией с внешними сервисами и приложениями. Применение спиральной модели позволило в каждом из четырёх витков уточнять требования к подсистемам захвата экрана, аннотирования, описания снимков средствами мультимодальной модели и автоматической вставки в документы Microsoft Word, после чего возвращаться к проектированию архитектуры с учётом обратной связи.

На рисунке 1 представлена спиральная модель жизненного цикла.

Спиральная модель жизненного цикла



Рисунок 1 - Спиральная модель жизненного цикла

В рамках первого витка были собраны функциональные требования к приложению, проанализированы аналогичные программные продукты и спроектирована модульная архитектура решения с разделением на слои захвата контента, рендеринга графического редактора, обработки текста и интеграции с внешними сервисами. Второй виток был посвящён реализации модуля захвата изображения с интеллектуальным выделением областей и встроенного графического редактора аннотаций. Третий виток включал интеграцию с мультимодальной языковой моделью Google Gemini и подсистему автоматической вставки в Microsoft Word через интерфейс COM-

автоматизации. На четвёртом витке были подключены вспомогательные подсистемы и проведено тестирование готового продукта.

2.2.2. Тесты

Приложение реализует значительное число функций, корректность которых должна быть проверена на каждом этапе проектирования. Для проверки работоспособности разработанных функций был сформирован перечень тест-кейсов, охватывающих основные сценарии использования и критические участки логики приложения. Перечень тест-кейсов представлен в таблице 10.

Таблица 10 - Тест-кейсы

№	Тест-кейс	Шаги выполнения	Ожидаемый результат
1	Активация захвата по горячей клавише	Нажать глобальную горячую клавишу, заданную в настройках	Открывается оверлей замороженного экрана; курсор переходит в режим выделения области
2	Прямоугольное выделение области	Удерживая ЛКМ, выделить произвольную прямоугольную область на оверлее	После отпускания ЛКМ открывается окно редактора с выделенным фрагментом
3	Авто-выделение окна	Навести курсор на окно приложения и нажать колесо мыши	Выделяется граница окна целиком средствами оператора Собеля и алгоритма заливки
4	Добавление стрелки	В редакторе выбрать инструмент «Стрелка» и провести линию по диагонали	На холсте появляется кубическая кривая Безье с тенью и стрелкой на конце
5	Добавление подсветки	В редакторе выбрать инструмент «Подсветка» и выделить прямоугольную область	На холсте появляется полупрозрачный жёлтый прямоугольник с обрамлением
6	Добавление маркера	В редакторе выбрать инструмент «Маркер» и кликнуть в нужной точке	На холсте появляется красный круг с порядковым номером
7	Размытие выделенной области	Выделить область подсветкой, нажать кнопку «Размытие»	Внутри выделенной области применяется эффект размытия по Гауссу с заданным радиусом

№	Тест-кейс	Шаги выполнения	Ожидаемый результат
8	Отмена операции	После добавления стрелки нажать сочетание Ctrl+Z	Стрелка удаляется с холста; стек отмены содержит обратную операцию
9	Запрос автоописания Gemini	В боковой панели нажать кнопку «Создать описание Gemini»	Снимок отправляется в облачный API; в поле «Описание» появляется связанный текст
10	Вставка в Word	Открыть документ Word, в боковой панели приложения нажать «Вставить рисунок в Word»	В позиции курсора Word появляется описательный абзац, изображение по центру и подпись «Рисунок N - ...»
11	Запись видео со звуком	Запустить запись видео в режиме монитора, через 5 секунд остановить	В выбранной папке появляется файл MP4 с видео и звуковой дорожкой
12	Сохранение настроек	Изменить горячую клавишу в окне настроек и нажать «Сохранить»	В файле %AppData%\WindowCapture\settings.ini обновляется значение ActivationKey
13	Уникализация текста	В боковой панели нажать «Уникализировать текст»	Каждое предложение в открытом документе перефразируется средствами Gemini
14	Поиск файла по MFT	В диалоге настроек указать путь к шаблону через подсистему поиска	Подсистема возвращает список путей, содержащих указанное имя
15	Аварийный режим	Отключить сеть Интернет и нажать «Создать описание Gemini»	Приложение выводит уведомление об ошибке сетевого запроса; интерфейс остаётся отзывчивым

Конец таблицы 10 - Тест-кейсы

2.3. Проектирование

2.3.1. Схема данных

Схема данных приложения описывает структуру сериализуемых пользовательских объектов, сохраняемых между сеансами работы программы. В отличие от веб-приложений, использующих реляционные базы данных, рассматриваемое настольное приложение хранит сведения о настройках и пользовательских предпочтениях в локальных файлах формата INI и JSON в каталоге AppData. Объектная модель приложения включает классы настроек, привязок горячих клавиш, данных титульного листа и аннотаций редактора. На рисунке 2 представлена обобщённая схема сериализуемых объектов модели приложения.

Схема сериализуемых объектов модели

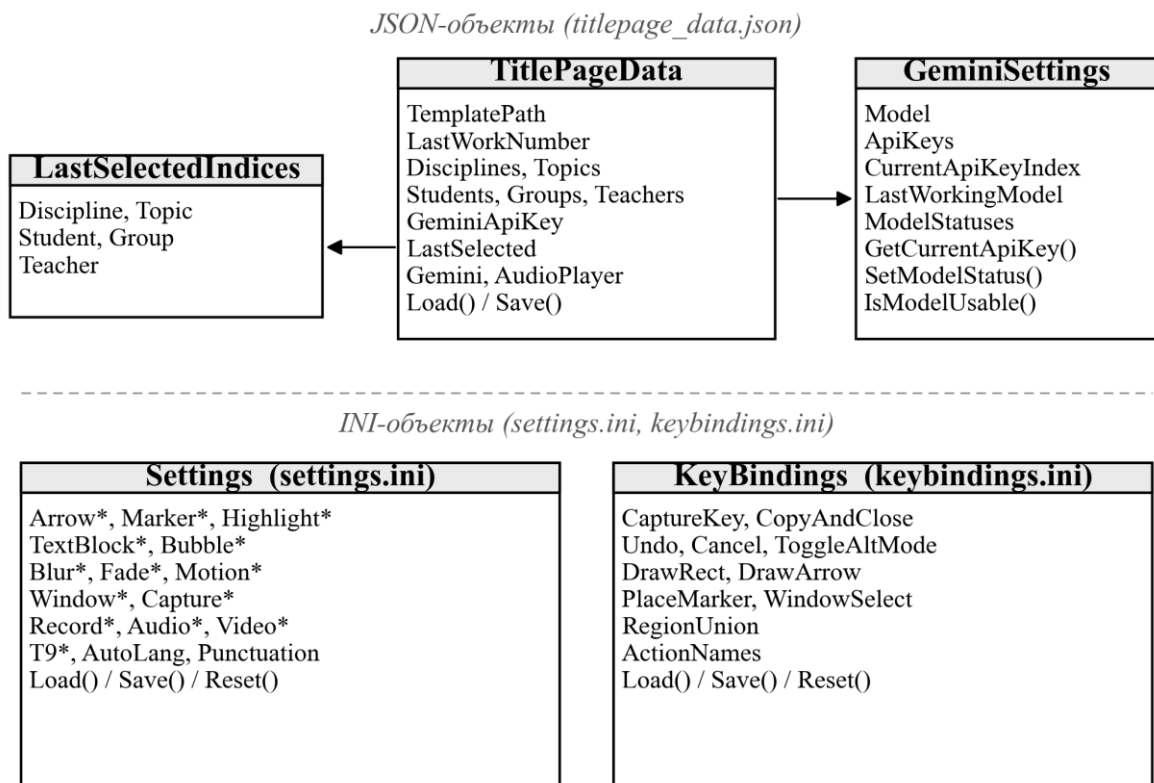


Рисунок 2 - Схема сериализуемых объектов модели

Структура каждого из объектов модели подробно рассмотрена в таблицах 11-15.

Таблица 11 - Объект «Settings»

Наименование поля	Физическое имя	Тип данных	Назначение
Горячая клавиша захвата	ActivationKey	int	Виртуальный код клавиши, активирующей захват экрана. По умолчанию используется код 0xA4 (левый Alt).
Имя клавиши	ActivationKeyName	string	Текстовое представление горячей клавиши для отображения в пользовательском интерфейсе.
Цвет стрелок	ArrowColor	Color	Цвет линий и заливки стрелок на основе кубических кривых Безье.
Толщина стрелок	ArrowWidth	float	Толщина линии стрелки в пунктах.
Цвет маркеров	MarkerColor	Color	Цвет заливки нумерованных маркеров.
Радиус размытия	BlurRadius	float	Базовый радиус размытия по Гауссу в пунктах.
Цвет фона	BlurTintColor	Color	Цвет тонировки эффекта акрилового стекла окна редактора.
Прозрачность	BlurTintAlpha	int	Прозрачность тонировки в диапазоне от 0 до 255.
Громкость	AudioVolume	float	Громкость воспроизведения медиа в диапазоне от 0,0 до 1,0.

Таблица 12 - Объект «KeyBindings»

Наименование действия	Физическое имя	Тип	Назначение
Захват экрана	Capture	Keys	Сочетание клавиш, активирующее модуль захвата контента.
Отмена	Cancel	Keys	Сочетание клавиш для отмены текущего выделения области.
Отмена операции	Undo	Keys	Сочетание клавиш для отмены последнего изменения в редакторе.
Повтор операции	Redo	Keys	Сочетание клавиш для повтора отменённой операции.

Наименование действия	Физическое имя	Тип	Назначение
Копирование	Copy	Keys	Сочетание клавиш для копирования снимка в системный буфер обмена.
Сохранение	Save	Keys	Сочетание клавиш для сохранения снимка в графический файл.
Экспорт в Word	ExportToWord	Keys	Сочетание клавиш для вставки снимка в активный документ Microsoft Word.

Конец таблицы 12 - Объект «KeyBindings»

Таблица 13 - Объект «TitlePageData»

Наименование поля	Физическое имя	Тип	Назначение
Путь к шаблону	TemplatePath	string	Абсолютный путь к файлу шаблона титульного листа в формате DOCX.
Дисциплины	Disciplines	List<string>	Список наименований дисциплин, доступных для подстановки в титульный лист.
Темы	Topics	List<string>	Список тем выпускных квалификационных работ.
Студенты	Students	List<string>	Список фамилий и инициалов авторов работ.
Группы	Groups	List<string>	Список наименований учебных групп.
Преподаватели	Teachers	List<string>	Список фамилий и инициалов руководителей и консультантов.
API-ключ Gemini	GeminiApiKey	string	Конфиденциальная строка, обеспечивающая аутентификацию запросов к облачному API.
Параметры Gemini	Gemini	GeminiSettings	Вложенный объект с настройками облачной мультимодальной модели.
Параметры медиаплеера	AudioPlayer	AudioPlayerSettings	Вложенный объект с параметрами встроенного медиаплеера.

Таблица 14 - Объект «ArrowAnnotation»

Наименование поля	Физическое имя	Тип	Назначение
Контрольные точки	ControlPoints	PointF[4]	Четыре контрольные точки кубической кривой Безье, определяющие форму стрелки.
Цвет	Color	Color	Цвет линии и заливки стрелки.
Толщина линии	Width	float	Толщина линии стрелки в пунктах.
Смещение тени	ShadowOffset	int	Смещение тени в пикселях по горизонтали и вертикали.
Прозрачность тени	ShadowAlpha	int	Прозрачность тени в диапазоне от 0 до 255.
Точка кривой	GetBezierPoint(t)	PointF	Метод получения координат точки на кривой Безье по параметру t от 0 до 1.
Касательная	GetBezierTangent(t)	PointF	Метод получения вектора касательной к кривой Безье в точке t.

Таблица 15 - Объект «HighlightRect»

Наименование поля	Физическое имя	Тип	Назначение
Границы	Bounds	Rectangle	Прямоугольник, определяющий область подсветки на снимке экрана.
Цвет заливки	FillColor	Color	Цвет полупрозрачной заливки выделенной области.
Прозрачность	Alpha	int	Прозрачность заливки в диапазоне от 0 до 255.
Цвет границы	BorderColor	Color	Цвет линии обрамления выделенной области.
Толщина границы	BorderWidth	float	Толщина линии обрамления в пунктах.
Состояние размытия	BlurState	enum	Текущее состояние эффекта размытия: None / BlurInside / BlurOutside / MotionInside / MotionOutside.

2.3.2. Функциональная схема

Функциональная схема представляет собой иерархическую декомпозицию основных возможностей приложения, отображающую полный функционал программы. На рисунке 3 представлена функциональная схема приложения.

Функциональная схема приложения



Рисунок 3 - Функциональная схема приложения

Более подробное описание функционала, изображённого на схеме, рассмотрено ниже.

Захват экрана - функция, обеспечивающая получение растрового образа произвольной области экрана по нажатию глобальной горячей клавиши. Захват выполняется средствами BitBlt, с последующим отображением замороженного оверлея во весь экран.

Авто-выделение области - функция, реализующая интеллектуальное определение границ окна, строки или экрана по нажатию колеса указательного устройства. Применяется оператор

Собеля для выделения краёв и алгоритм заливки по цветовому сходству на основе поиска в ширину.

Запись видео с системным звуком - функция, обеспечивающая захват видеопотока выбранного монитора с одновременной звуковой дорожкой. Сведение системного звука и сигнала микрофона выполняется через интерфейс WASAPI; экспорт - в форматах MP4, WMV и AVI.

Аннотирование - функция, предоставляющая инструменты для нанесения на снимок стрелок Безье, прямоугольных и эллиптических подсветок, нумерованных маркеров, текстовых блоков и комментариев в выносках. Все элементы являются перетаскиваемыми и поддерживают изменение размеров.

Эффекты обработки - функция, обеспечивающая применение размытия по Гауссу, размытия в движении и неразрушающее редактирование выделенных областей. Все операции выполняются с возможностью отмены через стек операций.

Запрос к Google Gemini - функция, реализующая обращение к облачному программному интерфейсу мультимодальной языковой модели для получения текстового описания снимка экрана с учётом контекста.

Автокоррекция русского текста - функция, обеспечивающая в реальном времени проверку и исправление опечаток в подписях и текстовых блоках на основе ансамбля из фильтра Блума, морфологического анализатора OpenCorpora и нейросетевой модели уровня символов.

UI Automation - функция чтения контекста элемента пользовательского интерфейса под текущим положением курсора, используемая модулем автокоррекции для уточнения языкового контекста.

Поиск файлов по NTFS MFT - функция полнотекстового поиска файлов на томах NTFS на основе прямого чтения таблицы Master File Table; работает быстрее штатного поиска проводника операционной системы и не требует индексации.

Вставка в Word - функция, обеспечивающая прямую вставку аннотированного снимка экрана в активный документ Microsoft Word через интерфейс СОМ-автоматизации с автоматическим формированием подписи «Рисунок N - наименование» по требованиям ГОСТ.

Подпись «Рисунок N - ...» - функция автоматической нумерации иллюстраций; подсчитывает существующие в документе подписи и формирует следующий порядковый номер.

Сборка DOCX - функция автономного формирования итогового файла отчёта без установленного пакета Microsoft Office. Реализована средствами библиотеки Open XML SDK по шаблону ГОСТ.

Экспорт в MP4 / WMV / AVI - функция сохранения записанного видеопотока в распространённые медиаконтейнеры с настройкой качества и параметров кодирования.

2.3.3. Структурная схема

Приложение построено по многослойной архитектуре с разделением программных компонентов на слой управления, слой пользовательского интерфейса, слой данных, слой интеграции с внешними сервисами и слой платформенного взаимодействия. Каждый слой содержит несколько модулей, отвечающих за самостоятельные подсистемы приложения. На рисунке 4 представлена структурная схема приложения.

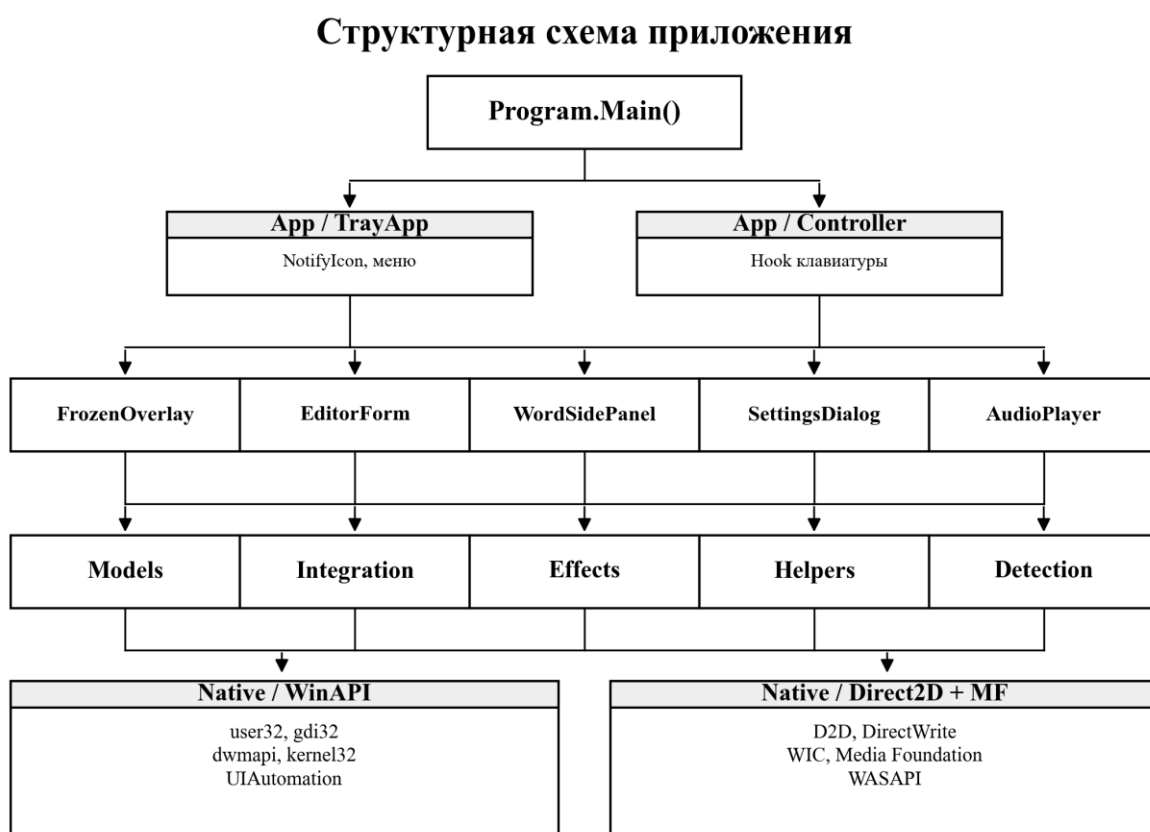


Рисунок 4 - Структурная схема приложения

2.3.4. Схема пользовательского интерфейса

Схема пользовательского интерфейса описывает навигацию пользователя по программе и переходы между основными формами и диалоговыми окнами. Поскольку приложение является однопользовательским и не предусматривает разграничения доступа на основе ролей, навигация представлена единой схемой. На рисунке 5 представлена схема пользовательского интерфейса приложения.

Схема пользовательского интерфейса

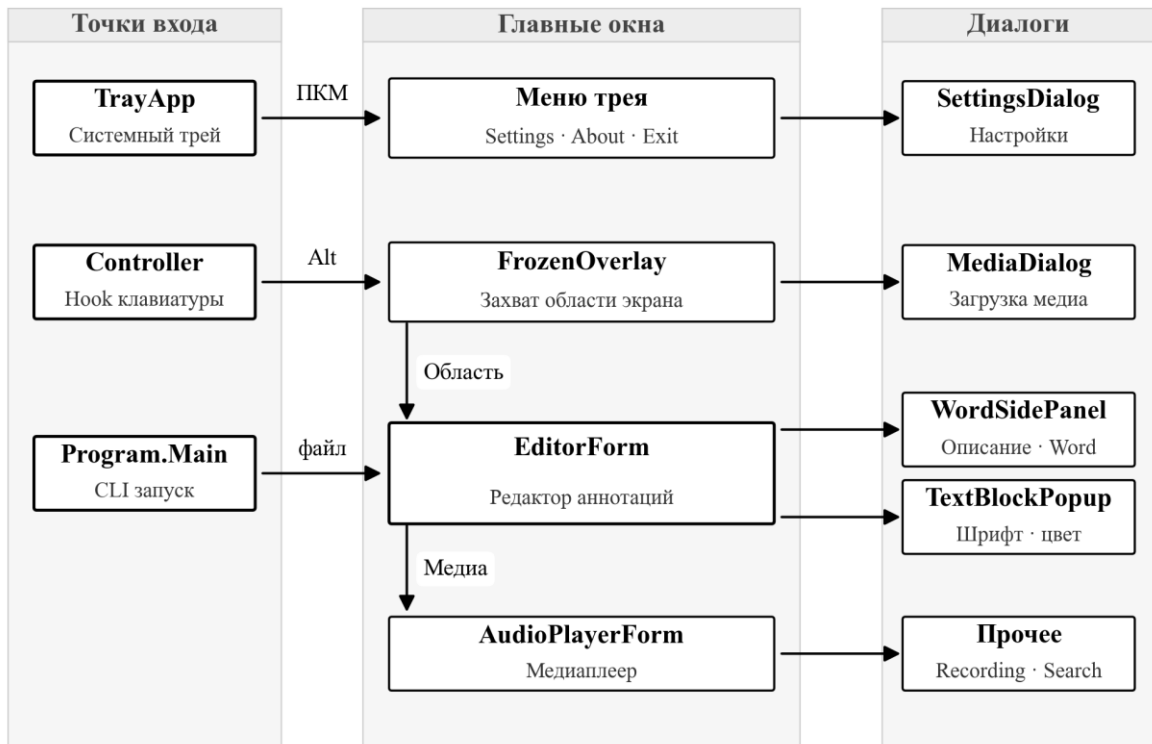


Рисунок 5 - Схема пользовательского интерфейса приложения

2.3.5. Укрупнённый алгоритм

Укрупнённый алгоритм представляет собой описание последовательности шагов или действий, не углубляясь в детали реализации приложения. Укрупнённый алгоритм отражает обобщённый сценарий применения программы - от нажатия глобальной горячей клавиши до получения готового абзаца с описанием и подписью внутри документа Microsoft Word. На рисунке 6 представлен укрупнённый алгоритм работы приложения.

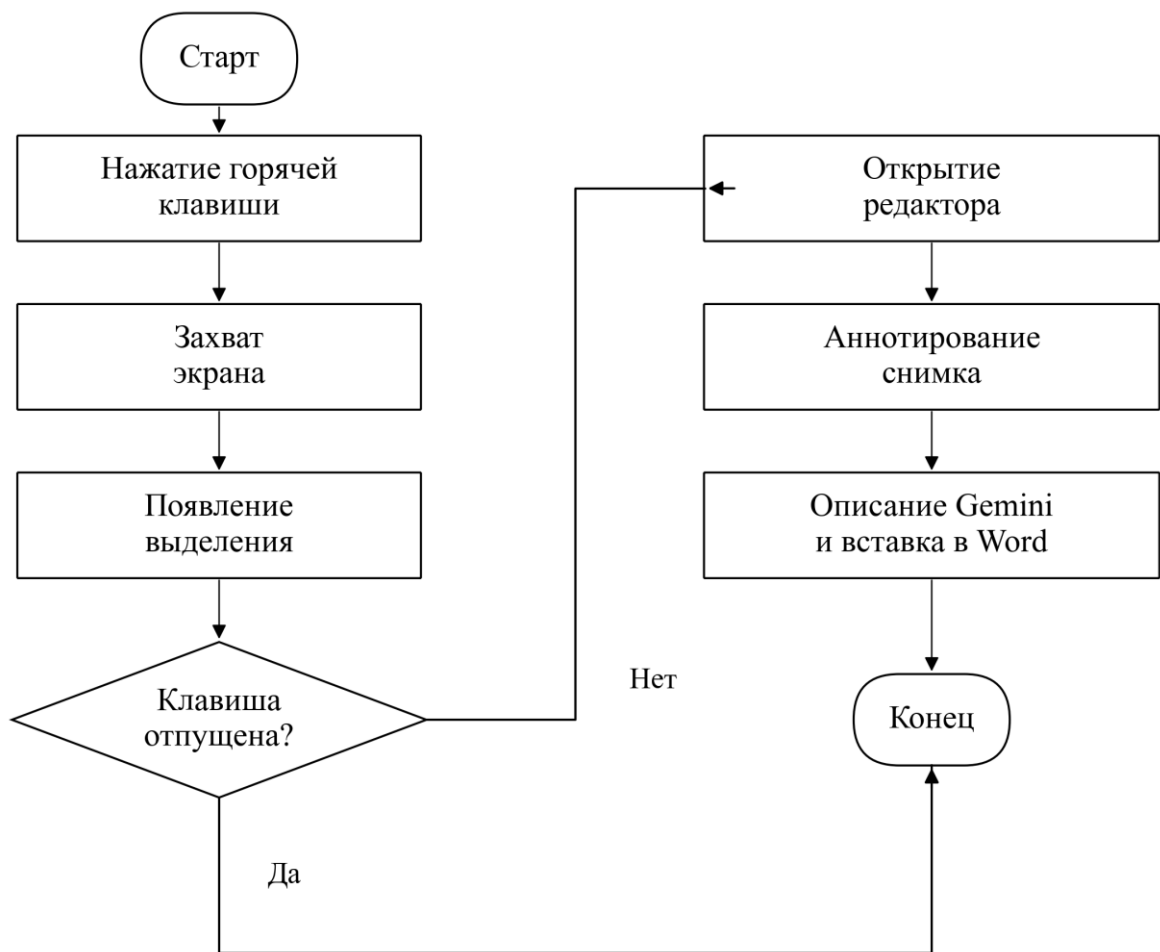


Рисунок 6 - Укрупнённый алгоритм работы приложения

2.3.6. Блок-схема

Блок-схема представляет собой визуальное представление логики выполнения задачи, где каждый блок обозначает действие, решение или данные, а стрелки показывают порядок и направление переходов между этапами. Главными подсистемами рассматриваемого приложения являются модуль захвата изображения с интеллектуальным выделением областей, подсистема обращения к облачной мультимодальной модели Google Gemini для получения описания снимков и подсистема автоматической вставки результата в активный документ Microsoft Word. На рисунке 7 представлен алгоритм захвата изображения, на рисунке 8 - алгоритм запроса описания у Google Gemini, на рисунке 9 - алгоритм автоматической вставки в документ Microsoft Word.

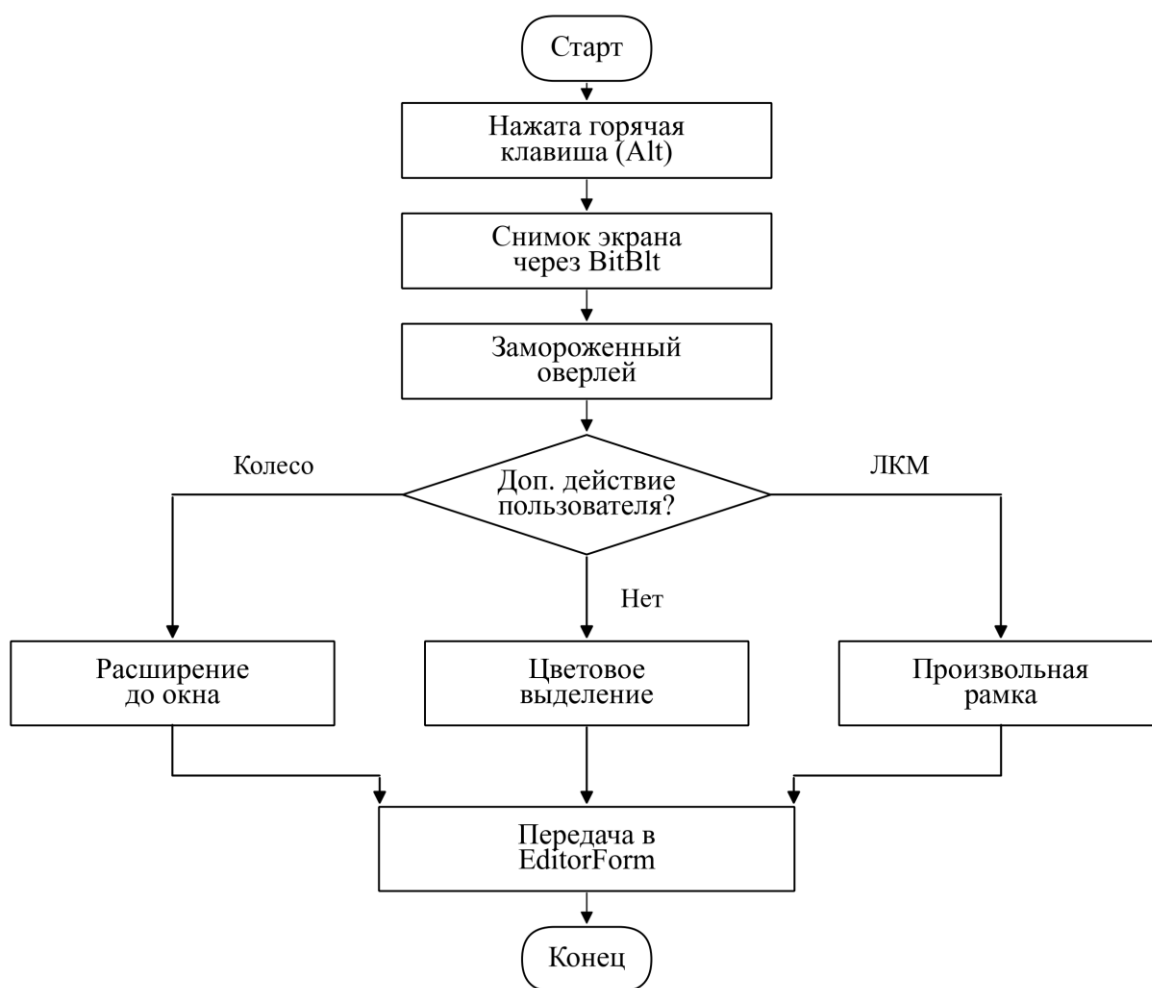


Рисунок 7 - Алгоритм захвата изображения

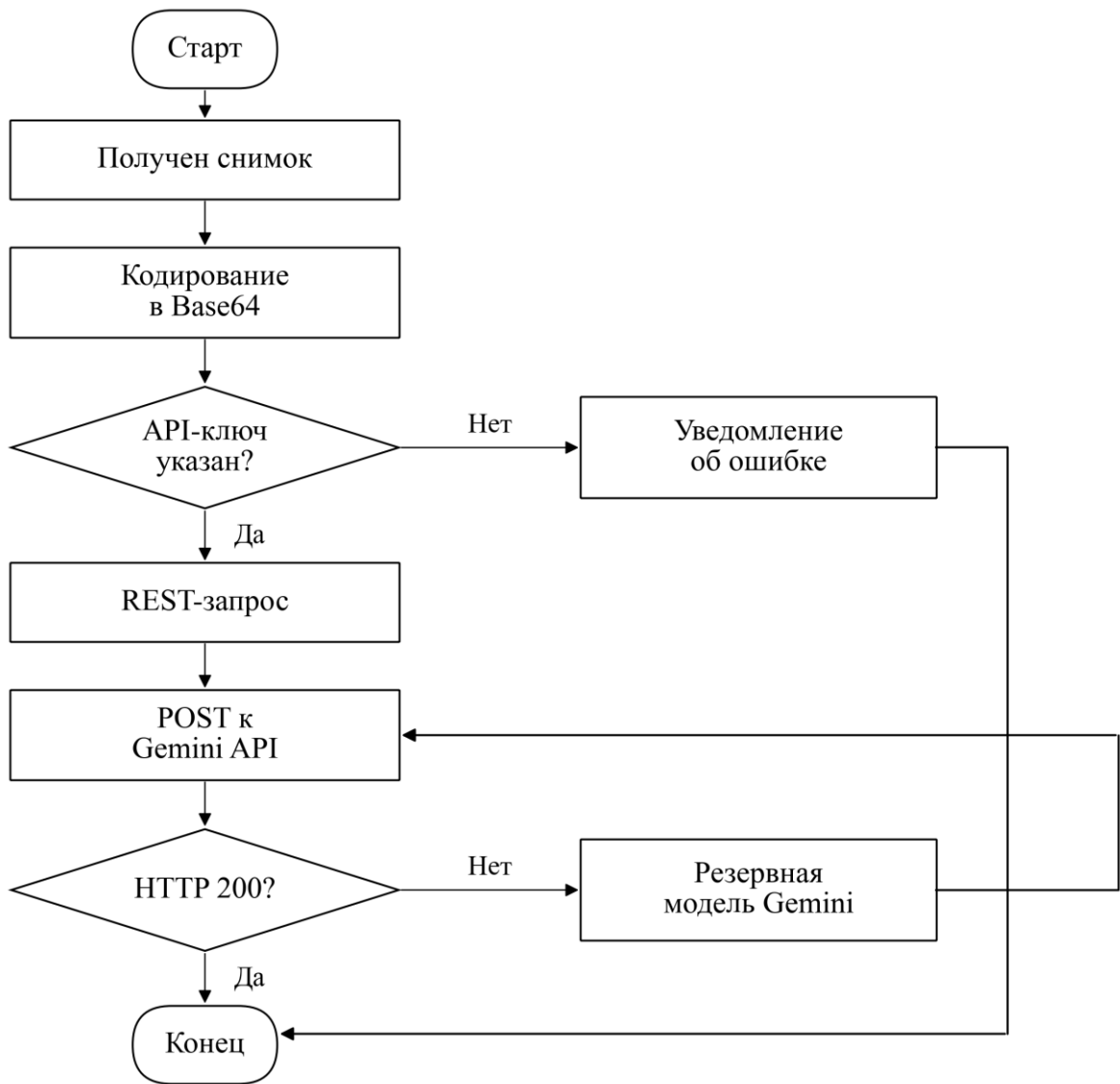


Рисунок 8 - Алгоритм запроса описания у Google Gemini

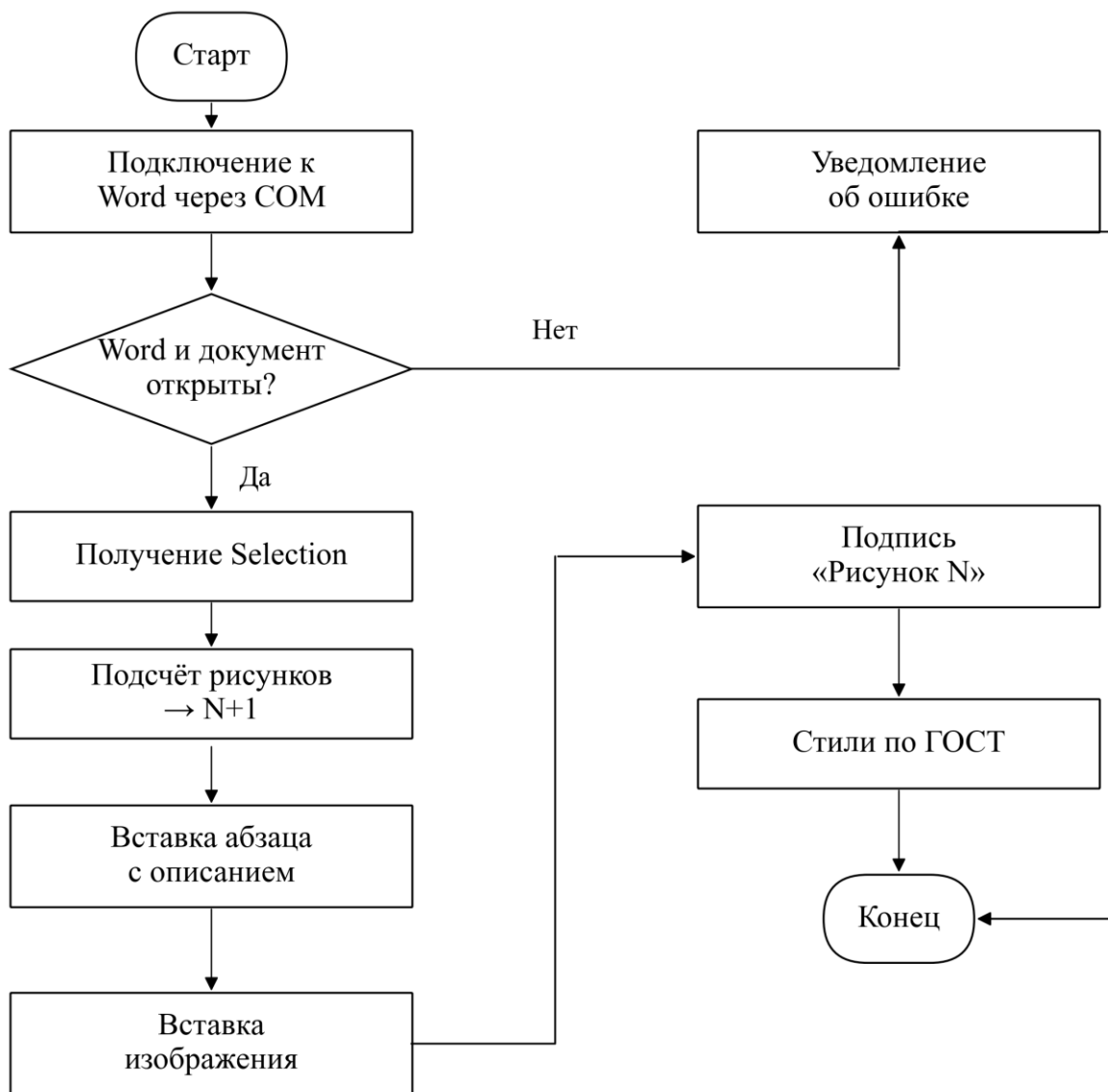


Рисунок 9 - Алгоритм автоматической вставки в Microsoft Word

2.4. Результаты работы программы

На рисунках 10-16 представлены результаты работы приложения. На рисунках 10 и 11 продемонстрирован запуск приложения и контекстное меню системного трея, через которое осуществляется доступ к настройкам и завершению работы программы.



Рисунок 10 - Иконка приложения в системном трее

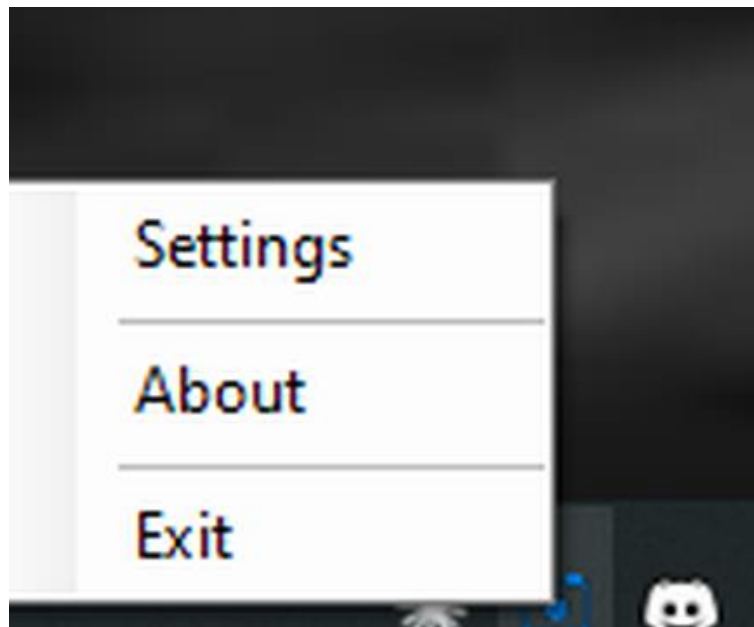


Рисунок 11 - Контекстное меню приложения

На рисунке 12 продемонстрирован замороженный оверлей с прямоугольным выделением области экрана, активируемым по нажатию глобальной горячей клавиши с одновременным удержанием левой кнопки мыши (данный способ выделения не является режимом по умолчанию).

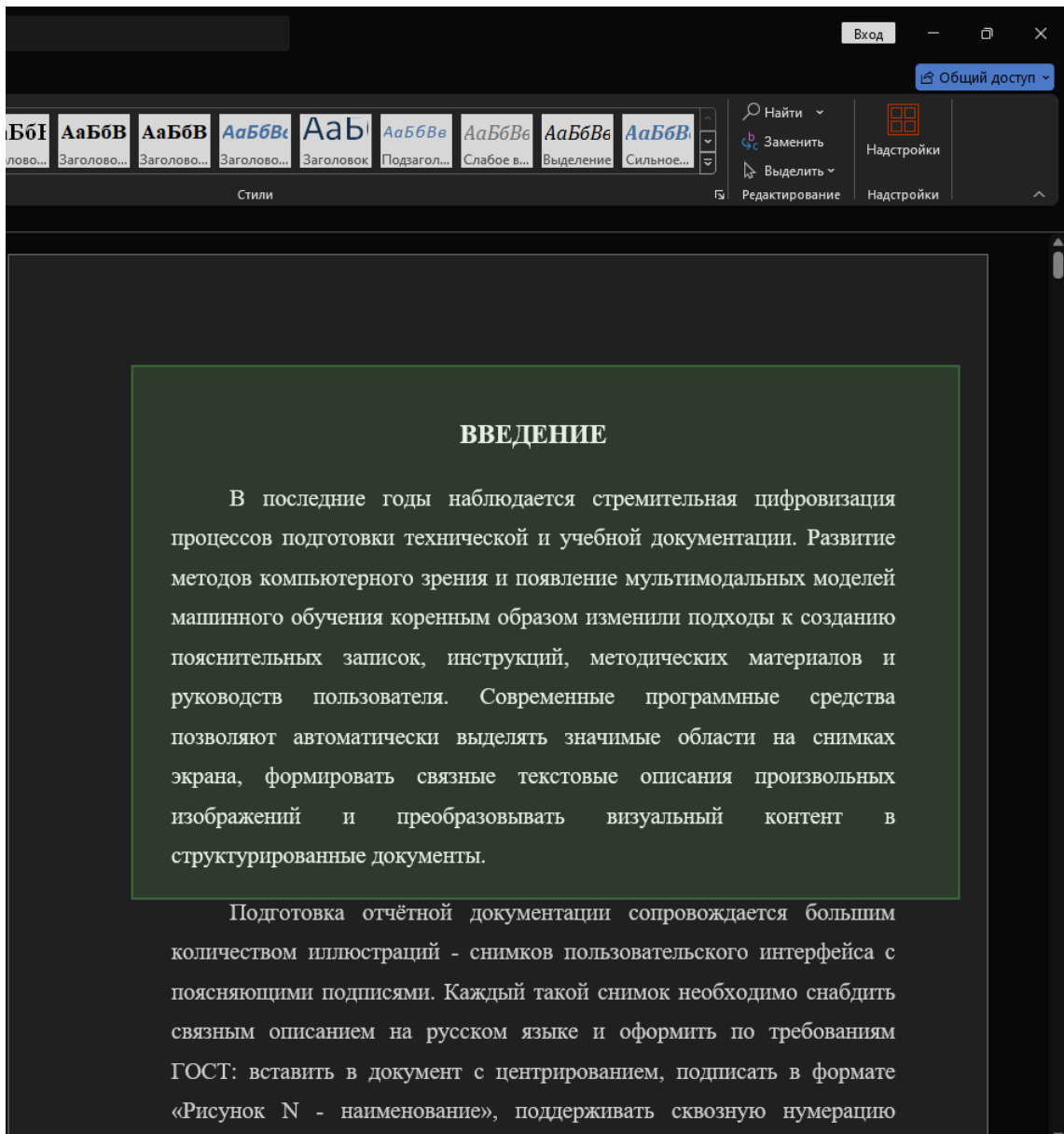


Рисунок 12 - Замороженный оверлей с выделением области

На рисунке 13 представлен встроенный графический редактор аннотаций, открытый после выделения области и содержащий примеры применения каждого инструмента редактора.

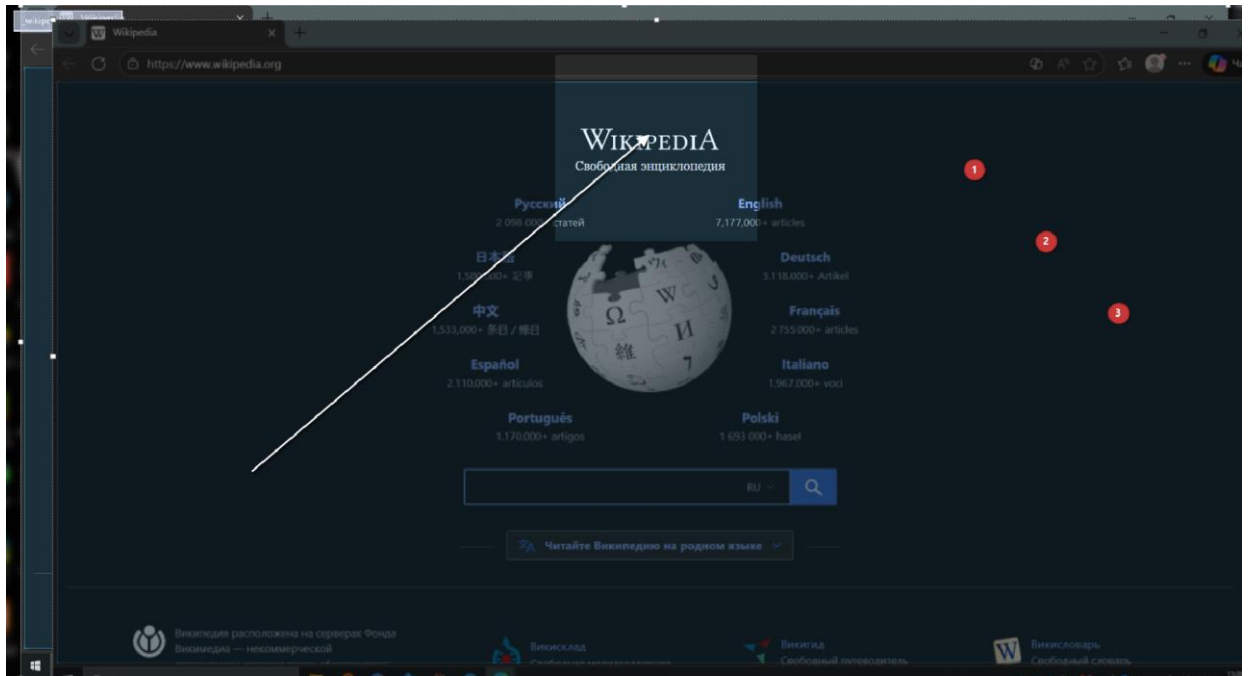


Рисунок 13 - Окно редактора аннотаций

На рисунке 14 представлено окно настроек приложения, обеспечивающее управление параметрами работы программы по вкладкам.

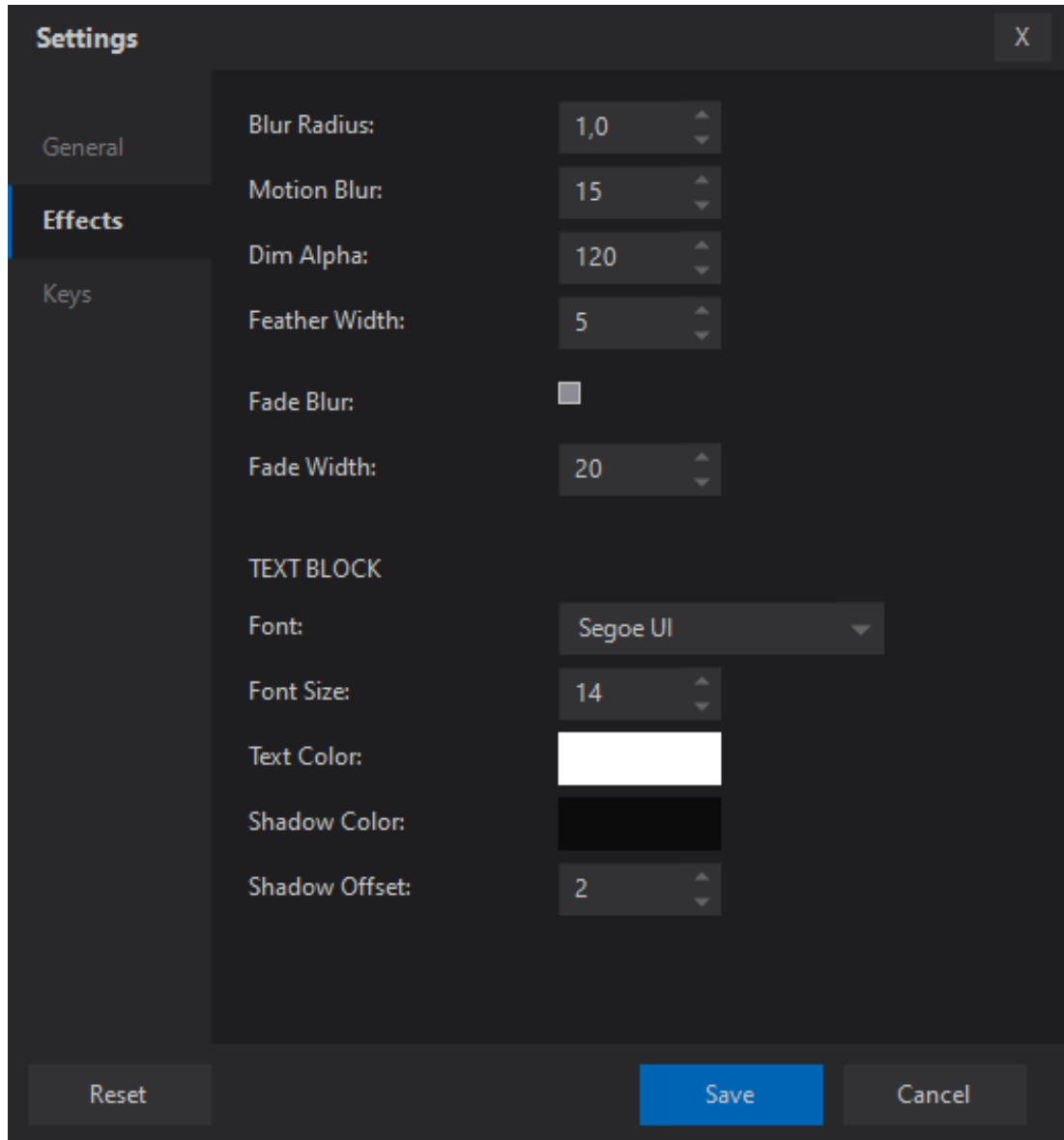


Рисунок 14 - Окно настроек приложения

На рисунке 15 представлена боковая панель экспорта в Microsoft Word - основной инструмент взаимодействия пользователя с целевым документом и облачной моделью Google Gemini.

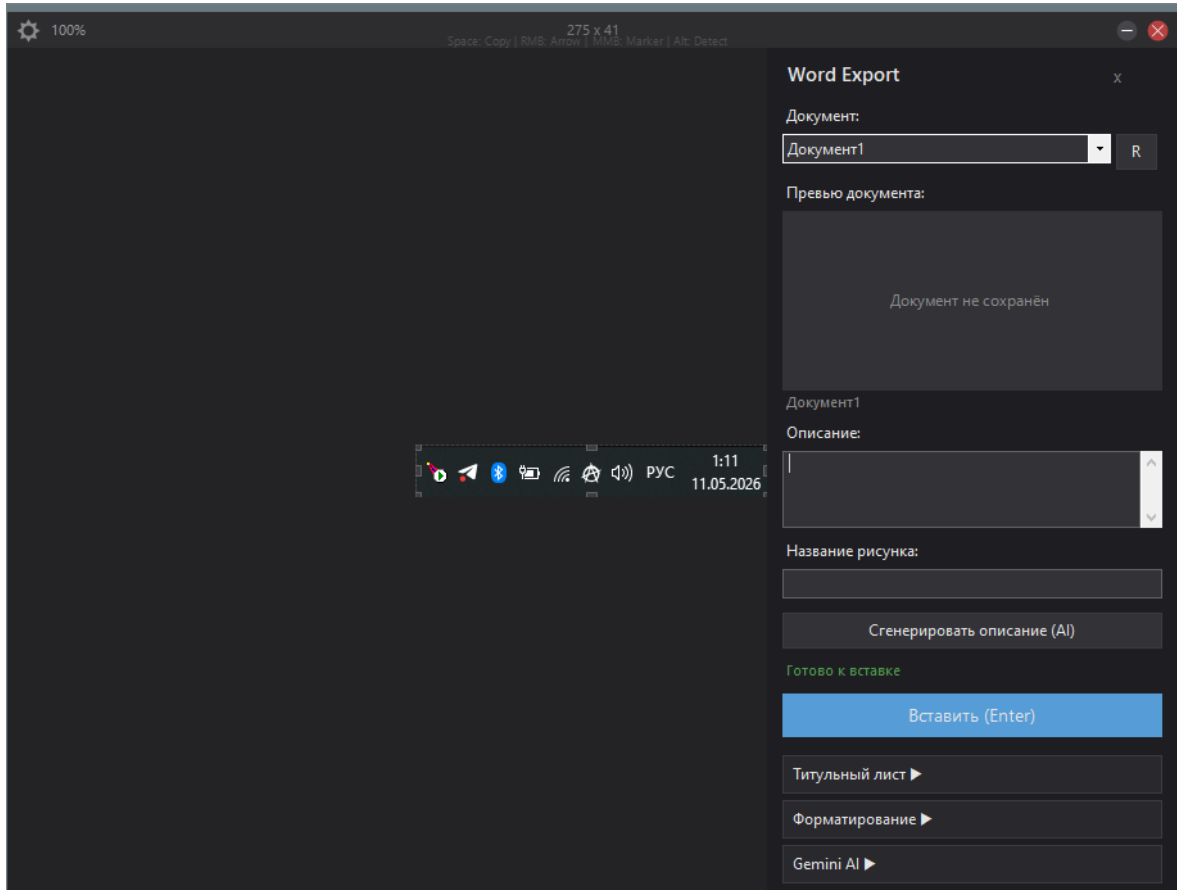


Рисунок 15 - Боковая панель экспорта в Microsoft Word

На рисунке 16 продемонстрирован результат автоматической вставки аннотированного снимка экрана в открытый документ Microsoft Word: описательный абзац перед изображением, центрированная иллюстрация и подпись «Рисунок N - наименование», сформированные в соответствии с требованиями ГОСТ.

На рисунке 1 представлено содержание сформированного документа, в котором наглядно видно иерархическую структуру пояснительной записки. Логическая последовательность разделов включает цели, средства разработки и детальное проектирование систем. Автоматическая генерация оглавления обеспечивает соответствие номеров страниц фактическому расположению материала в тексте согласно ГОСТ.

Содержание	
ВВЕДЕНИЕ	2
1. ОБЩАЯ ЧАСТЬ.....	7
1.1. Цели разработки.....	7
1.2. Средства разработки.....	8
1.2.1. Технические средства.....	8
1.2.2. Программные средства.....	9
2. СПЕЦИАЛЬНАЯ ЧАСТЬ	12
2.1. Постановка задачи	12
2.1.1. Описание задачи	12
2.1.2. Входные и выходные данные	16
2.2. Внешняя спецификация	19
2.2.1. Метод	19
2.2.2. Тесты	21
2.3. Проектирование	23
2.3.1. Схема данных	23
2.3.2. Функциональная схема.....	27
2.3.3. Структурная схема.....	30
2.3.4. Схема пользовательского интерфейса	31
2.3.5. Укрупнённый алгоритм.....	32
2.3.6. Блок-схема	33

Рисунок 1 — Содержание пояснительной записки

Рисунок 16 - Результат автоматической вставки в Microsoft Word

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

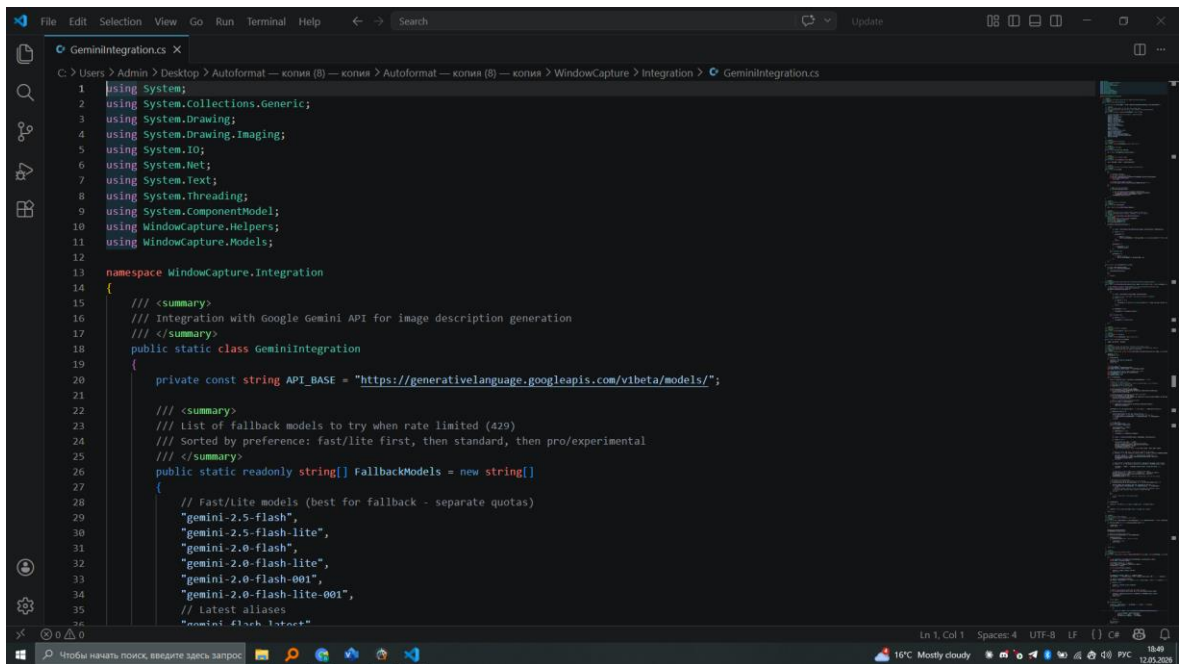
3.1. Инструментальные средства разработки

Для реализации настольного приложения был выбран комплект инструментальных средств, обеспечивающий компиляцию исполняемого файла без необходимости установки крупных интегрированных сред разработки на компьютер пользователя. Перечень используемых средств с указанием назначения приведён в таблице 6.

Основным языком программирования был выбран C# в редакции, совместимой с .NET Framework 4.0. Выбор обусловлен тем, что данная редакция .NET Framework предустановлена во всех версиях Windows, начиная с Windows 7 с пакетом обновления 1, что снимает необходимость в развёртывании дополнительных компонентов на компьютере конечного пользователя. C# в составе .NET Framework предоставляет полную поддержку платформенно-зависимых вызовов (P/Invoke) и автоматизации компонентной модели COM, что критически важно для интеграции с Microsoft Word и нативными графическими подсистемами Windows.

Сборка приложения выполняется через сценарий PowerShell build.ps1, вызывающий компилятор csc.exe из состава .NET Framework 4.0. Это позволяет исключить зависимость от установленной Visual Studio и собирать проект из 94 файлов исходного кода в единый исполняемый файл WindowCapture.exe. Совокупный объём исходного кода превышает 33 тысячи строк на языке C#.

В качестве редактора кода был выбран Visual Studio Code - лёгкий редактор от компании Microsoft с подсветкой синтаксиса C#, контекстными подсказками и встроенной интеграцией с системой контроля версий Git. На рисунке 17 показан исходный код модуля GeminiIntegration в окне Visual Studio Code.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Drawing.Imaging;
5 using System.IO;
6 using System.Net;
7 using System.Text;
8 using System.Threading;
9 using System.ComponentModel;
10 using WindowCapture.Helpers;
11 using WindowCapture.Models;
12
13 namespace WindowCapture.Integration
14 {
15     /// <summary>
16     /// Integration with google Gemini API for image description generation
17     /// </summary>
18     public static class GeminiIntegration
19     {
20         private const string API_BASE = "https://generativelanguage.googleapis.com/v1beta/models/";
21
22         /// <summary>
23         /// List of fallback models to try when rate limited (429)
24         /// Sorted by preference: fast/lite first, then standard, then pro/experimental
25         /// </summary>
26         public static readonly string[] FallbackModels = new string[]
27         {
28             // Fast/lite models (best for fallback - separate quotes)
29             "gemi-2.5-flash",
30             "gemi-2.5-flash-lite",
31             "gemi-2.0-flash",
32             "gemi-2.0-flash-lite",
33             "gemi-2.0-flash-001",
34             "gemi-2.0-flash-lite-001",
35             // Latest aliases
36             "gemi-flash-latest"
```

Рисунок 17 - Среда разработки Visual Studio Code с открытым исходным кодом

Для генерации текстовых описаний снимков экрана используется облачная нейросеть Google Gemini, доступная по REST-протоколу. Взаимодействие реализовано на стандартных средствах .NET HttpWebRequest и DataContractJsonSerializer, без подключения сторонних библиотек. Интеграция с Microsoft Word построена на технологии COM-автоматизации через диспетчер Type.GetTypeFromProgID. Графический вывод производится через нативные интерфейсы Direct2D, DirectWrite и WIC, подключаемые через P/Invoke к системным библиотекам d2d1.dll и dwwrite.dll. Аудио- и видеозапись построены на интерфейсах Media Foundation и WASAPI.

Сводный перечень использованных инструментальных средств с указанием их назначения представлен в таблице 6.

Таблица 6 - Инструментальные средства разработки

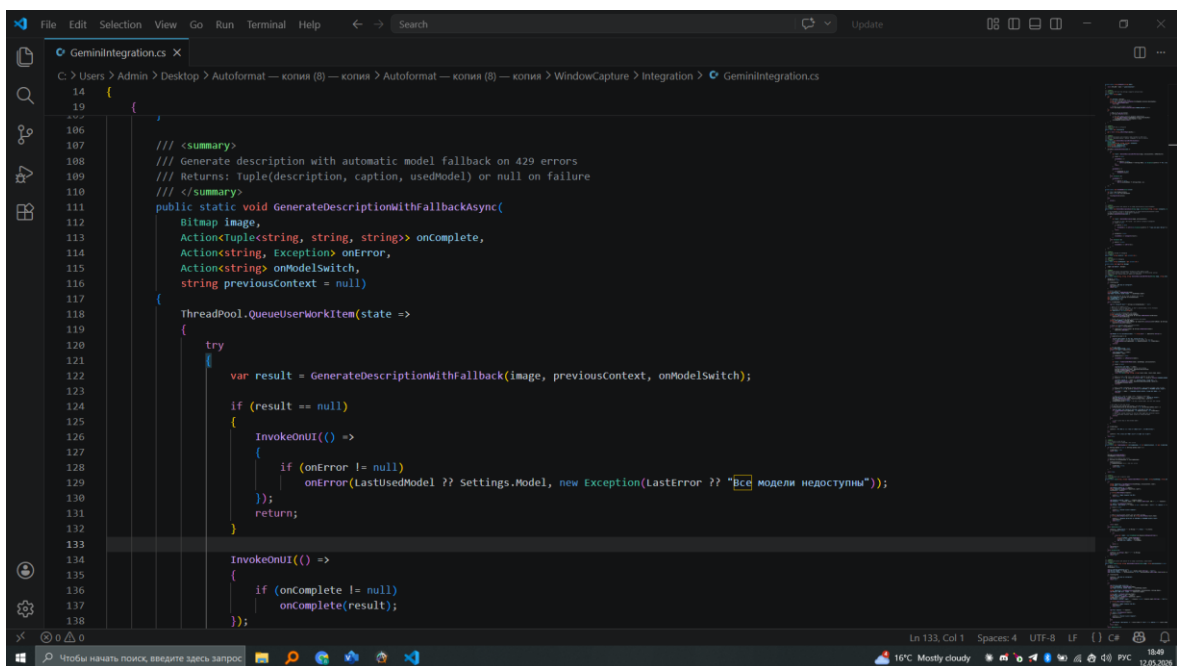
Наименование	Назначение
C# / .NET Framework 4.0	Основной язык программирования и среда выполнения
Компилятор csc.exe	Преобразование исходного кода в исполняемый файл
PowerShell 5 (build.ps1)	Сборочный сценарий, объединяющий 94 модуля
Visual Studio Code	Среда редактирования исходного кода
Microsoft Word (COM)	Интеграция с открытым документом
Google Gemini API	Облачный сервис генерации текстовых описаний
Direct2D, DirectWrite, WIC	Аппаратное ускорение графического вывода
Media Foundation, WASAPI	Кодирование видео и захват системного звука
Python 3.9	Подготовка словаря и обучение моделей подсистемы Т9

3.2. Отладка программы

В ходе разработки приложения была применена комбинация ручной отладки в редакторе Visual Studio Code и тестовых консольных утилит из каталога Tools. Логирование выполняется через статический класс `Logger`, записывающий события в текстовые файлы каталога `%APPDATA%\WindowCapture\logs`. Это позволяет воспроизводить и локализовывать ошибки даже после повторного запуска программы. В процессе работы над программой было выявлено и устранено несколько характерных дефектов, описанных ниже.

При первичной интеграции с Microsoft Word была обнаружена ошибка, возникавшая при отсутствии открытого документа: вызов свойства `Application.ActiveDocument` приводил к исключению `COMException` с кодом `0x800A11FD`. Для устранения дефекта в файл `WordIntegration.cs` добавлен предварительный опрос свойства `Application.Documents.Count`: если число открытых документов равно нулю, операция вставки прерывается с пользовательским уведомлением вместо аварийного завершения приложения.

При обработке сетевых запросов к Google Gemini была обнаружена нестабильность, связанная с превышением лимита частоты запросов (HTTP 429): основная модель временно становилась недоступной, что приводило к зависанию пользовательского интерфейса. Для устранения дефекта в файл GeminiIntegration.cs добавлен механизм автоматического переключения на резервные модели семейства Gemini в порядке убывания приоритета. На рисунке 18 показан фрагмент кода с блоком catch, фиксирующим неудачную попытку и инициирующим повторный запрос к следующей модели из списка.



```
111 public static void GenerateDescriptionWithFallbackAsync(
112     Bitmap image,
113     Action<Tuple<string, string, string>> onComplete,
114     Action<string, Exception> onError,
115     Action<string> onModelSwitch,
116     string previousContext = null)
117 {
118     ThreadPool.QueueUserWorkItem(state =>
119     {
120         try
121         {
122             var result = GenerateDescriptionWithFallback(image, previousContext, onModelSwitch);
123
124             if (result == null)
125             {
126                 InvokeOnUI(() =>
127                 {
128                     if (onError != null)
129                         onError>LastUsedModel ?? Settings.Model, new Exception>LastError ?? "Все модели недоступны");
130                 });
131                 return;
132             }
133
134             InvokeOnUI(() =>
135             {
136                 if (onComplete != null)
137                     onComplete(result);
138             });
139         }
140     });
141 }
```

Рисунок 18 - Обработка ошибок и переключение на резервную модель Gemini

При работе с несколькими мониторами с различным значением DPI было выявлено смещение прямоугольника выделения относительно курсора мыши. Ошибка вызывалась запуском приложения в режиме без учёта DPI. Для устранения проблемы в файл Program.cs в начало процедуры Main добавлен вызов SetProcessDpiAwareness с режимом PROCESS_PER_MONITOR_DPI_AWARE, что обеспечивает корректное

соответствие координат снимка экрана и координат курсора независимо от масштабирования системы.

При интенсивной работе приложения наблюдалась утечка GDI-ресурсов, связанная с использованием объектов `Bitmap` и `Graphics` без своевременного освобождения. Диагностика подтвердилась через утилиту `GDI View`: каждый снимок экрана оставлял в памяти по одному дескриптору. Для устранения дефекта во всех точках работы с растровыми изображениями введены конструкции `using`, гарантирующие вызов `Dispose` даже при возникновении исключений в теле блока.

3.3. Защитное программирование

Приложение не работает с банковскими данными, паролями и не передаёт пользовательские файлы третьим лицам, поэтому основной задачей защитного программирования стало обеспечение устойчивости к ошибкам выполнения и непредвиденным сценариям использования. Конфигурационные файлы `settings.ini`, `keybindings.ini` и `titlepage_data.json` хранятся локально в каталоге профиля пользователя и не передаются по сети.

Для защиты от ошибок выполнения во всех точках взаимодействия с внешними компонентами применяется конструкция `try-catch`. Совокупное число защищённых блоков в проекте превышает 78. Особо защищены вызовы COM-автоматизации `Word`, `P/Invoke` к нативным библиотекам `Windows`, операции файлового ввода-вывода настроек и сетевые запросы к `Google Gemini`.

Для защиты от ввода некорректных данных реализована валидация входных значений. При вставке изображения проверяется наличие открытого документа `Microsoft Word` и активного выделения для определения позиции вставки. При обращении к нейросети проверяется наличие сохранённого API-ключа: при отсутствии ключа операция прерывается с уведомлением. При сохранении настроек

контролируются диапазоны числовых параметров: частота кадров видеозаписи ограничена пределами от 1 до 120 кадров в секунду, чувствительность подсистемы T9 - пределами от 1 до 5, степень размытия - от 0 до 30.

Для защиты от случайного срабатывания глобальной горячей клавиши при работе других программ на этапе обработки сообщений низкоуровневого хука WH_KEYBOARD_LL анализируется флаг LLKHF_INJECTED, помечающий клавишу как программно сгенерированную. Программные нажатия игнорируются, в результате чего автоматизация других приложений не вызывает самопроизвольной активации подсистемы захвата экрана.

Дополнительно реализованы: защита от двойного срабатывания горячей клавиши через флаг ignoreActivationUntilRelease, требующий полного отпускания активирующей клавиши перед повторной обработкой; автоматическое сохранение настроек на диск после каждого изменения, что снимает риск потери конфигурации при аварийном завершении; защита окон редактора от случайного закрытия с предупреждающим сообщением при наличии несохранённых аннотаций.

3.4. Характеристики программы

Программа представляет собой настольное приложение для операционной системы Microsoft Windows, поставляемое в виде единственного исполняемого файла WindowCapture.exe. На рисунке 19 показано окно встроенного редактора аннотаций с загруженным тестовым изображением.

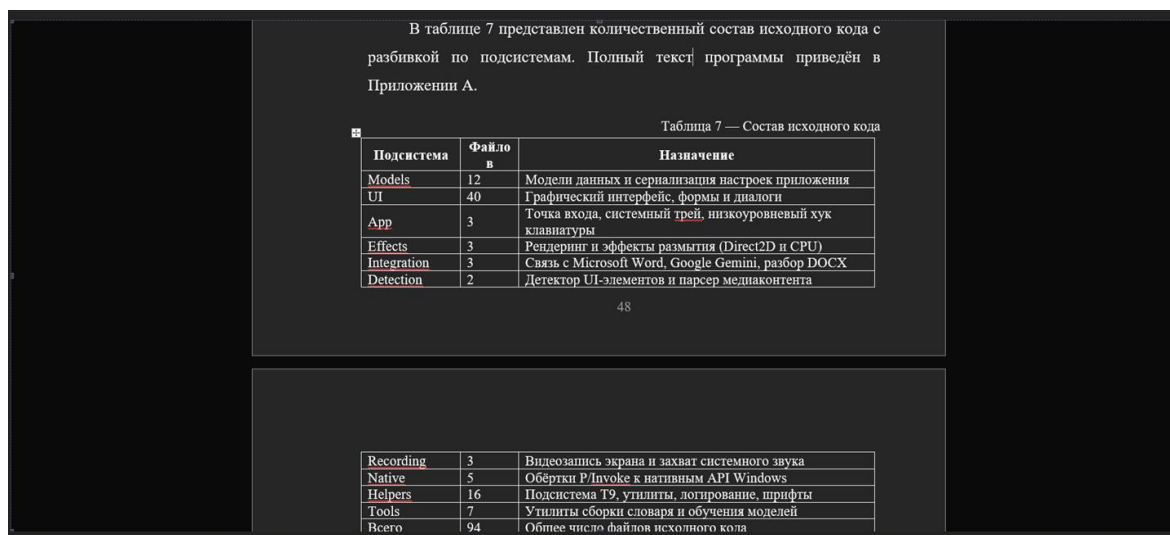


Рисунок 19 - Окно редактора аннотаций приложения WindowCapture

В таблице 7 представлен количественный состав исходного кода с разбивкой по подсистемам. Полный текст программы приведён в Приложении А.

Таблица 7 - Состав исходного кода

Подсистема	Файлов	Назначение
Models	12	Модели данных и сериализация настроек приложения
UI	40	Графический интерфейс, формы и диалоги
App	3	Точка входа, системный трей, низкоуровневый хук клавиатуры
Effects	3	Рендеринг и эффекты размытия (Direct2D и CPU)
Integration	3	Связь с Microsoft Word, Google Gemini, разбор DOCX
Detection	2	Детектор UI-элементов и парсер медиаконтента
Recording	3	Видеозапись экрана и захват системного звука
Native	5	Обёртки P/Invoke к нативным API Windows
Helpers	16	Подсистема T9, утилиты, логирование, шрифты
Tools	7	Утилиты сборки словаря и обучения моделей
Всего	94	Общее число файлов исходного кода

Скомпилированный 64-разрядный исполняемый файл занимает около 0,9 МБ. Для работы программы установка дополнительных компонентов не требуется: все используемые библиотеки входят в состав операционной системы Microsoft Windows. Минимальные и рекомендуемые требования к системе приведены в таблице 8.

Таблица 8 - Характеристики системы

Параметр	Минимально	Рекомендовано
Операционная система	Windows 7 SP1	Windows 10 / 11
.NET Framework	4.0	4.8
Microsoft Word	2010	2019 / 365
Процессор	2 ядра, 2 ГГц	4 ядра, 3 ГГц
Оперативная память	2 ГБ	8 ГБ
Видеоадаптер	DirectX 9.1	DirectX 11
Свободное место на диске	100 МБ	500 МБ
Подключение к Интернет	Требуется	Требуется

Среднее время отклика подсистемы захвата экрана не превышает 100 миллисекунд от момента нажатия глобальной горячей клавиши до появления оверлея выделения. Среднее время генерации текстового описания нейросетью Gemini составляет от 1,5 до 3 секунд в зависимости от модели и текущей нагрузки на сервер. Среднее время вставки изображения в открытый документ Microsoft Word - около 200 миллисекунд.

В рамках технологической части были обоснованы выбор языка программирования, среды разработки и инструментальных средств. Приведены примеры обнаруженных и устранённых дефектов выполнения, описаны применённые методы защитного программирования и количественные характеристики готового программного продукта. Полученное программное обеспечение отвечает требованиям надёжности, безопасности и производительности.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы решена задача разработки настольного приложения для операционной системы Microsoft Windows, обеспечивающего автоматизированный захват снимков экрана, их аннотирование во встроенном графическом редакторе, генерацию текстовых описаний средствами мультимодальной нейросети Google Gemini и автоматическую вставку результата в открытый документ Microsoft Word с оформлением по требованиям ГОСТ. Актуальность темы обусловлена возрастающим объёмом технической и учебной документации, повышением требований к качеству и единообразию оформления иллюстраций, а также появлением мультимодальных языковых моделей, делающих технически реализуемым автоматическое описание содержимого снимков экрана с учётом контекста. Особое внимание было уделено производительности, удобству пользовательского интерфейса и устойчивости работы при отказе сетевых сервисов.

Проект охватывает все стадии жизненного цикла программного обеспечения: от анализа предметной области и архитектурного проектирования до реализации, отладки и подготовки сопроводительной документации. На этапе анализа были изучены существующие коммерческие и свободные решения захвата экрана, выявлены их ограничения и сформулированы функциональные и нефункциональные требования к разрабатываемой системе. На этапе проектирования построены схема сериализуемых объектов модели, функциональная и структурная схемы приложения, схема пользовательского интерфейса, укрупнённый алгоритм и блок-схемы ключевых подсистем: захвата изображения, формирования запроса к нейросети Gemini и автоматической вставки в Microsoft Word.

Для реализации приложения был выбран язык программирования C# в редакции, совместимой с .NET Framework 4.0. Такой выбор обусловлен предустановленностью .NET Framework на всех версиях Windows, начиная с Windows 7 с пакетом обновления 1, что снимает с конечного пользователя необходимость развёртывания дополнительных компонентов. Сборка осуществляется через сценарий PowerShell, вызывающий компилятор csc.exe из состава .NET Framework, что исключает зависимость от установленной Visual Studio. Интеграция с Microsoft Word построена на технологии COM-автоматизации, графический вывод обеспечивается нативными интерфейсами Direct2D, DirectWrite и WIC, аудио- и видеозапись - интерфейсами Media Foundation и WASAPI.

Одним из основных направлений разработки стало внедрение методов защитного программирования, обеспечивающих устойчивость приложения к нештатным ситуациям. Во всех точках взаимодействия с внешними компонентами применяется конструкция try-catch - совокупное число защищённых блоков в проекте превышает 78. Сетевые запросы к Gemini защищены механизмом автоматического переключения на резервные модели в случае превышения лимита частоты запросов или временной недоступности основной модели. Глобальная горячая клавиша захвата защищена от программных нажатий через анализ флага LLKHF_INJECTED низкоуровневого хука WH_KEYBOARD_LL. Локальное хранение конфигурации в файлах settings.ini, keybindings.ini и titlepage_data.json исключает передачу пользовательских данных третьим лицам.

Разработанное программное обеспечение реализует следующий набор функций: быстрый захват снимков экрана с интеллектуальным выделением областей на основе оператора Собеля и алгоритма заливки по цветовому сходству; встроенный графический редактор аннотаций со

стрелками на основе кубических кривых Безье, прямоугольными и эллиптическими подсветками, нумерованными маркерами, текстовыми блоками и комментариями в выносках; генерацию текстовых описаний снимков экрана средствами облачной нейросети Google Gemini с автоматическим переключением между резервными моделями; автоматическую вставку аннотированных снимков в активный документ Microsoft Word с описательным абзацем, центрированием изображения и подписью «Рисунок N - наименование» в соответствии с требованиями ГОСТ; автоматическую коррекцию русскоязычного текста на основе ансамбля из фильтра Блума, морфологического анализатора OpenCorpora и нейросетевой модели уровня символов.

Особое место в системе занимает интеграция с мультимодальной нейросетью Google Gemini. Соответствующая подсистема инкапсулирует REST-взаимодействие с облачным API, кодирование изображения в Base64, формирование запроса с шаблонными подстановками (номер рисунка, заголовок, контекст документа), разбор JSON-ответа и извлечение текстового описания. Механизм поэтапного переключения между моделями семейства Gemini в порядке убывания приоритета (gemini-2.5-flash, gemini-2.5-flash-lite, gemini-2.0-flash и далее) обеспечивает работоспособность приложения даже при недоступности отдельных моделей или превышении квот, а отслеживание статуса каждой модели исключает повторные обращения к временно недоступным сервисам в течение заданного интервала ожидания.

Особое внимание было уделено процессу отладки и тестирования. Проведены тесты, описанные во внешней спецификации, которые подтвердили стабильную работу всех ключевых подсистем: захвата изображения с одним и несколькими мониторами с различным значением DPI, генерации описаний нейросетью Gemini с

переключением на резервные модели, вставки в открытый документ Microsoft Word, сохранения и загрузки конфигурации, обработки горячей клавиши при наличии других глобальных хуков в системе. Все обнаруженные в процессе разработки дефекты - исключение COMException при отсутствии открытого документа, утечка GDI-ресурсов при работе с растровыми изображениями, смещение оверлея на мониторах с различным значением DPI и зависание интерфейса при превышении лимита Gemini - были устранены до сдачи проекта.

По итогам выполнения выпускной квалификационной работы был достигнут комплексный результат, включающий в себя как теоретические аспекты проектирования настольных приложений с применением мультимодальных нейросетевых сервисов, так и практическую реализацию, соответствующую современным стандартам разработки на платформе Microsoft .NET. Полученный опыт позволил закрепить знания в области взаимодействия с нативными API Windows через P/Invoke и COM-автоматизацию, проектирования низкоуровневых клавиатурных хуков, обеспечения аппаратного ускорения графического вывода средствами Direct2D, а также организации взаимодействия с облачными нейросетевыми сервисами по REST-протоколу.

Реализованное программное обеспечение является стабильной основой для дальнейшего развития. В перспективе возможно расширение функциональности за счёт интеграции с другими мультимодальными нейросетевыми сервисами (OpenAI GPT-4o, Anthropic Claude, локальными моделями семейства LLaMA), добавления подсистемы распознавания текста на снимках экрана средствами OCR, реализации синхронизации сделанных снимков и аннотаций через облачное хранилище, поддержки автоматического оформления отчётов по другим стандартам (IEEE, СТП), а также расширения встроенного графического редактора аннотаций функциями совместного

редактирования и истории изменений. Перечисленные направления позволят проекту продолжать развиваться в соответствии с актуальными потребностями технической документации и эволюцией возможностей мультимодальных нейросетей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Российская Федерация. Законы. Об информации, информационных технологиях и о защите информации: Федеральный закон № 149-ФЗ от 27.07.2006 (действующая редакция 2024 г.). - Текст: электронный. - URL: http://www.consultant.ru/document/cons_doc_LAW_61798/ (дата обращения: 10.05.2026).
2. Российская Федерация. Законы. О персональных данных: Федеральный закон № 152-ФЗ от 27.07.2006 (действующая редакция 2024 г.). - Текст: электронный. - URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 10.05.2026).
3. ГОСТ Р ИСО/МЭК 27001-2021. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. - Москва: Российский институт стандартизации, 2021. - 35 с.
4. ГОСТ Р 56939-2024. Защита информации. Разработка безопасного программного обеспечения. Общие требования. - Москва: Российский институт стандартизации, 2024. - 28 с.
5. ГОСТ Р 70262.1-2022. Информационные технологии. Искусственный интеллект. Понятия и терминология. - Москва: Российский институт стандартизации, 2022. - 24 с.
6. Албахари, Д. С# 12 и .NET 8. Справочник. Полное описание языка / Д. Албахари; перевод с английского. - 9-е изд. - Москва: Диалектика, 2024. - 1216 с. - ISBN 978-5-907720-32-7.

7. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс; перевод с английского. - Москва: Техносфера, 2021. - 1104 с. - ISBN 978-5-94836-657-0.
8. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль; перевод с английского. - Москва: ДМК Пресс, 2021. - 652 с. - ISBN 978-5-97060-618-6.
9. Жерон, О. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow / О. Жерон; перевод с английского. - 4-е изд. - Санкт-Петербург: Питер, 2024. - 928 с. - ISBN 978-5-4461-2128-2.
10. Кадури́н, А. А. Глубокое обучение и нейронные сети: учебник / А. А. Кадури́н, Е. О. Архангельская. - Санкт-Петербург: Лань, 2022. - 348 с. - ISBN 978-5-507-43156-2.
11. Кормен, Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; перевод с английского. - 3-е изд. - Москва: Вильямс, 2021. - 1328 с. - ISBN 978-5-907144-87-1.
12. Лейн, Х. Обработка естественного языка в действии / Х. Лейн, Х. Хапке, К. Ховард; перевод с английского. - Санкт-Петербург: Питер, 2021. - 576 с. - ISBN 978-5-4461-1462-8.
13. Прайс, М. C# 12 и .NET 8 для современной кроссплатформенной разработки / М. Прайс; перевод с английского. - Санкт-Петербург: Питер, 2024. - 720 с. - ISBN 978-5-4461-2317-0.
14. Скит, Дж. C# для профессионалов. Тонкости программирования / Дж. Скит; перевод с английского И. В. Берштейна. - 4-е изд. - Москва: Вильямс, 2021. - 608 с. - ISBN 978-5-907203-77-7.
15. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман; перевод с английского. - Москва: Лаборатория знаний, 2022. - 763 с. - ISBN 978-5-93208-562-5.

16. Шилдт, Г. С# 10. Полное руководство / Г. Шилдт; перевод с английского. - Москва: Диалектика, 2022. - 1056 с. - ISBN 978-5-907458-87-2.
17. .NET Framework Documentation. Microsoft Learn. - Текст: электронный. - URL: <https://learn.microsoft.com/ru-ru/dotnet/framework/> (дата обращения: 10.05.2026).
18. Direct2D Documentation. Microsoft Learn. - Текст: электронный. - URL: <https://learn.microsoft.com/ru-ru/windows/win32/direct2d/> (дата обращения: 10.05.2026).
19. Microsoft Word Object Model Reference. Microsoft Learn. - Текст: электронный. - URL: <https://learn.microsoft.com/ru-ru/office/vba/api/overview/word/object-model> (дата обращения: 10.05.2026).
20. Gemini API Documentation. Google AI for Developers. - Текст: электронный. - URL: <https://ai.google.dev/gemini-api/docs> (дата обращения: 10.05.2026).

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ	62
ПРИЛОЖЕНИЕ Б. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	69
ПРИЛОЖЕНИЕ В. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	77
ПРИЛОЖЕНИЕ Г. ЗАПРОСЫ (ПРОМТ) К НЕЙРОСЕТЯМ.....	89

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ

АННОТАЦИЯ

Настоящее приложение является частью выпускной квалификационной работы и содержит исходный текст разработанного настольного приложения для генерации текстовых описаний снимков экрана и их автоматической вставки в открытый документ Microsoft Word. Приведены описание состава модулей, инструментальной среды разработки, а также фрагменты исходного кода ключевых подсистем приложения.

ВВЕДЕНИЕ

В качестве среды для написания исходного кода был выбран редактор Visual Studio Code. Visual Studio Code - бесплатный кроссплатформенный редактор кода, разработанный компанией Microsoft. Редактор обеспечивает подсветку синтаксиса для языка C#, навигацию по символам исходного кода, контекстные подсказки и встроенную интеграцию с системой контроля версий Git.

Исходный код приложения написан на языке программирования C# в редакции, совместимой с .NET Framework 4.0. Сборка выполняется через сценарий PowerShell с использованием компилятора csc.exe из состава операционной системы Microsoft Windows, без необходимости установки крупных интегрированных сред разработки.

Программа состоит из 94 модулей собственной разработки. Суммарный объем исходного кода составляет более 33 тысяч строк на языке C#. Используются стандартные библиотеки System.dll, System.Drawing.dll, System.Windows.Forms.dll, System.Runtime.Serialization.dll, а также нативные библиотеки Windows: user32.dll, gdi32.dll, dwmapi.dll, d2d1.dll, dwrite.dll, mfplat.dll.

ПЕРЕЧЕНЬ МОДУЛЕЙ

Программа разделена на десять подсистем: точка входа, приложение и системный трей, пользовательский интерфейс, модели данных, графические эффекты, интеграция с внешними сервисами, детекция, запись видео, нативные обёртки и вспомогательные утилиты. Сводный перечень основных модулей представлен в таблице А.1.

А.1. Перечень основных модулей

Имя модуля	Назначение	Тип
Program.cs	Точка входа, настройка DPI, маршрутизация	авторский
App\TrayApp.cs	Системный трей, контекстное меню	авторский
App\Controller.cs	Низкоуровневый хук клавиатуры, запуск захвата	авторский
UI\FrozenOverlay.cs	Замороженный оверлей выделения области	авторский
UI\EditorForm.cs	Главное окно редактора аннотаций	авторский
UI\AnnotationCanvas.cs	Канвас отрисовки аннотаций (D2D/GDI+)	авторский
UI\WordSidePanel.cs	Боковая панель вставки в Microsoft Word	авторский
UI\SettingsDialog.cs	Окно настроек приложения	авторский
Integration\WordIntegration.cs	COM-автоматизация Microsoft Word	авторский
Integration\GeminiIntegration.cs	REST-клиент к Google Gemini API	авторский
Detection\Detector.cs	Оператор Собеля и заливка по цветовому сходству	авторский
Effects\D2DRenderer.cs	Аппаратное ускорение через Direct2D	авторский
Helpers\TextProcessor.cs	Автокоррекция русского текста (Т9)	авторский
Native\WinApi.cs	Обёртки P/Invoke к WinAPI	смешанный
Native\Direct2D.cs	COM-интерфейсы Direct2D и DirectWrite	смешанный

ТЕКСТ МОДУЛЯ Program.cs

Точка входа приложения. Выполняет настройку DPI-aware режима, загрузку конфигурационных файлов и маршрутизацию между режимом системного трея и режимом просмотра указанного файла.

```
using System;
using System.IO;
using System.Windows.Forms;
using WindowCapture.App;
using WindowCapture.Helpers;
using WindowCapture.Models;
using WindowCapture.UI;

namespace WindowCapture
{
    static class Program
    {
        [STAThread]
        static void Main(string[] args)
        {
            // Включение DPI-aware режима для корректного захвата
            // на дисплеях с масштабированием отличным от 100%
            try { Native.WinApi.SetProcessDpiAwareness(
                Native.WinApi.PROCESS_PER_MONITOR_DPI_AWARE); }
            catch { try { Native.WinApi.SetProcessDPIAware(); }
                catch { } }

            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            // Загрузка пользовательских настроек
            Settings.Load();
            KeyBindings.Load();

            // Если передан путь к файлу - открыть в режиме просмотра
            if (args != null && args.Length > 0 && File.Exists(args[0]))
            {
                string ext =
                Path.GetExtension(args[0]).ToLowerInvariant();
                if (MediaTypes.IsAudio(ext))
                    Application.Run(new AudioPlayerForm(args[0]));
                else
                    Application.Run(new EditorForm(args[0]));
                return;
            }

            // Стандартный запуск - приложение системного трея
            Application.Run(new TrayApp());
        }
    }
}
```

ТЕКСТ МОДУЛЯ Controller.cs (фрагмент)

Обработчик низкоуровневого хука клавиатуры WH_KEYBOARD_LL. Анализирует флаг LLKHF_INJECTED для защиты от программных нажатий со стороны других приложений и активирует подсистему захвата экрана при нажатии глобальной горячей клавиши.

```
// Фрагмент Controller.cs - обработчик глобального хука клавиатуры
// (WH_KEYBOARD_LL) с защитой от инжектированных нажатий
```

```
private IntPtr KeyboardHookCallback(int nCode, IntPtr wParam,
                                    IntPtr lParam)
{
    if (nCode >= 0)
    {
        var kb = (KBDLLHOOKSTRUCT)Marshal.PtrToStructure(
            lParam, typeof(KBDLLHOOKSTRUCT));

        int wmsg = wParam.ToInt32();
        int vkc = (int)kb.vkCode;

        // Анализ флага LLKHF_INJECTED для защиты от
        // программных нажатий со стороны других приложений
        bool isInjected = (kb.flags & LLKHF_INJECTED) != 0;

        // Проверка совпадения с активирующей клавишей.
        // Для модификаторов учитываются левый и правый варианты
        bool isActivationKey = false;
        if (Settings.ActivationKey == 0xA4 ||
            Settings.ActivationKey == 0xA5)
        {
            isActivationKey = vkc == 0xA4 || vkc == 0xA5;
        }

        if (isActivationKey)
        {
            bool isKeyDown = (wmsg == WinApi.WM_KEYDOWN ||
                              wmsg == WinApi.WM_SYSKEYDOWN);
            bool isKeyUp = (wmsg == WinApi.WM_KEYUP ||
                            wmsg == WinApi.WM_SYSKEYUP);

            // Защита от повторного срабатывания: после отпускания
            // клавиши сбрасывается флаг блокировки
            if (isKeyUp)
            {
                ignoreActivationUntilRelease = false;
                if (active) Deactivate();
            }
            else if (isKeyDown && !active &&
                    !ignoreActivationUntilRelease)
            {
                // Запуск захвата экрана
                isRecordingMode =
                    (WinApi.GetKeyState(0x10) & 0x8000) != 0;
                Activate();
            }
        }
    }
}
```

```

    }
}
return WinApi.CallNextHookEx(hook, nCode, wParam, lParam);
}

```

ТЕКСТ МОДУЛЯ GeminiIntegration.cs (фрагмент)

Асинхронный вызов нейросети Google Gemini с поэтапным переключением на резервные модели семейства Gemini при недоступности основной модели. Внутренний механизм GenerateDescriptionWithFallback по очереди обращается к моделям из списка FallbackModels.

```

// Фрагмент GeminiIntegration.cs - асинхронный вызов
// нейросети Gemini с поэтапным переключением на резервные модели

public static void GenerateDescriptionWithFallbackAsync(
    Bitmap image,
    Action<Tuple<string, string, string>> onComplete,
    Action<string, Exception> onError,
    Action<string> onModelSwitch,
    string previousContext = null)
{
    ThreadPool.QueueUserWorkItem(state =>
    {
        try
        {
            var result = GenerateDescriptionWithFallback(
                image, previousContext, onModelSwitch);

            if (result == null)
            {
                InvokeOnUI(() =>
                {
                    if (onError != null)
                        onError>LastUsedModel ?? Settings.Model,
                            new Exception>LastError ??
                                "Все модели недоступны");
                });
                return;
            }

            InvokeOnUI(() =>
            {
                if (onComplete != null)
                    onComplete(result);
            });
        }
        catch (Exception ex)
        {
            // Перехват любых исключений с уведомлением UI-потока
            InvokeOnUI(() =>
            {
                if (onError != null)
                    onError>LastUsedModel ?? Settings.Model, ex);
            });
        }
    }
}

```

```
});  
}
```

ТЕКСТ МОДУЛЯ WordIntegration.cs (фрагмент)

Вставка аннотированного снимка экрана в активный документ Microsoft Word через интерфейс COM-автоматизации. Перед изображением вставляется описательный абзац, после - подпись формата «Рисунок N - наименование» с автоматической нумерацией.

```
// Фрагмент WordIntegration.cs - вставка изображения  
// в активный документ Microsoft Word через COM-автоматизацию  
  
public static bool InsertImage(string imagePath,  
                               string caption,  
                               string description)  
{  
    try  
    {  
        // Получение запущенного экземпляра Word  
        Type wordType = Type.GetTypeFromProgID("Word.Application");  
        dynamic wordApp = Marshal.GetActiveObject("Word.Application");  
  
        // Защитная проверка: документ должен быть открыт  
        if (wordApp.Documents.Count == 0)  
        {  
            ShowError("Откройте документ Microsoft Word");  
            return false;  
        }  
  
        dynamic doc = wordApp.ActiveDocument;  
        dynamic selection = wordApp.Selection;  
  
        // Подсчёт существующих рисунков для нумерации  
        int picNum = doc.InlineShapes.Count + 1;  
  
        // Вставка описательного абзаца перед изображением  
        if (!string.IsNullOrEmpty(description))  
        {  
            selection.TypeText(description);  
            selection.TypeParagraph();  
        }  
  
        // Вставка изображения с центрированием  
        dynamic shape = selection.InlineShapes.AddPicture(imagePath);  
        selection.ParagraphFormat.Alignment = 1; // wdAlignCenter  
  
        // Вставка подписи в формате ГОСТ  
        selection.TypeParagraph();  
        selection.TypeText("Рисунок " + picNum + " - " + caption);  
        return true;  
    }  
    catch (Exception ex)  
    {  
        ShowError("Ошибка вставки: " + ex.Message);  
        return false;  
    }  
}
```

ТЕКСТ МОДУЛЯ Detector.cs (фрагмент)

Поиск связной области под курсором мыши с использованием оператора Собеля для вычисления карты границ и алгоритма BFS-заливки по цветовому сходству. Карта границ кэшируется для ускорения повторных вычислений с тем же снимком экрана.

```
// Фрагмент Detector.cs - детекция области под курсором
// с применением оператора Собеля и заливки по цветовому сходству

public unsafe GraphicsPath Detect(Point cur, int expandLevel)
{
    if (screen == null || cur.X < 0 || cur.X >= w ||
        cur.Y < 0 || cur.Y >= h) return null;

    var data = screen.LockBits(
        new Rectangle(0, 0, w, h),
        ImageLockMode.ReadOnly,
        PixelFormat.Format32bppArgb);

    try
    {
        byte* ptr = (byte*)data.Scan0;
        int stride = data.Stride;

        // Кэширование карты границ оператора Собеля для
        // повторных вычислений с тем же снимком экрана
        if (cachedEdgeMap == null)
        {
            cachedEdgeMap = ComputeEdgeMap(ptr, stride);
            cachedStride = stride;
            cachedPixels = new byte[h * stride];
            Marshal.Copy(data.Scan0, cachedPixels, 0,
                cachedPixels.Length);
        }

        // Поиск связной области BFS-заливкой по цветовому сходству
        var result = DetectRegion(ptr, stride, cachedEdgeMap,
            cur, expandLevel);

        return result;
    }
    finally
    {
        screen.UnlockBits(data);
    }
}
```

ПРИЛОЖЕНИЕ Б. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

АННОТАЦИЯ

Настоящий документ является руководством пользователя настольного приложения для автоматической генерации текстовых описаний снимков экрана и их вставки в открытый документ Microsoft Word. Руководство содержит сведения о назначении приложения, технических и программных требованиях, порядке установки, описание основных операций, действия в аварийных ситуациях и рекомендации по освоению.

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

**Разработка настольного приложения для генерации текстовых
описаний снимков экрана и их автоматической вставки в открытый
документ Microsoft Word**

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

На 7 листах

2026

СОДЕРЖАНИЕ

1. Введение.....	3
1.1. Область применения.....	3
1.2. Краткое описание возможностей.....	4
1.3. Уровень подготовки пользователя.....	4
2. Назначение и условия применения.....	4
2.1. Предмет автоматизации.....	4
2.2. Программное обеспечение.....	4
3. Подготовка к работе.....	5
4. Описание операций.....	5
4.1. Запуск приложения.....	5
4.2. Использование редактора.....	6
4.3. Вставка в Microsoft Word.....	6
4.4. Поиск файлов.....	7
5. Аварийные ситуации.....	7
6. Рекомендации по освоению.....	8

1. Введение

1.1. Область применения

Приложение предназначено для специалистов, готовящих техническую и учебную документацию с большим количеством иллюстраций: технических писателей, преподавателей, аналитиков, инженеров-разработчиков и студентов. Приложение автоматизирует повторяющиеся операции захвата снимков экрана, их аннотирования и оформления в соответствии с требованиями ГОСТ.

1.2. Краткое описание возможностей

Приложение обеспечивает: быстрый захват снимков экрана с интеллектуальным выделением областей; встроенный графический редактор для аннотирования; автоматическую генерацию текстовых описаний средствами нейросети Google Gemini; автоматическую вставку аннотированных снимков в активный документ Microsoft Word с описательным абзацем и подписью; автоматическую коррекцию набираемого русскоязычного текста.

1.3. Уровень подготовки пользователя

Для работы с приложением требуются базовые навыки работы с операционной системой Microsoft Windows и текстовым редактором Microsoft Word. Специальная подготовка в области программирования или информационной безопасности не требуется.

2. Назначение и условия применения

2.1. Технические средства

Минимальные требования: процессор не менее 2 ядер с тактовой частотой 2 ГГц; оперативная память 2 ГБ; свободное место на жёстком диске 100 МБ; видеоадаптер с поддержкой DirectX версии 9.1 и выше; разрешение экрана не менее 1024x768. Рекомендуемые требования: процессор 4 ядра с тактовой частотой 3 ГГц; оперативная память 8 ГБ; свободное место на жёстком диске 500 МБ; видеоадаптер с поддержкой DirectX 11.

2.2. Программное обеспечение

Операционная система: Microsoft Windows 7 с пакетом обновления 1, Windows 8.1, Windows 10 или Windows 11. Установленный пакет .NET Framework версии 4.0 и выше (предустановлен на всех современных версиях Windows). Установленный Microsoft Word версии 2010 и выше для функции автоматической вставки. Постоянное подключение к сети Интернет для функции генерации описаний нейросетью Google Gemini.

3. Подготовка к работе

Дистрибутив приложения распространяется в виде единственного исполняемого файла WindowCapture.exe размером около 0,9 МБ. Установка как таковая не требуется: достаточно скопировать файл в произвольный каталог на жёстком диске и запустить его двойным щелчком мыши.

При первом запуске приложение создаёт каталог %APPDATA%\WindowCapture для хранения конфигурационных файлов settings.ini, keybindings.ini, titlepage_data.json и журналов работы. После запуска в системном трее отображается иконка приложения с подсказкой «WindowCapture. Press Alt to capture».

Для использования функции автоматической генерации описаний необходимо ввести API-ключ Google Gemini. Получение ключа выполняется в личном кабинете Google AI Studio по адресу ai.google.dev. Ввод ключа производится в окне настроек приложения на вкладке Keys.

Проверка работоспособности: после установки нажмите клавишу Alt в любом приложении. На экране должен появиться замороженный полупрозрачный оверлей с возможностью выделения области экрана. Если оверлей не появляется, проверьте, что приложение запущено и его иконка присутствует в системном трее.

4. Описание операций

4.1. Захват снимка экрана

Для захвата снимка экрана нажмите глобальную горячую клавишу Alt. На экране появится замороженный оверлей с текущим изображением рабочего стола. По умолчанию выделение области выполняется автоматически по цветовому сходству на основе оператора Собеля и алгоритма заливки. Для выделения произвольной прямоугольной области удерживайте левую кнопку мыши и перемещайте курсор. Прокрутка колеса мыши расширяет выделенную область до целого окна, строки или экрана. Подтверждение выбора - щелчок левой кнопкой мыши, отмена - клавиша Esc.

4.2. Аннотирование снимка

После подтверждения выделения откроется встроенный графический редактор аннотаций. Для нанесения стрелок удерживайте правую кнопку мыши. Прямоугольные подсветки наносятся левой кнопкой мыши с перетаскиванием. Маркеры с автоматической нумерацией ставятся средней кнопкой мыши. Текстовые блоки

добавляются двойным щелчком левой кнопки мыши. Комментарии в выноске - двойной щелчок правой кнопки. Отмена последнего действия - Ctrl+Z.

4.3. Вставка в Microsoft Word

Для вставки аннотированного снимка в открытый документ Microsoft Word установите курсор в нужную позицию документа, затем нажмите и удерживайте клавишу пробела в течение примерно 250 миллисекунд. Откроется боковая панель с полями для ввода описательного абзаца и подписи. После заполнения полей и нажатия кнопки «Вставить» аннотированный снимок будет вставлен в документ с центрированием, описательным абзацем и подписью формата «Рисунок N - наименование».

Для быстрой вставки с автоматически сгенерированным описанием нажмите Ctrl+Пробел. Описание будет получено нейросетью Google Gemini на основе анализа содержимого снимка экрана и текущего контекста документа.

4.4. Окно настроек

Окно настроек открывается щелчком правой кнопкой мыши по иконке приложения в системном трее с последующим выбором пункта Settings, либо двойным щелчком левой кнопки по иконке. Окно содержит три вкладки: General (общие настройки), Effects (параметры подсветок и размытия), Keys (горячие клавиши и API-ключи нейросети Gemini).

5. Аварийные ситуации

При отсутствии открытого документа Microsoft Word попытка вставки изображения завершится сообщением «Откройте документ

Microsoft Word». Для устранения откройте любой документ формата .docx и повторите операцию.

При недоступности нейросети Google Gemini (отсутствие подключения к сети Интернет, превышение лимита частоты запросов, истечение срока действия API-ключа) приложение автоматически переключается на резервные модели семейства Gemini. Если все модели недоступны, отображается сообщение с указанием возможной причины. Для устранения проверьте подключение к сети Интернет и корректность введённого API-ключа.

При обнаружении некорректных данных в конфигурационных файлах settings.ini, keybindings.ini или titlepage_data.json приложение использует значения по умолчанию и записывает сообщение об ошибке в журнал по пути %APPDATA%\WindowCapture\logs. Для сброса всех настроек до значений по умолчанию удалите указанные файлы и перезапустите приложение.

6. Рекомендации по освоению

Освоение приложения рекомендуется начать с проверки основной операции захвата снимка экрана: запустить любое приложение с произвольным содержимым, нажать клавишу Alt и опробовать различные способы выделения области (автоматический, произвольный прямоугольник, расширение колесом мыши). После освоения захвата следует ознакомиться с инструментами графического редактора аннотаций, опробовав каждый из инструментов по очереди. На завершающем этапе рекомендуется настроить интеграцию с Microsoft Word и нейросетью Google Gemini в окне настроек приложения. После указанной последовательности приложение готово к повседневному применению в задачах подготовки технической документации.

ПРИЛОЖЕНИЕ В. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

АННОТАЦИЯ

Настоящий документ описывает требования к разработке настольного приложения для генерации текстовых описаний снимков экрана и их автоматической вставки в открытый документ Microsoft Word. В нём указаны цели разработки, состав пользователей системы, а также функции и особенности, которыми приложение должно обладать. Документ содержит сведения об используемых технологиях и требованиях к составу, виду и принципам работы программного продукта. Также здесь описаны условия, при которых проект можно считать завершённым, и порядок ввода приложения в эксплуатацию. Техническое задание служит ориентиром для всех участников разработки — разработчика, руководителя выпускной квалификационной работы и членов экзаменационной комиссии. Документ оформлен в соответствии с ГОСТ 19.201-78.

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

**Разработка настольного приложения для генерации текстовых
описаний снимков экрана и их автоматической вставки в открытый
документ Microsoft Word**

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 16 листах

2026

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ.....	5
1.1. Назначение документа.....	5
1.2. Краткие сведения о разработчике	5
1.3. Плановые сроки начала и окончания работ	5
1.4. Порядок оформления и предъявления результатов работ	5
2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ ПРИЛОЖЕНИЯ	6
2.1. Назначение приложения.....	6
2.2. Цели создания приложения.....	6
2.2.1. Основные цели	6
2.2.2. Целевая аудитория	6
3. ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ	7
3.1. Требования к приложению в целом	7
3.1.1. Требования к структуре и функционированию	7
3.1.2. Требования к персоналу	7
3.1.3. Требования к сохранности информации	8
3.2. Требования к функциям, выполняемым приложением	8
3.2.1. Основные требования	8
3.3. Требования к видам обеспечения.....	9
3.3.1. Требования к информационному обеспечению.....	9
3.3.2. Требования к программному обеспечению.....	9

3.3.3. Требования к аппаратному обеспечению	10
4. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ ПРИЛОЖЕНИЯ.....	10
5. ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ.....	11
5.1. Виды, состав, объём и методы испытаний	11
5.2. Общие требования к приёмке приложения	11
6. ТРЕБОВАНИЯ К ВВОДУ В ЭКСПЛУАТАЦИЮ.....	12

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Назначение документа

Настоящий документ содержит требования к разработке программного продукта — настольного приложения для генерации текстовых описаний снимков экрана и их автоматической вставки в открытый документ Microsoft Word. Документ предназначен для согласования объёма и состава работ между разработчиком и руководителем выпускной квалификационной работы и используется в качестве основания для приёмки готового программного продукта.

1.2. Краткие сведения о разработчике

Разработка приложения выполнена в рамках выпускной квалификационной работы студентом Московского приборостроительного техникума федерального государственного бюджетного образовательного учреждения высшего образования «Российский экономический университет имени Г.В. Плеханова». Приложение разрабатывается для собственных нужд автора, преподавателей, технических писателей и инженеров-документаторов, регулярно подготавливающих отчётную документацию с большим количеством иллюстраций.

1.3. Плановые сроки начала и окончания работ

Начало работ — 01.09.2025, окончание работ — 31.05.2026. Сроки могут быть скорректированы по согласованию с руководителем выпускной квалификационной работы.

1.4. Порядок оформления и предъявления результатов работ

Все результаты выполнения работ оформляются в виде технической документации (техническое задание, пояснительная

записка, руководство пользователя) и комплекта исходного кода с инструкцией по сборке исполняемого файла. Документация оформляется в соответствии с требованиями ГОСТ 7.32-2017 и ГОСТ 19.106-78. Документация и исходный код предъявляются руководителю выпускной квалификационной работы в установленный кафедрой срок.

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ ПРИЛОЖЕНИЯ

2.1. Назначение приложения

Приложение предназначено для автоматизации повседневной деятельности технических писателей и инженеров-разработчиков, связанной с подготовкой отчётной документации с большим количеством снимков экрана. Приложение обеспечивает оперативный захват произвольной области экрана, нанесение поясняющих аннотаций (стрелки, подсветки, маркеры, текстовые блоки и комментарии в выносках), генерацию текстового описания снимка средствами облачной мультимодальной языковой модели и автоматическую вставку готового снимка с подписью в активный документ Microsoft Word.

2.2. Цели создания приложения

2.2.1. Основные цели

Основными целями создания приложения являются: сокращение времени подготовки технической документации с большим количеством иллюстраций; повышение качества и единообразия подписей к снимкам экрана; снижение количества ошибок ручного ввода описаний; обеспечение возможности оформления документации в соответствии с требованиями ГОСТ по нумерации и подписи рисунков.

2.2.2. Целевая аудитория

Целевой аудиторией приложения являются технические писатели, инженеры-разработчики программного обеспечения, преподаватели информационных дисциплин, методисты, аналитики и иные специалисты, регулярно подготавливающие отчётные документы с большим количеством снимков экрана. Дополнительная целевая аудитория — студенты, выполняющие курсовые и выпускные квалификационные работы по техническим направлениям подготовки.

3. ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ

3.1. Требования к приложению в целом

3.1.1. Требования к структуре и функционированию

Приложение должно быть выполнено в виде самостоятельного исполняемого файла для операционной системы Microsoft Windows версии 7 SP1 и выше. Приложение должно состоять из подсистемы захвата снимков экрана с замороженным оверлеем, графического редактора аннотаций, подсистемы обращения к облачной мультимодальной языковой модели для генерации текстового описания, подсистемы автоматической вставки результата в открытый документ Microsoft Word средствами COM-автоматизации, подсистемы записи экранного видео со звуком, а также подсистемы настроек и хранения пользовательских предпочтений.

3.1.2. Требования к персоналу

Для эксплуатации приложения от пользователя требуются базовые навыки работы с операционной системой Microsoft Windows, владение пакетом Microsoft Office на уровне уверенного пользователя, а также базовое знание правил оформления технической документации.

Специальная подготовка не требуется. Освоение приложения возможно по руководству пользователя, приведённому в Приложении Б.

3.1.3. Требования к сохранности информации

Приложение должно обеспечивать сохранность пользовательских настроек между сеансами работы за счёт хранения их в каталоге AppData пользователя. Снимки экрана и сгенерированные описания не сохраняются автоматически на постоянной основе, а передаются непосредственно в открытый документ Microsoft Word. Передача данных в облачную мультимодальную модель производится по защищённому каналу HTTPS. Передача API-ключа осуществляется только на серверы поставщика мультимодальной модели.

3.2. Требования к функциям, выполняемым приложением

3.2.1. Основные требования

Приложение должно обеспечивать выполнение следующих функций: захват произвольной области экрана по нажатию глобальной горячей клавиши Alt; отображение замороженного оверлея с текущим изображением рабочего стола и средствами выделения области (автоматический режим с использованием оператора Собеля и заливки по цветовому сходству, режим произвольного прямоугольника, режим расширения выделенной области до окна, строки или экрана через прокрутку колеса указательного устройства); нанесение поясняющих аннотаций на снимок (стрелки на основе кубических кривых Безье, прямоугольные и эллиптические подсветки, нумерованные маркеры, текстовые блоки и комментарии в выносках); применение эффектов размытия по Гауссу и размытия в движении к выделенным областям; отмену и повтор операций редактирования через стек операций; запрос текстового описания снимка к облачной мультимодальной модели Google Gemini с учётом контекста элемента пользовательского

интерфейса под курсором; автоматическую коррекцию русскоязычного текста на основе фильтра Блума, морфологического анализатора OpenCorpora и нейросетевой модели уровня символов; автоматическую нумерацию рисунков и вставку готового снимка с подписью в открытый документ Microsoft Word; запись экранного видео со звуком в форматах MP4, WMV и AVI; настройку параметров приложения (горячие клавиши, цветовая схема, шрифт, кегль, путь к шаблону, API-ключ облачной модели).

3.3. Требования к видам обеспечения

3.3.1. Требования к информационному обеспечению

Пользовательские настройки приложения хранятся в виде файлов формата INI и JSON в каталоге AppData пользователя. Файл настроек должен содержать выбранные пользователем параметры захвата, редактирования и экспорта. Файл данных должен содержать пользовательские списки значений (наименования тем, дисциплин, групп) и параметры интеграции с облачной мультимодальной моделью. Передача снимка экрана в облачную модель осуществляется в кодировке Base64 в составе REST-запроса по протоколу HTTPS.

3.3.2. Требования к программному обеспечению

Приложение должно функционировать в среде операционной системы Microsoft Windows версии 7 SP1 и выше с установленной платформой Microsoft .NET Framework версии 4.0 и выше. Для функции автоматической вставки в документ Microsoft Word требуется установленный пакет Microsoft Office версии 2010 и выше. Для функции запроса описания к облачной мультимодальной модели требуется подключение к сети Интернет. Для функции записи экранного видео требуется наличие в системе аудиоустройства, поддерживающего интерфейс WASAPI.

3.3.3. Требования к аппаратному обеспечению

Минимальные требования к аппаратному обеспечению: процессор с тактовой частотой 2 ГГц и не менее 2 ядер; оперативная память объёмом 2 ГБ; свободное место на накопителе 100 МБ; экран с разрешением не менее 1280×720 пикселей. Рекомендуемые требования: процессор с тактовой частотой 3 ГГц и не менее 4 ядер; оперативная память объёмом 8 ГБ; экран с разрешением 1920×1080 пикселей и частотой обновления 60 Гц; графический адаптер с поддержкой DirectX 11.

4. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ ПРИЛОЖЕНИЯ

Работы по созданию приложения должны быть проведены в четыре основных этапа в рамках спиральной модели жизненного цикла программного обеспечения. На первом этапе должны быть собраны функциональные требования к приложению, проанализированы аналогичные программные продукты и спроектирована модульная архитектура решения с разделением на слои захвата контента, рендеринга графического редактора, обработки текста и интеграции с внешними сервисами. На втором этапе должна быть реализована подсистема захвата изображения с интеллектуальным выделением областей и встроенный графический редактор аннотаций. На третьем этапе должна быть реализована интеграция с облачной мультимодальной языковой моделью Google Gemini и подсистема автоматической вставки в Microsoft Word через интерфейс COM-автоматизации. На четвёртом этапе должны быть подключены вспомогательные подсистемы (запись экранного видео, автоматическая коррекция русскоязычного текста, поиск файлов на локальном носителе,

управление буфером обмена) и проведено комплексное тестирование готового продукта.

5. ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

5.1. Виды, состав, объём и методы испытаний

Контроль готовности приложения осуществляется путём проведения функционального тестирования по перечню тест-кейсов, охватывающих основные сценарии использования: активацию захвата по горячей клавише, выделение области в различных режимах, добавление аннотаций каждого поддерживаемого типа, применение эффектов размытия, генерацию описания средствами облачной модели и автоматическую вставку результата в открытый документ Microsoft Word. Дополнительно проводятся испытания вспомогательных подсистем: записи экранного видео, автоматической коррекции русскоязычного текста и поиска файлов по таблице Master File Table файловой системы NTFS.

5.2. Общие требования к приёмке приложения

Приёмка приложения осуществляется руководителем выпускной квалификационной работы и членами экзаменационной комиссии Московского приборостроительного техникума федерального государственного бюджетного образовательного учреждения высшего образования «Российский экономический университет имени Г.В. Плеханова» по результатам публичной защиты выпускной квалификационной работы. Приложение считается принятым при условии успешного выполнения всех функциональных требований, корректного отображения интерфейса при произвольных параметрах разрешения экрана и коэффициентах масштабирования системы, а также соответствия пакета документации требованиям ГОСТ.

6. ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ВВОДУ ПРИЛОЖЕНИЯ В ЭКСПЛУАТАЦИЮ

Ввод приложения в эксплуатацию осуществляется конечным пользователем самостоятельно путём копирования исполняемого файла в произвольный каталог накопителя и запуска приложения двойным щелчком указательного устройства. Установка дополнительных компонентов не требуется при условии наличия в системе всех зависимостей, перечисленных в подразделе 3.3.2 настоящего документа. При первом запуске пользователь должен указать путь к шаблону документа, API-ключ облачной мультимодальной модели и при необходимости настроить горячие клавиши и цветовую схему приложения. Дополнительная настройка не требуется. Освоение приложения рекомендуется проводить в соответствии с указаниями руководства пользователя, приведённого в Приложении Б.

ПРИЛОЖЕНИЕ Г. ЗАПРОСЫ (ПРОМПТ)

К НЕЙРОСЕТЯМ

НАИМЕНОВАНИЕ НЕЙРОСЕТИ	ЗАПРОС (ПРОМПТ)	РЕЗУЛЬТАТ
Google Gemini	Опиши, что изображено на этом снимке экрана. Дай краткое техническое описание в одно предложение для подписи к рисунку в дипломной работе.	На снимке показан замороженный оверлей приложения с выделенной прямоугольной областью в окне браузера Mozilla Firefox с открытой статьёй в Wikipedia.
ChatGPT	Напиши на C# метод захвата произвольной области экрана с использованием функции BitBlt из библиотеки gdi32.dll.	Сгенерирован метод CaptureScreenRegion(Rectangle rect) с применением CreateCompatibleDC, CreateCompatibleBitmap, BitBlt и копированием в System.Drawing.Bitmap для последующей обработки.
Claude	Реализуй на C# алгоритм оператора Собеля для выделения краёв на полутоновом изображении.	Сгенерирован метод ApplySobelOperator(Bitmap source) с двумя свёрточными матрицами Gx и Gy 3 на 3, нормализацией значений и возвратом изображения с подсвеченными границами.
Google Gemini	Напиши Python-скрипт для построения диаграммы «функциональная схема приложения» с использованием библиотеки matplotlib.	Создан скрипт с использованием matplotlib.patches.Rectangle для блоков и matplotlib.pyplot.plot для линий-соединителей. Применена иерархическая компоновка с корневым блоком и четырьмя колонками подсистем.

YandexGPT	Сформулируй цель и задачи выпускной квалификационной работы по теме «настольное приложение для генерации текстовых описаний снимков экрана».	Сгенерированы одна цель и пять задач: разработка системы захвата, графического редактора аннотаций, интеграции с облачной языковой моделью, подсистемы автоматической вставки в Microsoft Word и комплекта документации.
ChatGPT	Объясни принцип работы алгоритма BFS-заливки по цветовому сходству для интеллектуального выделения области на снимке экрана.	Объяснены этапы: выбор стартового пикселя по позиции курсора, поэтапный обход соседей в 4-связности с проверкой цветового сходства по евклидову расстоянию в пространстве RGB и накопление множества пикселей выделенной области.
Claude	Сгенерируй REST-запрос к API Google Gemini для получения текстового описания изображения в кодировке Base64.	Сгенерирован POST-запрос к <code>generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash:generateContent</code> с JSON-телом, содержащим <code>inline_data</code> с <code>mimeType image/png</code> и системным указанием для краткого технического описания.
Google Gemini	Объясни как использовать интерфейс COM-автоматизации Microsoft Word из приложения на C# для вставки изображения с подписью.	Описано подключение к <code>Word.Application</code> через <code>Marshal.GetActiveObject</code> , использование <code>Selection.InlineShapes.AddPicture</code> , вставка абзаца с центрированной подписью «Рисунок N – ...» и применение стиля <code>Caption</code> .

ChatGPT	<p>Напиши на C# алгоритм построения кубической кривой Безье по четырём контрольным точкам для рисования стрелок аннотаций.</p>	<p>Сгенерирован метод <code>DrawBezier(Graphics g, PointF p0, PointF p1, PointF p2, PointF p3)</code> с использованием <code>Graphics.DrawBezier</code> и дополнительной отрисовкой стрелки-наконечника на основе касательного вектора в конечной точке.</p>
Claude	<p>Опиши принципы работы алгоритма размытия по Гауссу и приведи реализацию на C#.</p>	<p>Описана свёртка изображения с одномерным ядром по двум осям, формула вычисления коэффициентов ядра и реализация метода <code>ApplyGaussianBlur(Bitmap source, double sigma)</code> с разделимой свёрткой по строкам и столбцам.</p>
YandexGPT	<p>Сформулируй краткое описание (аннотацию) к руководству пользователя настольного приложения для генерации описаний снимков экрана.</p>	<p>Сформулирована аннотация на четыре предложения с описанием назначения приложения, его основных функций, целевой аудитории и соответствия требованиям ГОСТ 19.505-79.</p>
Google Gemini	<p>Сгенерируй структуру технического задания в соответствии с ГОСТ 19.201-78 для разработки прикладного программного обеспечения.</p>	<p>Сгенерирована структура из шести разделов: общие сведения, назначение и цели создания, требования к программе, состав и содержание работ, порядок контроля и приёмки, требования к составу и содержанию работ по вводу в эксплуатацию.</p>

ChatGPT	Объясни как реализовать глобальный хук клавиатуры на C# с использованием SetWindowsHookEx из user32.dll для перехвата нажатий клавиши Alt.	Объяснён процесс регистрации низкоуровневого хука WH_KEYBOARD_LL, реализация callback-функции HookCallback, обработка структуры KBDLLHOOKSTRUCT и корректное освобождение хука через UnhookWindowsHookEx при выходе из приложения.
Claude	Напиши на Python скрипт для извлечения списка всех файлов с расширением .docx с локального тома с использованием прямого чтения таблицы Master File Table файловой системы NTFS.	Сгенерирован Python-скрипт с использованием библиотеки pytsk3 для прямого чтения тома NTFS, парсинга записей MFT и фильтрации файлов по расширению .docx без обращения к стандартному API проводника операционной системы.
Google Gemini	Опиши на этом снимке экрана пользовательский интерфейс редактора аннотаций приложения. Дай описание для подписи рисунка в дипломной работе.	На снимке показан главный экран редактора аннотаций с открытым изображением рабочего стола, набором инструментов в левой панели и активным инструментом «Стрелка», применяемым к выделенной области.
ChatGPT	Напиши на C# код для записи системного звука одновременно с захватом экрана через интерфейс WASAPI.	Сгенерирован код с использованием классов IAudioClient и IAudioCaptureClient в режиме loopback, инициализации буфера в формате WAVEFORMATEX и потоковой записи аудио-фреймов в файл WAV параллельно с захватом видео.

Claude	Сгенерируй регулярное выражение для проверки корректности API-ключа Google Gemini, имеющего длину 39 символов и состоящего из латинских букв, цифр, дефисов и символов подчёркивания.	Сгенерировано регулярное выражение <code>^[A-Za-z0-9_-]{39}\$</code> с пояснением каждого элемента: якоря начала и конца строки, символьный класс с допустимыми символами и точная длина в фигурных скобках.
YandexGPT	Подскажи как реализовать фильтр Блума для быстрой проверки наличия словоформы при автокоррекции русского текста.	Объяснены принципы фильтра: массив битов фиксированной длины, k независимых хеш-функций, операция вставки слова через установку k битов, операция проверки через проверку всех k битов. Указаны ограничения по ложным срабатываниям.
Google Gemini	Напиши на C# обработчик нажатия клавиш Ctrl+Z и Ctrl+Y для реализации механизма отмены и повтора операций в графическом редакторе аннотаций.	Сгенерирован обработчик KeyDown с проверкой ModifierKeys.Control, реализация стека операций IUndoAction с методами Execute и Undo, очередь повторов и обновление состояния холста после каждой операции.
ChatGPT	Сгенерируй структуру руководства пользователя в соответствии с ГОСТ 19.505-79 для прикладного программного обеспечения.	Сгенерирована структура из шести разделов: введение, назначение и условия применения, подготовка к работе, описание операций, аварийные ситуации, рекомендации по освоению. Указаны рекомендуемые подразделы для каждого пункта.

Claude	Опиши на этом снимке экрана окно настроек приложения и перечисли видимые элементы управления.	На снимке показано окно настроек с тремя вкладками: General, Effects, Keys. Активна вкладка Effects с настройками радиуса размытия, прозрачности затемнения и параметрами шрифта текстовых блоков.
Google Gemini	Объясни принципы работы метода InlineShape.AddCaption для автоматической нумерации рисунков в Microsoft Word через COM-автоматизацию.	Описан вызов метода с параметрами Label равным «Рисунок» и Position равным wdCaptionPositionBelow, использование Document.Fields.Add со swType равным wdFieldSequence для последовательной нумерации и применение стиля Caption.