

Федеральное государственное автономное образовательное учреждение высшего  
образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра «Инфокогнитивных технологий»

Направление подготовки/ специальность: Системная и программная инженерия

## ОТЧЕТ

по проектной практике

Студент: Махров Александр Леонидович Группа: 241-3211

Место прохождения практики: Московский Политех, кафедра Инфокогнитивных технологий

Отчет принят с оценкой \_\_\_\_\_ Дата \_\_\_\_\_

Руководитель практики: \_\_\_\_\_

Москва 2025

## ОГЛАВЛЕНИЕ

1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ.....	3
1.1. РАЗРАБОТКА И ПУБЛИКАЦИЯ СТАТИЧЕСКОГО ВЕБ-САЙТА.....	3
ТЕМА САЙТА: "ODC — ОТКРЫТЫЙ ПРОТОКОЛ ЦИФРОВЫХ НАЛИЧНЫХ" (OPEN DIGITAL CASH).....	3
1.2. РАЗРАБОТКА МИНИ-БИБЛИОТЕКИ ДЛЯ РАБОТЫ С XLSX НА ЯЗЫКЕ SWIFT .....	5
1.3. ОБЩАЯ СУММА ЗАТРАЧЕННОГО ВРЕМЕНИ.....	6
2 ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ (ЗАКАЗЧИКА ПРОЕКТА) ...	7
3.1. БАЗОВАЯ ЧАСТЬ: РАЗРАБОТКА И ПУБЛИКАЦИЯ СТАТИЧЕСКОГО САЙТА.....	8
3.1.1. ПОДГОТОВКА К РАБОТЕ И НАСТРОЙКА СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ .....	8
3.1.2. ОСВОЕНИЕ СИНТАКСИСА MARKDOWN ДЛЯ ПРОЕКТНОЙ ДОКУМЕНТАЦИИ.....	8
3.1.3. ИЗУЧЕНИЕ И НАСТРОЙКА ГЕНЕРАТОРА СТАТИЧЕСКИХ САЙТОВ HUGO .....	9
3.1.4. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ И НАПОЛНЕНИЕ САЙТА КОНТЕНТОМ .....	9
3.1.5. ФИНАЛЬНАЯ ОТЛАДКА, ТЕСТИРОВАНИЕ И ПУБЛИКАЦИЯ САЙТА .....	10
3.2. ВАРИАТИВНАЯ ЧАСТЬ: РЕАЛИЗАЦИЯ МИНИ-БИБЛИОТЕКИ ДЛЯ РАБОТЫ С XLSX НА ЯЗЫКЕ SWIFT .....	12
4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ ПО ПРОЕКТНОЙ ПРАКТИКЕ .....	15
ЗАКЛЮЧЕНИЕ .....	17
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	18

## **1. Общая информация о проекте**

Проектная практика включала выполнение двух основных самостоятельных частей:

1. создание и публикация статического тематического веб-сайта,
2. разработка мини-библиотеки для работы с файлами XLSX (Excel) на языке Swift.

Работа велась индивидуально, с подробным учётом этапов и трудозатрат.

### **1.1. Разработка и публикация статического веб-сайта**

Тема сайта:

"ODC — открытый протокол цифровых наличных" (Open Digital Cash).

Описание этапов работы:

- Работа с Git и репозиторием (5 часов):  
На этом этапе был создан личный репозиторий на GitHub, настроены связи с локальным репозиторием, освоены и применялись базовые команды (init, add, commit, push, pull). Работа велась в одной ветке. Регулярно фиксировались все значимые изменения структуры и контента сайта. Навыки работы с системой контроля версий активно закреплялись на практике.
- Оформление документов и контента в Markdown (5 часов):  
Вся документация, тексты страниц и служебная информация сайта оформлялись в формате Markdown. Использовались базовые элементы разметки: заголовки, списки, ссылки, блоки кода, вставки изображений и таблиц. Markdown применялся как для описания структуры сайта, так и для технических разделов, ресурса и журнала.
- Изучение Hugo и настройка сайта (10 часов):  
В рамках этого этапа изучалась документация по Hugo, осваивались команды для генерации, настройки и локального запуска сайта. Подбиралась и адаптировалась тема PaperMod, настраивались конфиги для структуры и разделов. Также отрабатывались вопросы локальной отладки и развертывания проекта.

- Дизайн, структурирование и наполнение сайта (5 часов):

Выполнялось наполнение сайта уникальным контентом по теме цифровых наличных:

- аннотация на главной странице,
- раздел "О проекте" с историей и особенностями ODC,
- страницы с описанием этапов разработки,
- журнал публикаций о ходе работ,
- подборка ресурсов и технологий,
- информация об участнике.

Использовались схемы, иллюстрации и диаграммы (в том числе на языке Mermaid).

- Дополнительная отладка и публикация (3 часа):

Проводилась итоговая проверка структуры, устранение багов в вёрстке, тестирование корректности отображения всех разделов. На этом же этапе была настроена публикация сайта через GitHub Pages, проверялась доступность ресурса для пользователей.

- Взаимодействие с организацией-партнёром (4 часа):

В качестве ознакомительного опыта участвовал в онлайн-конференции, посвящённой современным вопросам информационной безопасности (например, мероприятие R-Vision). Изучались практические кейсы и тенденции отрасли, что позволило расширить представление о современных ИТ-задачах, связанных с темой сайта.

- Оформление и сдача итогового отчёта (4 часа):

На завершающем этапе был подготовлен индивидуальный отчёт, отражающий все ключевые этапы работы, временные затраты и анализ приобретённых навыков. Отчёт оформлен в требуемых форматах (docx и pdf), размещён в репозитории и загружен для проверки в СДО.

Итоговая трудоёмкость по сайту: 36 часов.

## 1.2. Разработка мини-библиотеки для работы с XLSX на языке Swift

Цель:

Создать собственную минималистичную библиотеку для чтения и записи файлов формата XLSX (Microsoft Excel) на языке Swift без применения сторонних решений, с освоением принципов работы стандарта OpenXML.

Описание этапов работы:

- Изучение предметной области, анализ формата XLSX и спецификаций OpenXML (7 часов):  
Анализовалась структура XLSX-файла как ZIP-архива, рассматривались ключевые XML-файлы (workbook.xml, sheet, sharedStrings, styles). Изучались официальные спецификации и примеры из открытых исходных кодов, ставились задачи по объёму реализуемого функционала.
- Проектирование основных структур данных (5 часов):  
Разрабатывалась архитектура библиотеки: определялись сущности (Workbook, Sheet, Cell, Style), способы хранения и обработки данных, типы ячеек и их представление в коде, конвертация адресаций ("A1", "B2").
- Парсинг и генерация XML для рабочих книг и листов (8 часов):  
Реализовывались функции для чтения и записи основных XML-файлов, отвечающих за структуру книги, листов, текстовых значений (sharedStrings), числовых и булевых данных. Проверялась корректность сериализации/десериализации, реализованы классы XLSXReader и XLSXWriter.
- Реализация поддержки объединённых ячеек (merged cells) (4 часа):  
Изучалась специфика хранения слияний в XML, добавлялись методы для объединения/разъединения ячеек и корректного сохранения этих операций при экспорте и импорте.
- Реализация поддержки базовых стилей (4 часа):  
Описывались структуры для хранения информации о стилях (шрифты, цвета, выравнивания). Реализовывалась их привязка к ячейкам, парсинг и генерация XML для файла стилей (styles.xml).

- Отладка, тестирование, работа с краевыми случаями (9 часов): Библиотека проверялась на реальных файлах, созданных в разных редакторах (Excel, Google Sheets, Numbers). Исправлялись баги совместимости, обрабатывались граничные случаи (пустые строки, некорректные стили, сложные merge).
- Документирование библиотеки (1 час): Подготовлен README.md с архитектурным описанием, инструкциями по установке и использованию, примерами кода, визуализацией классов и пояснениями по работе с граничными случаями.

Итоговая трудоёмкость по мини-библиотеке: 38 часов.

### **1.3. Общая сумма затраченного времени**

На выполнение всего проекта (сайт + библиотека + отчётность и коммуникация) затрачено 74 часа.

## **2 Общая характеристика деятельности организации (заказчика проекта)**

**Наименование заказчика:**

**Московский Политех, кафедра «Инфокогнитивные технологии»**

**Организационная структура:** проектная команда из студентов направления Системная и программная инженерия.

**Описание деятельности:** Разработка цифровых обучающих решений и информационных ресурсов, направленных на популяризацию современных технологий в области цифровых финансов и продвижение знаний о безопасном использовании электронных инструментов в учебных и профессиональных сообществах. Особое внимание уделяется созданию просветительских материалов и программных продуктов, способствующих пониманию новых цифровых протоколов, включая протоколы электронных денег и цифровых наличных, а также вопросам информационной безопасности в условиях перехода к цифровой экономике.

### **3.1. Базовая часть: разработка и публикация статического сайта**

#### **3.1.1. Подготовка к работе и настройка системы контроля версий**

- Задачи:
  - Установка и настройка Git на локальном компьютере.
  - Создание личного удалённого репозитория на платформе GitHub для хранения всех файлов и контроля версий по проекту.
  - Инициализация локального репозитория, настройка связи с удалённым (команды `git init`, `git remote add`).
  - Регулярное ведение истории изменений: коммиты при каждом значимом обновлении структуры или содержимого сайта.
  - Проведение базовых операций (`git add`, `git commit`, `git push`, `git pull`), проверка корректности синхронизации между локальным и удалённым репозиториями.
- Промежуточный результат:

Надёжно организованная структура хранения всех материалов проекта, полный контроль версий, обеспечена прозрачность и воспроизводимость всей работы над сайтом.

#### **3.1.2. Освоение синтаксиса Markdown для проектной документации**

- Задачи:
  - Изучение основных элементов разметки Markdown (заголовки, списки, таблицы, вставки кода, изображения, ссылки).
  - Практическое оформление всех разделов сайта, технической документации, журнала и ресурсных страниц с помощью Markdown.
  - Адаптация Markdown-файлов под требования Hugo: оформление frontmatter, использование встроенных шаблонов Hugo для автоматического формирования структуры сайта.
  - Создание схем, таблиц, технических описаний с помощью встроенных возможностей Markdown и расширений (например, Mermaid.js для диаграмм).



- Промежуточный результат:  
Все тексты, инструкции и сопроводительные материалы оформлены в формате Markdown, что обеспечивает простоту сопровождения и единый стиль всей документации.

### **3.1.3. Изучение и настройка генератора статических сайтов Hugo**

- Задачи:
  - Ознакомление с документацией по Hugo, базовыми принципами работы генераторов статических сайтов.
  - Установка Hugo и тестовый запуск проекта на локальной машине.
  - Выбор подходящей темы оформления (PaperMod), её загрузка и интеграция в проект.
  - Изучение структуры каталогов и шаблонов Hugo (content, layouts, static, config.yaml).
  - Первичная настройка параметров сайта: установка метаданных, указание авторской информации, определение разделов и их структуры.
  - Адаптация шаблонов темы под требования сайта (например, настройка главной страницы, добавление собственных стилей или скриптов при необходимости).
- Промежуточный результат:  
Полностью работоспособная локальная версия сайта с корректно настроенными разделами и выбранным внешним видом.

### **3.1.4. Проектирование структуры и наполнение сайта контентом**

- Задачи:
  - Определение логической структуры сайта:
    - главная страница (аннотация и описание проекта),
    - раздел «О проекте» (подробная информация о тематике ODC),
    - блог/журнал (публикации о ходе работы),

- страница с ресурсами (технологии, полезные ссылки),
- страница участника (отражение индивидуального вклада).
- Разработка контента для каждого раздела: написание текстов, подготовка иллюстраций, создание схем и диаграмм (в т.ч. с использованием Mermaid.js).
- Формирование ленты новостей/журнала с хронологией выполнения ключевых этапов (например: «Старт проекта», «Добавлены схемы протокола», «Рефакторинг описания»).
- Оформление ссылок на используемые технологии, лицензионную информацию, указание источников и авторских прав.
- Проверка структуры внутренних ссылок, корректности навигации, метаданных для поисковых систем.
- Промежуточный результат:  
Готовый к публикации сайт с уникальным авторским наполнением, отражающий выбранную тему и полностью соответствующий техническому заданию.

### **3.1.5. Финальная отладка, тестирование и публикация сайта**

- Задачи:
  - Тестирование работоспособности всех разделов сайта в локальном режиме.
  - Проверка корректности отображения контента, ссылок, изображений и вёрстки на различных устройствах и браузерах.
  - Исправление выявленных ошибок и багов, оптимизация структуры страниц.
  - Настройка и запуск публикации сайта с помощью GitHub Pages:
    - настройка конфигурации для автоматической сборки и публикации (в том числе с помощью GitHub Actions)
  - Финальная проверка доступности сайта для внешних пользователей, тестовая загрузка с разных устройств.

- Промежуточный результат:  
Публично доступный, корректно работающий сайт с уникальным контентом, размещённый на платформе GitHub Pages.

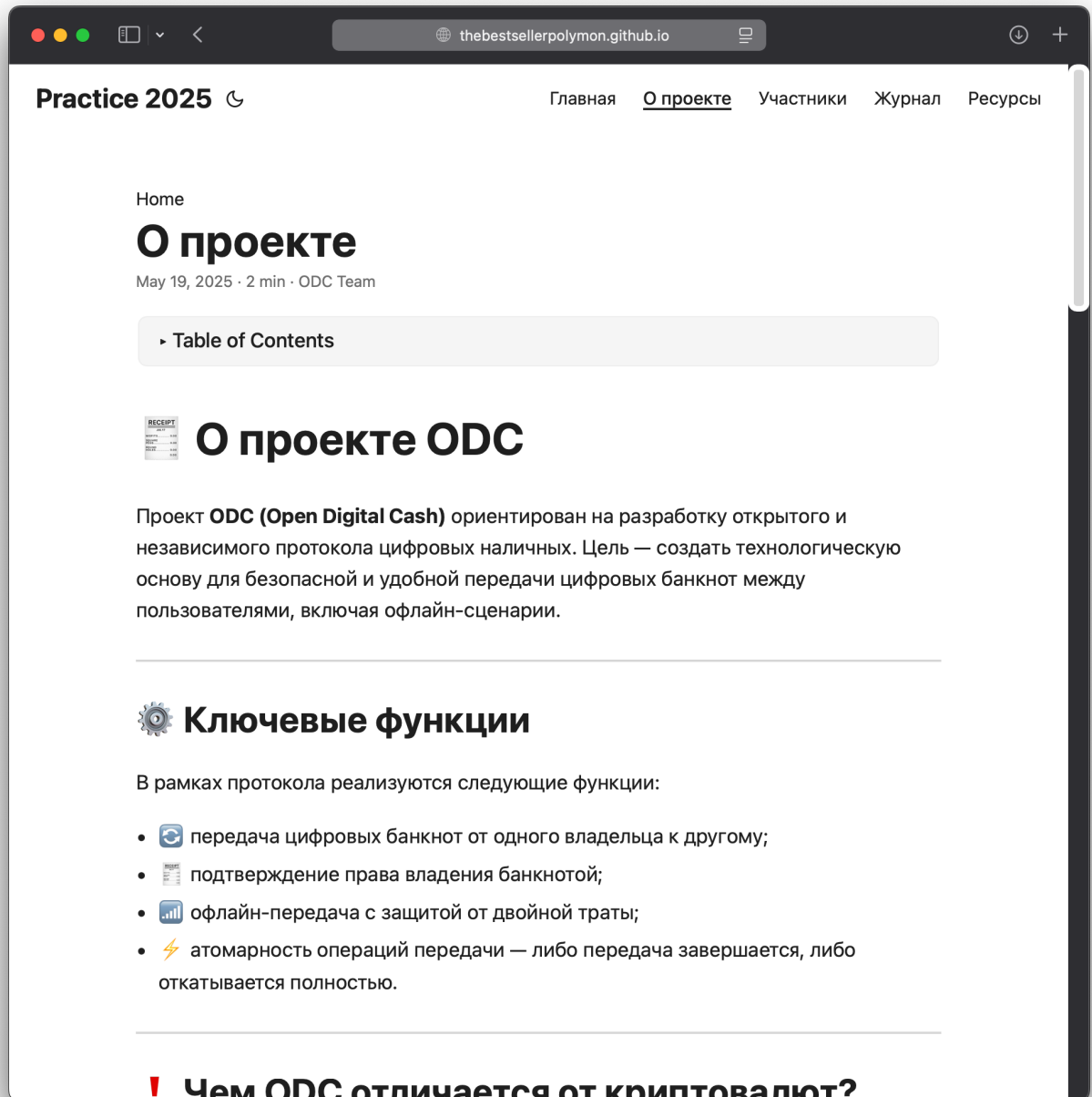


Рисунок 1 — скриншот сайта

Итоговые результаты по базовой части:

- Полноценный авторский статический сайт по теме цифровых наличных,
- Грамотно структурированная и оформленная проектная документация,

- Продемонстрированные навыки работы с современными инструментами (Git, Markdown, Hugo),
- Положительный опыт публикации и сопровождения IT-проекта,
- Подтверждённое участие в мероприятии отраслевого профессионального сообщества,
- Индивидуальный отчёт, отражающий качество и объём проделанной работы.

### **3.2. Вариативная часть: реализация мини-библиотеки для работы с XLSX на языке Swift**

Суть

задания:

Необходимо было самостоятельно реализовать мини-библиотеку для чтения и записи файлов формата XLSX (Excel) на языке Swift, не используя готовые сторонние библиотеки, с изучением структуры OpenXML и воспроизведением минимального набора функций для импорта/экспорта табличных данных.

Основные этапы работы:

#### **1. Изучение формата XLSX и постановка задачи**

- Анализ структуры файлов Excel (zip-архив, XML-файлы).
- Изучение спецификаций OpenXML, просмотр примеров реальных файлов, разбор основных составляющих (workbook.xml, sharedStrings.xml, отдельные листы).
- Формулировка минимально необходимого набора возможностей для библиотеки (чтение/запись ячеек, поддержка текста, чисел, базовые стили и объединения).

#### **2. Проектирование основных структур данных**

- Описание и реализация базовых сущностей: Workbook (рабочая книга), Sheet (лист), Cell (ячейка).
- Создание перечислений для хранения разных типов ячеек (CellType).
- Определение способов адресации ячеек (например, преобразование “A1” в индексы).

#### **3. Реализация парсинга и генерации XML**

- Программная работа с XML для чтения и записи основных структур XLSX:

- Чтение workbook.xml (имена и id листов),
- Чтение/запись sharedStrings.xml (работа с текстовыми ячейками),
- Парсинг sheetX.xml (чтение всех типов ячеек: text, number, bool, inlineStr),
- Генерация аналогичных XML при экспорте таблиц.
- Тестирование на простых примерах: чтение и запись файлов с минимальными таблицами.

#### 4. Поддержка объединённых ячеек (merge)

- Изучение, как Excel хранит информацию о слиянии ячеек (mergeCells, mergeCell).
- Добавление функций для объединения и разъединения ячеек, корректная запись диапазонов объединения в XML.
- Проверка корректной обработки объединённых областей при чтении и сохранении.

#### 5. Поддержка базовых стилей ячеек

- Анализ структуры файла styles.xml, выбор реализуемых параметров (например, шрифты, цвет, выравнивание).
- Создание структур для стилей (Style, Font, Fill, Color и т.п.).
- Привязка стиля к каждой ячейке, отражение стилей в XML.
- Проверка того, что базовые стили сохраняются и читаются корректно.

#### 6. Тестирование, отладка, исправление ошибок

- Многократные проверки работы библиотеки на разных реальных файлах Excel.
- Исправление ошибок совместимости (особенно с текстовыми ячейками, inlineStr, sharedStrings).
- Ловля багов при чтении/записи нестандартных случаев (пустые ячейки, стили, большие таблицы).
- Проверка совместимости с Google Sheets и Numbers.

#### 7. Оформление технической и пользовательской документации

- Написание README.md для проекта:

- Описание архитектуры, используемых структур и классов,
- Примеры кода по основным сценариям (создание книги, листа, запись и чтение ячеек, стили, merge),
- Вставка фрагментов XML и пояснения к ним,
- Перечисление ограничений и известных отличий от «большого» Excel.

Итог: реализована минимальная, но рабочая библиотека для чтения и записи XLSX-файлов на Swift с базовой поддержкой типов данных, объединённых ячеек и стилей, протестированная на реальных документах. Все исходные коды и примеры использования оформлены и задокументированы в репозитории.

#### **4. Описание достигнутых результатов по проектной практике**

В ходе выполнения проектной практики были достигнуты следующие конкретные результаты, соответствующие поставленным задачам и индивидуальному заданию:

##### **4.1. В области разработки и публикации статического сайта**

- Разработан и опубликован авторский статический сайт по теме "ODC — открытый протокол цифровых наличных" с использованием генератора Hugo и темы PaperMod.
- Структура сайта полностью соответствует техническому заданию: реализованы главная страница (аннотация и цели), раздел "О проекте" с подробным тематическим описанием, блог/журнал разработки (минимум три публикации), страница с ресурсами и технологиями, отдельная страница об участнике.
- Весь контент сайта оформлен в формате Markdown. Для иллюстрации структуры протокола и отдельных этапов использованы схемы и диаграммы, в том числе с помощью Mermaid.js.
- Сайт протестирован и размещён на платформе GitHub Pages, доступен для просмотра по публичной ссылке.
- Вся работа по сайту задокументирована в личном репозитории GitHub, с историей изменений.

##### **4.2. В области реализации мини-библиотеки для работы с XLSX на Swift**

- Реализована с нуля минимальная библиотека для чтения и записи файлов XLSX (Excel) на языке Swift, без использования сторонних библиотек.
- Библиотека поддерживает основные функции: чтение и запись текстовых, числовых и логических ячеек, работу с объединёнными ячейками, а также базовые стили (шрифты, цвет, выравнивание).
- Выполнено тестирование библиотеки на различных файлах, в том числе созданных в Microsoft Excel, Google Sheets и Apple Numbers. Проверена

совместимость и корректность обработки edge-cases (пустые ячейки, стили, слияния).

- Вся архитектура и порядок работы библиотеки подробно описаны в README.md, с примерами кода, диаграммами структуры и пояснениями по возможным ограничениям.
- Исходный код и документация размещены в репозитории на GitHub, доступны для проверки и использования.



## **Заключение**

В ходе прохождения проектной практики задачи, определённые индивидуальным заданием, были успешно реализованы в полном объёме. Были получены и практически закреплены навыки разработки программных решений, включая проектирование, программирование, тестирование, публикацию и документирование. В результате выполнения работы создан и опубликован авторский статический сайт, посвящённый тематике цифровых наличных и реализованный с использованием современных инструментов и технологий. Кроме того, проведена самостоятельная реализация мини-библиотеки для работы с файлами XLSX на языке Swift, что обеспечило углублённое изучение стандартов структурирования и обмена табличными данными.

Выполнение всех этапов практики способствовало формированию устойчивых компетенций в области проектирования IT-решений, работы с системами контроля версий, генерации и сопровождения документации, а также освоению методов взаимодействия с современными инструментами для разработки и публикации программных продуктов. Практические результаты могут быть использованы в образовательных целях: статический сайт — как тематический ресурс по вопросам цифровых наличных, а мини-библиотека — как пример реализации прикладных инструментов для работы с табличными данными на языке Swift.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Office Open XML. Part 1. Fundamentals and Markup Language Reference [Электронный ресурс]. — ISO/IEC 29500-1:2016. — Режим доступа: <https://www.iso.org/standard/71691.html> (дата обращения: 15.05.2025).
2. Документация по Hugo [Электронный ресурс]. — Режим доступа: <https://gohugo.io/documentation/> (дата обращения: 18.05.2025).
3. Документация по Git [Электронный ресурс]. — Режим доступа: <https://git-scm.com/doc> (дата обращения: 18.05.2025).
4. Документация по Swift [Электронный ресурс]. — Режим доступа: <https://docs.swift.org/swift-book/> (дата обращения: 18.05.2025).
5. Документация по Mermaid.js [Электронный ресурс]. — Режим доступа: <https://mermaid-js.github.io/mermaid/#/> (дата обращения: 18.05.2025).
6. Репозиторий темы PaperMod для Hugo [Электронный ресурс]. — Режим доступа: <https://github.com/adityatelange/hugo-PaperMod> (дата обращения: 17.05.2025).
7. GitHub Pages Documentation [Электронный ресурс]. — Режим доступа: <https://docs.github.com/en/pages> (дата обращения: 17.05.2025).
8. Microsoft. Open XML SDK Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/en-us/office/open-xml/open-xml-sdk> (дата обращения: 16.05.2025).