

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
| (ДВФУ) |

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
(ШКОЛА)**

**Департамент программной инженерии  
и искусственного интеллекта**

Василец Роман Вадимович

Голомидов Никита Сергеевич

Исихара Никита

Кидяев Антон Антонович

**ДОКУМЕНТАЦИЯ ПО ПРОЕКТУ**

**«Авто-форматирование документов»**

Практика по дисциплине:  
«Основы проектной деятельности»

Б9124-09.03.04 (10)

Преподаватели:  
Логачев Е.М., ассистент департамента

Иваненко Ю.С., старший преподаватель департамента

Васильев О.И., ассистент департамента

г. Владивосток

2025

**Содержание/оглавление**

# **Аннотация**

В рамках аннотации для документации обычно указывается наименование типа работы, её название, перечисляются авторы, руководитель, год сдачи/написания, данные по содержанию + ключевые слова. После ключевых слов идёт описание структуры работы (сколько глав и что в каждой из них раскрывается). Далее приведён пример.

Документация по проекту – Наименование проектаНазвание темы ВКР.

Автор(ы) работы – ФИО.

Научный руководитель / преподаватель – ФИО руководителя/преподавателя, должность, учёная степень, звание.

Год защиты/написания – 2023.

Отчёт по дисциплине «Наименование» – 142 с., 27 рис., 8 табл., 61 источник, 2 прил.

КЛЮЧЕВЫЕ СЛОВА.

Описание структуры работы………

# **Введение**

В современном мире информационные технологии играют всё более значимую роль, обеспечивая удобные способы обучения, работы и хранения данных. Одной из важнейших составляющих цифрового пространства являются текстовые документы, которые активно используются в образовании, науке, бизнесе и управлении.

Несмотря на обилие текстовых редакторов, многие пользователи сталкиваются с проблемой удобного и гибкого редактирования форматированного текста. В большинстве редакторов отсутствуют интеллектуальные механизмы, позволяющие автоматически адаптировать стиль и структуру документа в соответствии с требованиями пользователя или среды. Это приводит к дополнительным временным затратам на ручную корректировку форматирования, что особенно критично для специалистов, работающих с документами в различных форматах и редакторах.

Новизна данной работы заключается в разработке мультиплатформенного авто-редактора, способного автоматически адаптировать форматирование текста при переносе в другие редакторы. Это позволит минимизировать временные затраты пользователей и повысить удобство работы с документами.

Объектом исследования данной работы являются системы редактирования текста, обеспечивающие работу с форматированными документами. Предмет исследования – алгоритмы, отвечающие за корректный перенос форматированного текста между различными платформами.

Практическая значимость проекта заключается в создании инструмента, который упростит взаимодействие с текстовыми редакторами, обеспечит корректный перенос данных и улучшит продуктивность пользователей.

Цель работы – разработка удобного мультиплатформенного авто-редактора документов, позволяющего переносить отформатированный текст без потерь.

Для достижения этой цели необходимо решить следующие задачи:

* Проанализировать существующие решения и выявить их недостатки.
* Разработать алгоритмы, обеспечивающие корректный перенос форматированного текста.
* Реализовать программную модель авто-редактора.
* Провести тестирование системы и оценить её эффективность.
* Подготовить рекомендации по использованию редактора в различных сценариях.

# **План и график работ**

В таблице 1 приведён план и график работ по проекту «Авто-форматирование документов».

Таблица 1 – План и график работ по проекту

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Задача проекта** | **Срок (до)** | **Ответственны(й/е)** |
| 1 | Разработка плана и графика работ над проектом в команде | 11.03.2025 | Василец Роман, Голомидов Никита |
| 2 | Проведение анализа предметной области, обоснование актуальности | 18.03.2025 | Голомидов Никита,  Исихара Никита |
| 3 | Формализация постановки задач программного средства | 25.04.2025 | Исихара Никита |
| 4 | Специфицирование требований к программному средству | 6.05.2025 | Василец Роман, Кидяев Антон |
| 5 | Разработка технического проекта программного средства | 20.05.2025 | Василец Роман, Исихара Никита |
| 6 | Разработка прототипа, презентации программного средства | 17.06.2025 | Василец Роман, Исихара Никита |
| 7 | Проведение оценки соответствия результатов цели (по критериям) | 15.07.2025 | Голомидов Никита, Кидяев Антон |
| 8 | Представление успешно завершенного проекта экспертной комиссии | 15.07.2025 | Василец Роман, Голомидов Никита, Исихара Никита, Кидяев Антон |

# **Команда проекта**

В таблицах 2 и 3 приведена информация по составу проектной команды и ролям в рамках работы над проектом.

Таблица 2–Обязанности участников проекта

|  |  |
| --- | --- |
| **Участник** | **Роли в проекте** |
| Василец Роман Вадимович | Разработчик, анализатор |
| Голомидов Никита Сергеевич | Исследователь ресурсов, координатор, автор документации |
| Исихара Никита | Разработчик, исследователь ресурсов |
| Кидяев Антон Антонович | Исследователь ресурсов, автор документации |

Таблица 3–Распределение командных ролей (по Белбину)

|  |  |
| --- | --- |
| **Роли в проекте (по Белбину)** | **Участник/участники** |
| Координатор (Co-ordinator) | Голомидов Никита Сергеевич |
| Мотиватор (Shaper) | Кидяев Антон Антонович |
| Генератор идей (Plant) | Василец Роман Вадимович, Исихара Никита |
| Аналитик (Monitor evaluator) | Голомидов Никита Сергеевич |
| Реализатор (Implementer) | Василец Роман Вадимович, Исихара Никита |
| Вдохновитель команды (Team worker) | Голомидов Никита Сергеевич, Кидяев Антон Антонович |
| Исследователь ресурсов (Resource investigator) | Голомидов Никита Сергеевич, Кидяев Антон Антонович |
| Завершитель (Completer finisher) | Василец Роман Вадимович |

# **Обзор существующих подходов решения задачи**

В наше время существует большое количество текстовых редакторов, которые позволяют работать с различными видами файлов, но они не имеют возможности подготовить паттерны форматирования для определенной среды, что заставляет затрачивать большое количество времени на ручную корректировку параметров. В этом вопросе может помочь искусственный интеллект (ИИ), способный работать с документами, но не все языковые модели могут выполнять данную задачу, а также использование ИИ не гарантирует наличие подходящей версии файла на выходе.

## **Глоссарий терминов**

ИИ – искусственный интеллект

ПС – программное средство

Паттерн форматирования – заранее заданный шаблон оформления документа, включающих параметры шрифт, отступов, заголовков и других элементов.

Мультиплатформенность – способность программного средства работать на различных операционных системах и устройствах.

## **Обзор существующих ПО и алгоритмов**

Найденные нами программные средства имеют функционал различных сфер, существуют как привычные текстовые редакторы, так и ИИ, способный как вносить, так и рекомендовать пользователю определенные правки.

### **Microsoft Word**

Описание: текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных документов, с локальным применением простейших форм таблично-матричных алгоритмов. [<https://ru.ruwiki.ru/wiki/Microsoft_Word>]

Достоинства: возможность ручного редактирования текста, импорт имеющегося файла для дальнейшей работы с ним, возможность экспорта готового файла в удобном расширении.

Недостатки: отсутствие возможности редактирования текста по сформированному запросу, отсутствие возможности создания паттернов форматирования файла, коммерческая форма распространения.

### **Google Docs**

Описание: бесплатный облачный сервис от компании Google, который позволяет работать с документами, таблицами, презентациями и т.д. Такой функционал делает его прямым конкурентом MS Office. [<https://workspace.ru/blog/rukovodstvo-po-google-docs>]

Достоинства: возможность ручного редактирования текста, импорт имеющегося файла для дальнейшей работы с ним, возможность экспорта готового файла в удобном расширении. Возможность импорта PDF-файлов, бесплатная форма распространения.

Недостатки: отсутствие возможности редактирования текста по сформированному запросу, отсутствие возможности создания паттернов форматирования файла.

### **Chat GPT**

Описание: чат-бот с генеративным искусственным интеллектом, разработанный компанией OpenAI и способный работать в диалоговом режиме, поддерживающий запросы на естественных языках. Система способна отвечать на вопросы, генерировать тексты на разных языках, относящиеся к различным предметным областям. [<https://ru.wikipedia.org/wiki/ChatGPT>]

Достоинства: возможность редактирования текста по сформированному запросу, импорт имеющегося файла для дальнейшей работы с ним, экспорт готового документа.

Недостатки: отсутствие возможности ручного редактирования текста, отсутствие возможности создания паттернов форматирования файла, невозможность бесплатной работы с файлами.

### **DeepSeek**

Описание: нейросеть, разработанная одноименной китайской компанией. Она пишет тексты, анализирует документы, программирует и делает многое другое, не требуя покупать подписку. [<https://habr.com/ru/companies/x-com/articles/878218/>]

Достоинства: возможность редактирования текста по сформированному запросу, импорт имеющегося файла для дальнейшей работы с ним, бесплатная форма распространения.

Недостатки: отсутствие возможности ручного редактирования текста, отсутствие возможности создания паттернов форматирования файла, невозможность экспорта готового файла.

### **Yandex GPT**

Описание: нейросеть семейства GPT от компании «Яндекс», которая может создавать и перерабатывать тексты, предлагать новые идеи и учитывать контекст беседы с пользователем. [<https://ru.wikipedia.org/wiki/YandexGPT>]

Достоинства: возможность редактирования текста по сформированному запросу, бесплатная форма распространения.

Недостатки: отсутствие возможности ручного редактирования текста, отсутствие возможности создания паттернов форматирования файла, не принимает материал на импорт, невозможность экспорта готового файла.

## **Сводная/сравнительная таблица**

Критерии для сравнения были выбраны следующие:

1. Возможность ручного редактирования – чтобы в случае несоответствия требованиям, пользователь мог самолично сделать то, что не могут сделать алгоритмы.
2. Импорт файла для редактирования – для удобства пользователя программное средство должно принимать на вход уже готовый файл.
3. Редактирование по сформированному запросу – чтобы исключить лишнюю трату времени.
4. Создание паттернов форматирования файлов – дает возможность вручную настроить оформление один раз, чтобы не полагаться на неисправные алгоритмы.
5. Экспорт готового файла – чтобы не переносить правки в другой файл после завершения работы.

Таблица 4– Сравнение программных средств

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Критерий\ПС** | **Microsoft Word** | **Google Docs** | **Chat GPT** | **DeepSeek** | **Yandex GPT** |
| Ручное редактирование | + | + | – | – | – |
| Импорт файла для редактирования | + | + | + | + | – |
| Редактирование по сформированному запросу | – | – | + | + | + |
| Создание паттернов форматирования файлов | – | – | – | – | – |
| Экспорт готового файла | + | + | + | – | – |

## **Выводы по анализу**

Исходя из сравнительной таблицы, все программные средства объединяет отсутствие возможности создания паттернов для форматирования файла, из-за чего уходит большое количество времени на оформление документов при работах в организациях с определенными требованиями к внешнему виду работ, так как редактирование по запросу реализовано только в ИИ-системах, но без возможности ручного вмешательства, а в текстовых редакторах нет возможности создания полноценных паттернов.

# **Анализ предметной области и построение модели**

В настоящей главе выделяются основные понятия и информационные объекты предметной области, рассматриваются примеры ситуаций предметной области, строится математическая и онтологическая модели предметной области.

## **Целевая аудитория**

Целевой аудиторией проекта являются:

* Студенты ДВФУ (люди от 17 до 30 лет)

Форматирование отчетов, практических заданий, лабораторных, курсовых и дипломных работ.

* Преподаватели ДВФУ (люди от 20 до 60 лет)

Форматирование отчетов, документов, оформление заявлений, приказов.

* Сотрудники ДВФУ (люди от 18 до 60 лет)

Оформление документов.

## **Задачи целевой аудитории**

Главной задачей целевой аудитории данного программного средства является корректное автоматизированное оформление созданных или отредактированных документов в соответствии с шаблонами Дальневосточного Федерального Университета или на основе параметров, введенных пользователем, без необходимости выполнения ручного форматирования. Также подразумевается возможность создания паттернов от пользователя.

## **Множество задач предметной области**

Предметная область проекта охватывает процессы автоматизации оформления текстовых документов в соответствии с заданными форматами. Несмотря на то, что ПС не является универсальным редактором, оно решает ряд специализированных, но значимых задач в образовательной и административной среде.

Реализация включает в себя автоматическое приведение документа к заданному стилю оформления; распознавание заголовков, подзаголовков, списков, таблиц и других элементов структуры при помощи специальных символов, и приведение их к единообразному виду; прием файлов различных форматов на импорт и экспорт в заданном формате (DOCX, PDF и др.) без потери структуры и форматирования; сохранение документа в чате с Telegram-ботом для доступа к файлу на любой платформе; обеспечение пользователю возможности получения корректного по оформлению документа без необходимости ручного редактирования.

## **Информационные объекты**

Требования к форматированию документов для IT-проекта

4.4.1. Общие положения

Документ считается соответствующим стандарту, только если выполнены все требования, приведённые ниже. Нарушение любого из ограничений делает документ невалидным.

4.4.2. Форматы документов

Разрешённые форматы файлов:

- DOCX

- PDF

- TXT

Файлы других форматов не допускаются.

4.4.3. Структура и параметры форматирования DOCX, PDF и прочих форматов

4.4.3.1. Поля страницы

- Левое поле: от 15 до 25 мм

- Правое поле: от 10 до 20 мм

- Верхнее поле: от 15 до 25 мм

- Нижнее поле: от 15 до 25 мм

- Использование полей вне указанных границ запрещено.

4.4.3.2. Отступы

- Отступ первой строки абзаца: только 1,25 см

- Межабзацный отступ (интервал перед/после абзаца): только 0 pt или 6 pt

- Другие значения запрещены.

4.4.3.3. Межстрочный интервал

- Разрешены только значения: одинарный (1,0), полуторный (1,5), двойной (2,0)

- Использование дробных или иных значений запрещено.

4.4.3.4. Стиль и параметры текста

- Шрифты: только Arial, Times New Roman, Courier New

- Размер шрифта: от 10 до 16 pt

- Цвет шрифта: только чёрный (RGB 0,0,0)

- Начертание: обычное, полужирное, курсив

- Подчёркивание запрещено

- Разрешены только прямое и наклонное начертание

- Разрешены только латиница и кириллица

4.4.3.5. Структурные элементы

- Заголовки: не более трёх уровней, каждый выделяется размером и/или полужирным

- Списки: только маркированные (•) или нумерованные (1., 2., 3.)

- Абзацы: каждый начинается с отступа, не допускается пустых строк между абзацами

- Недопустимо размещение изображений или таблиц внутри заголовков

- Каждый структурный элемент должен соответствовать своему типу (заголовок, абзац, таблица, изображение, подпись)

4.4.4. Таблицы и изображения

4.4.4.1. Таблицы

- Только прямоугольные, без вложенных таблиц

- Количество строк: от 1 до 40

- Количество столбцов: от 1 до 12

- Ширина таблицы: не более ширины рабочей области страницы

- Подпись: обязательна, размещается над таблицей, шрифт Times New Roman, 12 pt, курсив

- Выравнивание таблицы: слева, по центру, справа (указывается явно для каждой таблицы)

- Формат границ: только одинарная линия, чёрного цвета

- Цвет заливки ячеек: только белый

- Запрет на объединение ячеек по диагонали

- Описание таблицы (структура, размеры, подпись, выравнивание) включается в отдельное поле шаблона

4.4.4.2. Изображения

- Форматы: только JPEG, PNG

- Размер изображения: не менее 32×32 px, не более 2048×2048 px

- Размещение: только в теле документа, не в заголовках

- Положение: слева, по центру, справа (должно быть указано для каждого изображения)

- Подпись: обязательна, размещается под изображением, шрифт Times New Roman, 12 pt, курсив

- Выравнивание подписи: по центру

- Цветовое пространство: только RGB

- Использование прозрачных изображений не допускается

- Описание изображения (формат, размер, подпись, положение, выравнивание) включается в отдельное поле шаблона

4.4.5. Нумерация страниц

- Расположение номера: только снизу по центру или по правому краю

- Шрифт: Times New Roman, 12 pt

- Стиль: обычный

- Формат номера: арабские цифры

- Стиль нумерации должен быть единым для всего документа

4.4.6. Валидация

- Каждый параметр документа проверяется на соответствие допустимым значениям

- При обнаружении несоответствия указывается конкретный параметр и его значение

- Документ, не соответствующий хотя бы одному из требований, возвращается на

4.4.7. Пример формализованного шаблона

Формат: DOCX

Поля: левое 20 мм, правое 15 мм, верхнее 20 мм, нижнее 20 мм

Отступ первой строки: 1,25 см

Межстрочный интервал: 1,5

Шрифт основного текста: Arial, 12 pt, чёрный, обычный

Заголовки: Times New Roman, 14 pt, полужирный

Таблица: 5 строк × 3 столбца, подпись: "Таблица 1 — Результаты тестирования", выравнивание по центру

Изображение: PNG, 640×480 px, подпись: "Рисунок 1 — Архитектура системы", размещение по центру

Нумерация страниц: снизу по центру, Times New Roman, 12 pt

## **Задачи обработки, хранения и передачи информации**

Обработка информации:

* Конвертация файлов допускается только между разрешёнными форматами: DOCX, PDF и прочее.
* Перед конвертацией выполняется анализ исходного форматирования документа на соответствие установленному шаблону.
* Любые попытки конвертации в неразрешённые форматы или применения неописанных параметров блокируются системой.

Хранение данных:

* Все шаблоны форматирования и пользовательские настройки могут храниться локально или в облачной базе данных.
* Доступ к хранилищу осуществляется только авторизованными пользователями по защищённому протоколу (например, HTTPS или VPN).
* Вся информация в базе данных проходит обязательную валидацию перед записью.

Передача данных:

* Передача данных между компонентами системы (клиент, сервер, база данных) выполняется по защищённым каналам связи (например, HTTPS, TLS).
* Экспорт документов осуществляется только в разрешённые форматы (DOCX, PDF, TXT) с проверкой валидности результата.
* При экспорте и передаче файлов ведётся журналирование операций (логирование даты, времени, пользователя и типа операции).
* Переданные файлы могут быть переданы пользователю через чат-бота только после успешного прохождения всех этапов проверки.
* При необходимости интеграции с внешними сервисами или сетями база данных, обмен осуществляется исключительно через API с авторизацией и контролем доступа.

Описание обмена данными с базой данных и по сети:

* Все операции чтения и записи данных в базу осуществляются через стандартизированные запросы (например, REST API или SQL-запросы), с обязательной проверкой корректности данных на стороне сервера.
* Межсетевой обмен информацией допускается только по защищённым протоколам, с шифрованием передаваемых данных и обязательной аутентификацией.
* Любые попытки несанкционированного доступа или передачи данных фиксируются и блокируются системой безопасности.

## **Анализ смысла ситуаций предметной области**

4.6 Анализ смысла ситуаций предметной области

Ситуация 1: Форматирование отчёта по стандартам ДВФУ

Описание:

Пользователь загружает документ (допустимые форматы: DOCX, PDF) в систему. Редактор автоматически применяет к содержимому документа предустановленный или пользовательский шаблон без участия пользователя в процессе разметки.

Решение:

Документ автоматически преобразуется к виду, соответствующему выбранному оформлению (например, ДВФУ). Все параметры форматирования (шрифты, размеры, поля, отступы, интервалы, структура, оформление таблиц и изображений, нумерация страниц и др.) приводятся в строгое соответствие с шаблоном, исключая необходимость ручных правок пользователем.

Система поддерживает автоматическую конвертацию между форматами DOCX, PDF и прочее. Конвертация возможна только между этими форматами.

Ограничения:

- Шаблон (ДВФУ или любой пользовательский) должен быть предварительно загружен в систему администратором либо самим пользователем.

- Пользователь не должен использовать в исходном документе нестандартные или специализированные символы, скрытые теги, управляющие коды, нестандартные маркеры или собственные макросы для указания составляющих шаблона.

- Допустимы только следующие форматы для загрузки и конвертации: DOCX, PDF, прочее. Использование других форматов (например, ODT, RTF, HTML, JPG и других) запрещено.

- Максимальный размер документа — 20 МБ.

- Количество страниц в документе — не более 300.

- Количество таблиц в одном документе — не более 50, каждая таблица — максимум 40 строк и 12 столбцов.

- Количество изображений в одном документе — не более 50, каждый файл изображения — не менее 32×32 пикселей и не более 2048×2048 пикселей.

- При конвертации между форматами запрещено терять структурную разметку документа (заголовки, списки, таблицы, изображения, подписи, нумерацию страниц и т.п.).

- В формате PDF не допускается наличие редактируемых полей.

- Все параметры форматирования (поля, интервалы, шрифты, размеры, стили, структура, подписи, выравнивания, форматы изображений и таблиц, стиль нумерации и т.д.) должны строго соответствовать выбранному шаблону. Отклонения (например, поля более 25 мм, шрифты вне согласованного перечня, межстрочные интервалы, не указанные в шаблоне и т.д.) запрещены.

- Каждый документ после форматирования автоматически проходит валидацию на соответствие шаблону. При обнаружении несоответствий система возвращает ошибку с указанием конкретного нарушенного параметра.

- Если шаблон отсутствует, или не загружен, или некорректен — операция невозможна, пользователю выводится соответствующее уведомление.

- Конвертация между форматами осуществляется только при условии, что итоговый документ соответствует всем ограничениям шаблона (при невозможности полного соответствия выдаётся ошибка).

## **Анализ знаний предметной области**

Для каждого документа существует непустое множество параметров форматирования.

• Связи между параметрами (например, заголовок 1 уровня всегда имеет определенный шрифт и размер).

• Документ не может быть отформатирован без указания хотя бы одного параметра.

• Файл не может быть отредактирован без внесения специальных символов внутри текста.

• Каждый параметр форматирования влияет на внешний вид или структуру документа.

• Некоторые параметры могут быть взаимозависимы (например, изменение стиля может автоматически изменить межстрочный интервал).

• Для корректного отображения форматирования требуется поддержка соответствующего программного обеспечения.

• Пользователь может изменять параметры форматирования как для всего документа, так и для отдельных его частей.

• При сохранении документа параметры форматирования сохраняются вместе с содержимым.

• Нарушение связей между параметрами может привести к некорректному отображению документа.

• В системе предусмотрены ограничения на допустимые значения параметров форматирования (например, минимальный и максимальный размер шрифта).

• Для разных типов документов может быть предусмотрен разный набор обязательных параметров форматирования.

• При экспорте или импорте документа параметры форматирования должны быть корректно преобразованы или сохранены.

• Анализ и обработка параметров форматирования необходимы для автоматизации процессов редактирования и валидации документов.

## **Формальная постановка задач**

Дано:

Строка сообщения или текстовый файл, содержащий произвольный текст (inputMessage ∈ String или inputMessage ∈ File).

Тип выходного файла (type ∈ TypeFile), определяющий формат создаваемого документа.

Требуется:

На основании входных данных (inputMessage, type) сформировать новый файл (outputMessage ∈ File), в котором содержимое исходного сообщения или текстового файла автоматически структурировано и отформатировано согласно заранее установленным правилам для выбранного типа документа.

Формализованное описание задачи:

Пусть

I — множество допустимых входных данных (строк или файлов),

T — множество допустимых типов файлов,

F — множество выходных файлов, удовлетворяющих требованиям форматирования.

Требуется построить отображение

A: I × T → F,

такое, что для любых входных данных i ∈ I и типа t ∈ T, результатом работы алгоритма является файл f ∈ F, обладающий следующими свойствами:

Содержит структурированное и форматированное представление данных из i согласно правилам для типа t.

Включает титульный лист, внутренние структурированные разделы, нумерацию страниц и иные элементы, предписанные форматом t.

Форматирование и структура документа соответствуют заданному стандарту для выбранного типа.

Ограничения и условия:

Все этапы преобразования выполняются автоматически согласно формальному алгоритму.

Входные данные могут быть либо текстом, либо файлом.

Результирующий файл должен быть валиден и готов к использованию без дополнительной ручной доработки.

Требуется реализовать алгоритм A, обеспечивающий выполнение вышеуказанных свойств для любых допустимых входных данных и типов выходного файла.

## **Технические противоречия (2 ШТ)**

Описание технического противоречия сначала словами, а затем при помощи схемы.

Свойство «…» положительно влияет на характеристику «…» и негативно влияет на характеристику «…». Противоположное свойство «…» положительно влияет на характеристику «…» и негативно влияет на характеристику «…».



Рисунок 1 – Первое техническое противоречие

## **ИКР**

Идеальным конечным результатом проекта является аудитория всех студентов, преподавателей и сотрудников ДВФУ, которая благодаря программному средству экономит большое количество времени на оформлении.

## **Описание решения ТРИЗ-задачи**

Согласно ТРИЗ (теории решения изобретательских задач), предложены следующие способы преодоления технических противоречий разрабатываемого программного средства.

Описание того, за счёт чего будут преодолены все указанные в разделе 4.9 противоречия (для приближения к идеальному конечному результату). Здесь могут быть компромиссы, упрощения и т.п.

## **Критерии завершенности проекта**

Проект «Авто-форматирование документов» можно считать успешно завершенным при достижении следующих значений критериев качества:

* Документы форматируются корректно без необходимости внесения дополнительных правок;
* Время оформления документов сводится к паре секунд или минут;
* Программное средство получается не менее 100 запросов в первый месяц работы;
* Реализована поддержка различных форматов файлов (PDF, DOCX и т.д.);
* Выявляется минимальное количество ошибок (не более 2-х на документ) или не выявляется вовсе;

## **SWOT-анализ проекта**

Проект имеет преимущества в скорости и качестве форматирования, но полностью автоматизировать данный процесс нельзя, поэтому пользователю необходимо использовать технические символы для обозначения компонентов оформления. Также есть риск, что руководство пользователя будет неправильно прочитано, из-за чего ПС не выдаст необходимый результат. На данном этапе имеется только один паттерн для форматирования, но в дальнейшем набор шаблонов может быть расширен.

Таблица 5–SWOT-анализ проекта

|  |  |
| --- | --- |
| **Сильные стороны (strengths)** | **Слабые стороны (weaknesses)** |
| Скорость форматирования файла; отсутствие необходимости в ручном редактировании; | Необходимость введения специальных символов в документ для работы ПС; |
| **Возможности (opportunities)** | **Риски (threats)** |
| Последующее добавление других паттернов форматирования; | Неправильное считывание символов, что приводит к неправильному оформлению; |

# **Проектирование программной системы**

Общий текст по главе

## **СДР**

1. Анализ требований и планирование:

1. Определение целевой аудитории и их потребностей.
2. Анализ существующих решений и выявление недостатков.
3. Формализация постановки задач программного средства.
4. Специфицирование требований к программному средству (пользовательские, функциональные, нефункциональные).
5. Разработка плана проекта и распределение задач между членами команды.

2. Проектирование системы:

1. Разработка технического проекта (архитектура системы, выбор технологий).
2. Проектирование базы данных.

3. Разработка прототипа.

4. Разработка основной версии:

1. Реализация всей функциональности программного средства в соответствии с требованиями.
2. Реализация обработки ошибок и исключений.

5. Тестирование и отладка:

1. Разработка плана тестирования (определение тестовых сценариев, создание тестовых данных).
2. Проведение тестирования программного средства (функциональное тестирование, тестирование производительности, тестирование безопасности).
3. Отладка программного средства (исправление ошибок и недочетов).

6. Документирование:

1. Подготовка технической документации (описание архитектуры системы, алгоритмов, API).
2. Подготовка пользовательской документации (руководство пользователя, справочная система).

7. Внедрение и сопровождение:

1. Развертывание программного средства на целевых платформах.
2. Обучение пользователей.
3. Сопровождение программного средства (исправление ошибок, добавление новых функций).

8. Оценка результатов:

1. Проведение оценки соответствия результатов цели проекта (по критериям).
2. Подготовка отчета о результатах проекта.
3. Представление завершенного проекта.

## **Диаграмма Гантта**

Рисунок 1 – Диаграмма Гантта

## **Программное средство коллективной разработки**

GitHub – это веб-платформа, созданная для хостинга, отслеживания и совместной работы над проектами разработки программного обеспечения. В основе лежит система контроля версий Git, позволяющая разработчикам работать одновременно, отслеживая изменения и обеспечивая возможность возврата к предыдущим версиям GitHub расширяет Git, предоставляя веб-интерфейс и инструменты для совместной работы.

Главный экран (дашборд) предоставляет обзор активности и проектов. Он включает: ленту активности (коммиты, pull requests, issues), список репозиториев, рекомендации репозиториев, новости GitHub и уведомления.

Страница репозитория – центральное место управления проектом. Она включает:

* Код - структура файлов и директорий, история изменений.
* Issues (Задачи) - система отслеживания ошибок и задач.
* Pull Requests - механизм для предложения изменений кода и их рецензирования.
* Actions - автоматизация задач разработки (тестирование, сборка).
* Projects - инструмент для организации задач.
* Wiki - документация по проекту.
* Insights - статистика о проекте.

Страница Issues позволяет отслеживать и управлять задачами. Она включает: список Issues (открытые и закрытые, с фильтрацией), форму для создания нового Issue (заголовок, описание, назначение, метки), страницу отдельного Issue (подробная информация о задаче).

## **Спецификация требований**

Основные компоненты DFD:

Внешние сущности:

Пользователь – загружает документ, выбирает шаблон, получает результат.

Telegram-бот (опционально) – альтернативный интерфейс для взаимодействия.

Процессы:

Загрузка документа – прием файла от пользователя.

Валидация документа – проверка формата, структуры и параметров.

Применение шаблона – автоматическое форматирование согласно выбранному шаблону.

Экспорт документа – сохранение в нужном формате (DOCX, PDF).

Хранилища данных:

База данных шаблонов – содержит предустановленные и пользовательские шаблоны.

Журнал операций – логирует действия пользователей и ошибки.

Потоки данных:

Файл документа (DOCX, PDF, TXT) → Загрузка → Валидация.

Запрос шаблона → База данных шаблонов → Применение шаблона.

Отформатированный документ → Экспорт → Пользователь.

Ошибка валидации → Уведомление пользователю.

Детализация процессов:

Валидация документа:

Проверяет:

Формат файла (DOCX, PDF, TXT).

Поля, шрифты, отступы, нумерацию.

Наличие запрещенных элементов (например, изображения в заголовках).

При ошибке возвращает список нарушений.

Применение шаблона:

Автоматически корректирует:

Поля страницы (левое: 20 мм, правое: 15 мм и т.д.).

Шрифты (Arial, Times New Roman).

Структуру (заголовки, списки, таблицы).

Экспорт:

Конвертирует документ в выбранный формат с сохранением форматирования.

Проверяет соответствие выходного файла требованиям.

## **Диаграмма вариантов использования**

Участники и их роли:

Пользователь – инициирует процесс.

Интерфейс системы – принимает файл и управляет процессом.

Валидатор – проверяет документ.

База данных шаблонов – предоставляет шаблоны.

Форматтер – применяет форматирование.

Модуль экспорта – сохраняет результат.

Сценарий с детализацией:

Шаг 1: Загрузка файла

Пользователь загружает файл через интерфейс.

Интерфейс передает файл валидатору.

Шаг 2: Валидация

Валидатор проверяет:

Соответствие формата (DOCX, PDF, TXT).

Наличие обязательных структурных элементов (например, заголовков).

Если ошибка:

Валидатор возвращает список нарушений интерфейсу.

Интерфейс уведомляет пользователя.

Если успех:

Валидатор запрашивает шаблон из БД.

Шаг 3: Получение шаблона

БД возвращает шаблон (например, "Шаблон ДВФУ").

Шаг 4: Форматирование

Валидатор передает файл и шаблон форматтеру.

Форматтер:

Изменяет шрифты, поля, отступы.

Нумерует страницы.

Корректирует таблицы и изображения.

Шаг 5: Экспорт

Форматтер передает результат модулю экспорта.

Модуль экспорта сохраняет файл в выбранном формате.

Шаг 6: Завершение

Интерфейс уведомляет пользователя о готовности файла.

## **Архитектурно-контекстная диаграмма**

Состояния:

Загружен – документ принят системой.

На проверке – выполняется валидация.

Ошибка – выявлены несоответствия.

Форматирование – применение шаблона.

Экспорт – подготовка к сохранению.

Завершен – документ готов.

Переходы и триггеры.

Загружен → На проверке

Триггер: Автоматически после загрузки.

Действие: Запуск валидации.

На проверке → Ошибка

Триггер: Обнаружены нарушения (например, неверные поля).

Действие: Формирование отчета об ошибках.

На проверке → Форматирование

Триггер: Документ валиден.

Действие: Загрузка шаблона из БД.

Форматирование → Экспорт

Триггер: Успешное применение шаблона.

Действие: Конвертация в выбранный формат.

Экспорт → Завершен

Триггер: Файл сохранен.

Действие: Уведомление пользователя.

Ошибка → Загружен

Триггер: Пользователь повторно загружает исправленный файл.

Действие: Повторная валидация.

Обработка исключений:

Если шаблон не найден в БД:

Переход в состояние Ошибка с сообщением: "Шаблон недоступен".

Если форматирование не удалось:

Откат в состояние “На проверке” для повторной обработки.

## **Диаграмма потоков данных**

Диаграмма, которая отражает пути, по которым циркулируют данные внутри системы, а также между системой и внешним миром. Часть описание заимствуется из пункта 4.8, а именно входные и выходные данные для разных функций (в сокращённой форме). Кроме самих данных на диаграмме отражаются запросы внутри системы и сами функции модулей.

## **Диаграмма последовательностей состояний**

Диаграммы последовательностей моделируют взаимодействия между объектами в едином сценарии использования. Они иллюстрируют, как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая использования. (Пример проиллюстрирован ниже)

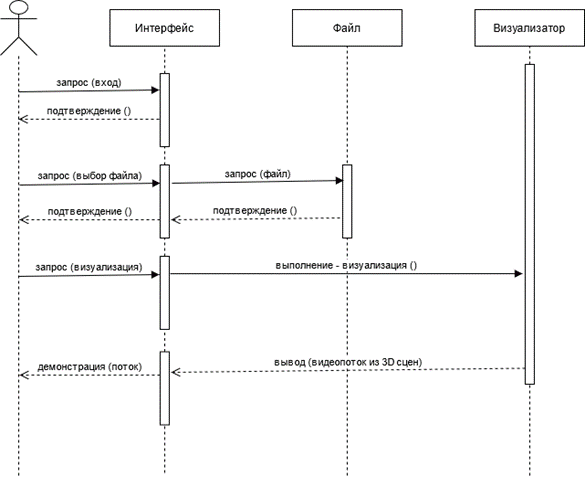


Рисунок n – Диаграмма последовательностей состояний

## **Диаграмма перехода из состояния в состояние**

При проектировании сложной системы принято делить ее на части, каждую из которых затем рассматривать отдельно. Таким образом, при объектной декомпозиции система разбивается на объекты или компоненты, которые взаимодействуют друг с другом, обмениваясь сообщениями и данными.

Сообщения описывают или представляют собой некоторые события. Получение объектом сообщения активизирует его и побуждает выполнять предписанные действия. Даётся также краткое описание.

Диаграмма состояний и переходов показывает:

* пространство состояний данного класса;
* события, которые влекут переход из одного состояния в другое;
* действия, которые происходят при изменении состояния.

(Пример ниже)

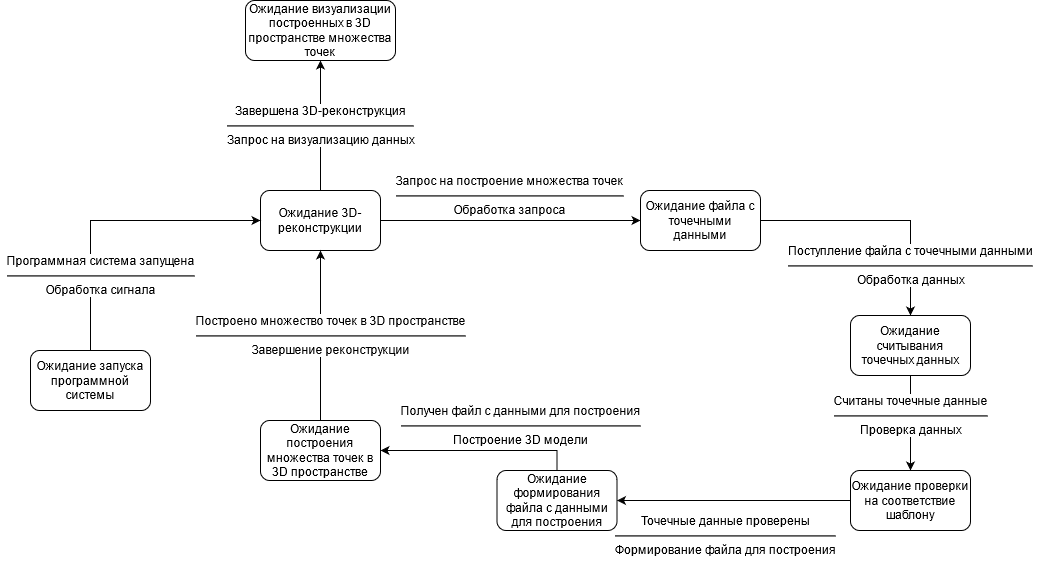


Рисунок n – Диаграмма перехода из состояния в состояние

## **Разработка интерфейса**

Формирование структуры интерфейса с описанием окон и их наполнения.

## **Диалог с пользователем**

Описание действий пользователя по взаимодействию с предполагаемым интерфейсом проекта.

## **Прототип программного средства**

Скриншоты с кратким описанием формочного прототипа / прототипа интерфейсов (возможен перенос работ на 3ий семестр)

## **План тестирования**

Представляется таблица с классами эквивалентности, содержащая описания тестовых ситуаций с их идентификаторами. Для данных ситуаций формируется набор тестов покрывающий все классы эквивалентности. Для разного рода событий формируется набор сообщений (чаще всего для ошибок)

# **Реализация, тестирование и испытание……**

В данной главе рассматривается процесс реализации, тестирования и испытания разных компонентов программного средства по … .

## **Вспомогательное ПО**

Предоставляется информация о выбранном ПО, которое будет использоваться при реализации Вашего проекта, информация о языке программирования.

## **Особенности реализации ИЛИ Испытания реализованных методов/функций**

В зависимости от типа работ изменяется название и содержание данного раздела (ПО СОГЛАСОВАНИЮ С ПРЕПОДАВАТЕЛЕМ).

В случае если сложность проекта высока и/или у команды нет достаточного уровня знаний и навыков для реализации какой-либо части проекта, то выбирается «Особенности реализации». Здесь Вам необходимо проанализировать и описать все аспекты, которые будут необходимы при реализации Вашего проекта (библиотеки, сервисы, построение связей (в том числе как их реализовать) и т.д.).

Для тех, кто планирует реализовать какой-либо функционал своего проекта данный раздел будет носить название «Испытания реализованных…». Здесь необходимо будет привести примеры работы реализованного Вами функционала (с разными данными, возможно показать разные итерации функционала) и, соответственно, сделать выводы относительно работы реализованного кода. Описывать более подробно структуру кода, библиотек или связей не требуется, только если это не несёт важную для Вас составляющую при формулировании вывода или при описании испытаний/экспериментов.

## **Тестирование**

В зависимости от работ в разделе 6.2 описывается тестирование по чёрному или белому ящику. Примеры были представлены в рамках 2го семестра.

## **Ограничения для первичного программного тестирования**

В данном разделе описываются ограничения, которые могут вытекать из разделов 4.4 − 4.8. Чаще всего данные ограничения сводятся к каким-либо идеальным условиям.

ПРИМЕЧАНИЕ: РАЗДЕЛЫ 6.2 − 6.4 МОГУТ МЕНЯТЬСЯ МЕСТАМИ НА ОСНОВЕ ПОВЕСТВОВАНИЯ.

## **Презентация проекта**

На рисунке N − рисунок N представлена презентация проекта. (презентация оформлена в официальном стиле на основе фирменного стиля ДВФУ о котором речь шла ранее (представлен на сайте ДВФУ).

# **Заключение**

Заключение содержит основные выводы по работе. Делается акцент на выполнении целей и задач работы, а также описываются основные аспекты, которые Вы можете представить как выводы по всей проделанной работе.

# **Список источников**