

Санкт-Петербургский государственный университет

Технологии программирования

Группа 20.Б10-мм

Бакова Елена Зауровна

Редактор рабочих программ дисциплин

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
доцент кафедры системного программирования СПбГУ
к.т.н. Литвинов Ю.В.

Санкт-Петербург
2023

Оглавление

Введение	3
Постановка задачи	4
1. Обзор	5
1.1. Существующие решения	5
1.1.1. Конструктор образовательных программ	5
1.1.2. RpdPatcher	7
1.1.3. Редакторы РПД других университетов	7
1.2. Используемые технологии	8
1.2.1. ASP.NET Core	8
1.2.2. Python	8
1.2.3. React	8
1.2.4. Docker	9
2. Описание решения	10
2.1. Фронтенд-часть	11
2.2. Бэкенд-часть	13
2.3. Разбор DOCX-файла	14
3. Апробация	15
Заключение	16
Список литературы	17

Введение

СПбГУ имеет довольно сложную структуру. Сюда входят и учебно-методические комиссии по укрупненной группе (укрупненным группам) специальностей и направлений подготовки¹. Они занимаются вопросами организации учебного процесса, разработки и предварительной экспертизы учебно-методической документации (учебные планы, рабочие программы дисциплин и иные методические документы), а также следят за качеством образования.

РПД — рабочая программа дисциплины — один из методических документов, используемых в университете. Рабочая программа дисциплины имеет определенную структуру, регулируемую приказом. Созданием РПД занимаются преподаватели соответствующего курса. Далее документ отправляется на проверку в учебно-методическую комиссию. При наличии замечаний, файл возвращается на доработку. В данный момент соответствие документа структуре проверяется вручную, что значительно увеличивает объем работы, выполняемой человеком, а также усложняет процесс создания документации. Отсюда возникла потребность в автоматизации процесса экспертизы и создания РПД.

В других вузах уже имеются свои средства создания и автоматической проверки РПД, однако рабочая программа дисциплины в СПбГУ имеет довольно специфическую структуру, что не позволяет использовать уже имеющиеся решения.

Поэтому требуется создать веб-приложение, в котором преподаватель вместо того, чтобы отправлять документ на проверку в УМК, будет загружать в редактор DOCX-файл, затем будет проведена проверка на соответствие структуре и имеющиеся замечания отобразятся пользователю. Такое приложение должно упростить процедуру валидации документа: преподаватель быстро может проверить свой текст, а учебно-методическая комиссия сможет быстрее принимать документы.

¹Сайт с УМК СПбГУ <https://spbu.ru/universitet/podrazdeleniya-i-rukovodstvo/uchebno-metodicheskie-komissii> (дата обращения 07.12.2022)

Постановка задачи

Целью данной учебной практики является создание редактора РПД. В осеннем семестре работа заключалась в создании верификатора РПД, в весеннем семестре — в создании редактора рабочих программ дисциплин. Для достижения цели были поставлены следующие задачи:

Осенний семестр:

1. Провести обзор существующих решений.
2. Изучить используемые технологии.
3. Реализовать проверку загруженного файла на соответствие структуре.
4. Реализовать фронтенд-часть сервиса.
5. Контейнеризовать приложение.
6. Провести апробацию.

Весенний семестр:

1. На странице проверки добавить возможность загружать несколько файлов.
2. Добавить страницу редактирования.
3. Реализовать отображение содержимого файла по разделам.
4. Реализовать создание и сохранение DOCX-файла с форматированием.
5. Контейнеризовать приложение.
6. Провести апробацию.

1. Обзор

1.1. Существующие решения

Обзор проводился с целью изучения уже имеющихся решений. Обзор осуществлялся по следующим критериям:

- наличие возможности проверить готовый DOCX-файл;
- проверка структуры файла (наличие всех необходимых разделов, корректное их наименование, правильное оформление титульной страницы);
- наличие возможности редактировать или создавать РПД;
- отсутствие необходимости установки приложения на компьютер.

1.1.1. Конструктор образовательных программ

Сервис, разработанный Университетом ИТМО, имеет интерфейсы для создания (например, рисунок 2) и проверки учебных планов, рабочих программ и образовательных программ. Однако сервис доступен только для работников Университета ИТМО, сторонние пользователи не могут использовать функциональные возможности Конструктора ОП.

Конструктор образовательных программ не предоставляет возможности загрузки и проверки готовых РПД, некоторые данные (такие как название дисциплины, язык и форма её реализации) проверяются автоматически на этапе разработки, а затем заполненная РПД отправляется на ручную проверку [7].

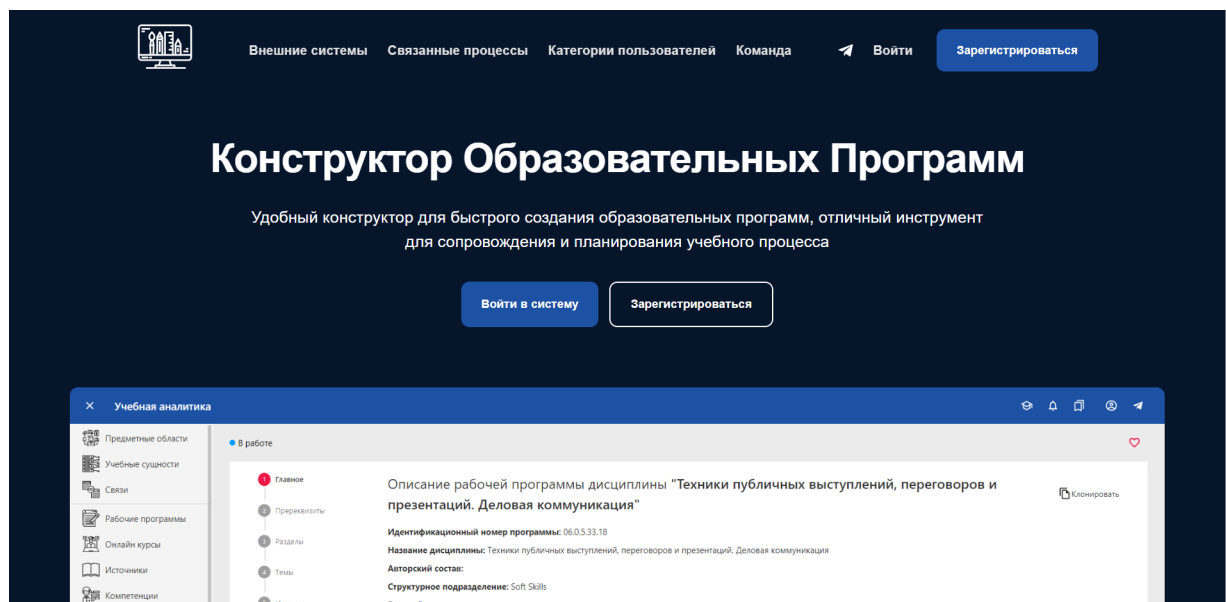


Рис. 1: Главная страница Конструктора ОП [13]

Рис. 2: Интерфейс для создания рабочей программы

1.1.2. RpdPatcher²

Утилита, позволяющая проверять наличие необходимых разделов и их корректное наименование, отсутствие запрещенных слов (например, использование слова «студент» вместо «обучающийся»³), в некоторых разделах делаются проверки на наличие стандартной фразы (например, в разделе 3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса, строго должна быть фраза «Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке»), также есть команды для автоматического переименования файлов и исправления типичных ошибок в них, однако работа с приложением осуществляется через консоль, то есть требует установки на компьютер.

1.1.3. Редакторы РПД других университетов

Доступ к редакторам рабочих программ дисциплин других университетов возможен только для авторизованных пользователей (сотрудников конкретного учебного заведения), например рисунок 3.

Таким образом, в ходе обзора не было найдено веб-приложений, позволяющих загружать готовый DOCX-файл для автоматической проверки по критериям, либо для его редактирования.

²RpdPatcher на GitHub: <https://github.com/yurii-litvinov/TeachingScripts/tree/main/RpdPatcher> (дата обращения 08.12.2022)

³По существующим в СПбГУ рекомендациям РПД пишутся так, чтобы быть переиспользуемыми и для студентов, и для аспирантов, поэтому используется нейтральное слово «обучающийся».

Вход в редактор РПД

Версия программы: 1.0.8208.28657

Дата компиляции: 22.06.2022 15:55:16

Логин:

Пароль:

☐ Запомнить меня

[Войти](#)

Рис. 3: Страница авторизации редактора РПД СПбГАСУ [12]

1.2. Используемые технологии

В этом разделе описаны используемые технологии.

1.2.1. ASP.NET Core

ASP.NET Core [3] — фреймворк, применяемый для создания веб-приложений. Использовалась модель Web API — наиболее подходящая модель для создания одностраничных приложений, представляющая из себя набор контроллеров, соответствующих одному HTTP-запросу. Язык реализации C# был выбран, так как автор уже был знаком с данным языком программирования.

1.2.2. Python

Python — реализация скрипта для разбора DOCX-файлов.

1.2.3. React

Для создания фронтенд-части сервиса был использован язык TypeScript, библиотека React [9], а также Material UI [2] — библиотека

React-компонентов. React — один из самых популярных фреймворков для разработки веб-интерфейсов. Благодаря наличию синтаксиса JSX код становится менее громоздким и более простым для написания (по сравнению с другими фреймворками, например, Angular или Vue.js).

1.2.4. Docker

Для развертывания приложения, включающего в себя несколько микросервисов, используется контейнеризация. Также она позволяет обеспечить кроссплатформенность, то есть работу приложения вне зависимости от операционной системы хоста. Для этого был использован Docker [1]. Для каждого микросервиса создаётся образ, где настроено окружение и содержатся все зависимости. Экземпляром образа является контейнер, и с помощью Docker Compose эти контейнеры могут запускаться одновременно, что позволяет быстро и удобно развернуть приложение.

2. Описание решения

На рисунке 4 представлена диаграмма компонентов веб-сервиса. Веб-приложение можно разделить на три компонента:

- Фронтенд-часть:
 - загрузка файлов,
 - отправка файлов на сервер,
 - отображение ошибок,
 - отображение содержимого файла,
 - редактирование текста,
 - формирование и скачивание DOCX-файла.
- Бэкенд-часть:
 - получение загруженного файла,
 - обработка файла,
 - отправка списка ошибок на фронтенд,
 - отправка содержимого файла на фронтенд.
- Скрипт для разбора DOCX-файла:
 - разбор документа,
 - формирование двух JSON-файлов с содержанием и структурой файла.

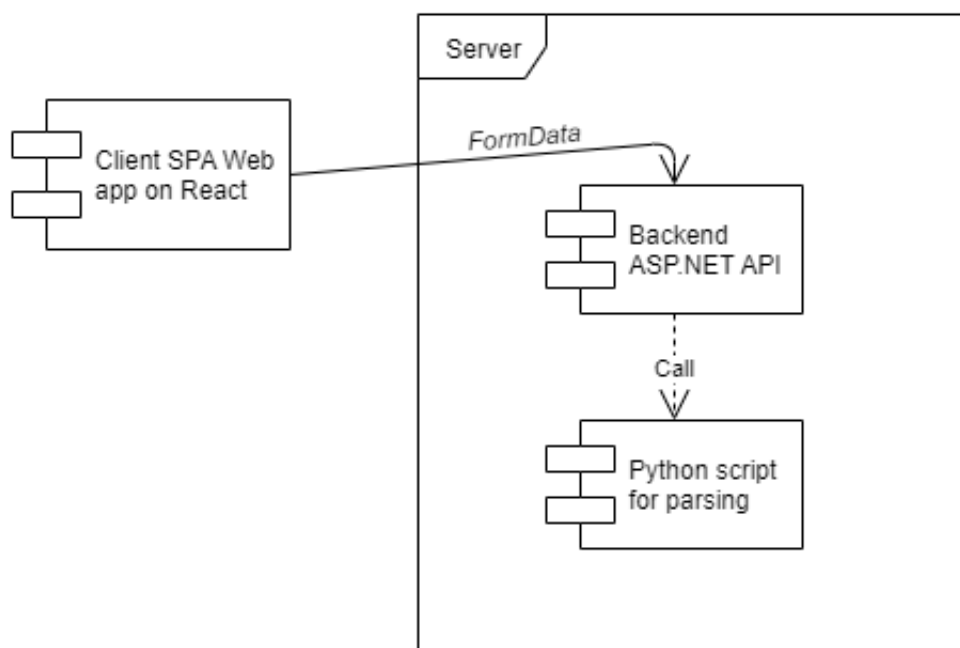


Рис. 4: Диаграмма компонентов приложения

На странице проверки пользователь загружает документы, которые после нажатия кнопки «Проверить» отправляются на бэкенд, где происходит разбор файлов, проверка оформления титульной страницы, наличия необходимых разделов и их правильного наименования, затем сообщения об ошибках возвращаются в виде ответа на фронтенд, где списком отображаются пользователю.

На странице редактирования пользователь загружает документ, который отправляется на бэкенд, где происходит парсинг файла, и затем содержимое файла отображается на фронтенде. В случае если был загружен пустой файл, пользователю отобразится шаблон РПД.

2.1. Фронтенд-часть

Приложение состоит из двух страниц: страница проверки, где можно загрузить файл и получить список сообщений об ошибках (например, рисунок 5), а также страница редактирования, где можно загрузить РПД, отредактировать её, а затем скачать как DOCX-файл (например, рисунок 6). На странице редактирования, после того как пользователь

нажимает кнопку «Сохранить», формируется DOCX-файл с форматированием титульной страницы и заголовков разделов (см. рисунок 7). Для генерации файла использовалась библиотека docx.js [10], использующая стандарт Office Open XML [11].

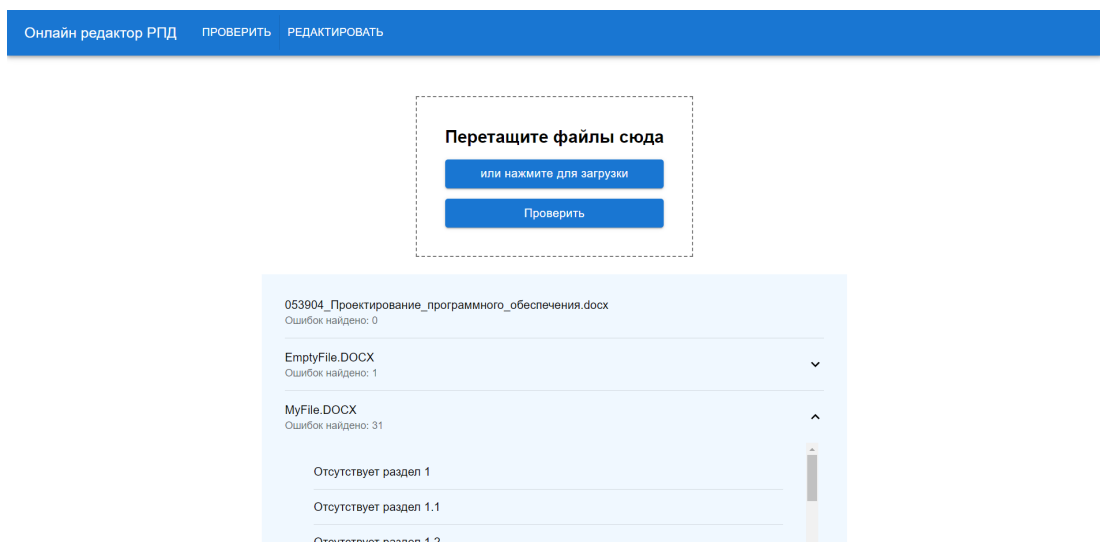


Рис. 5: Пример выводимого списка ошибок

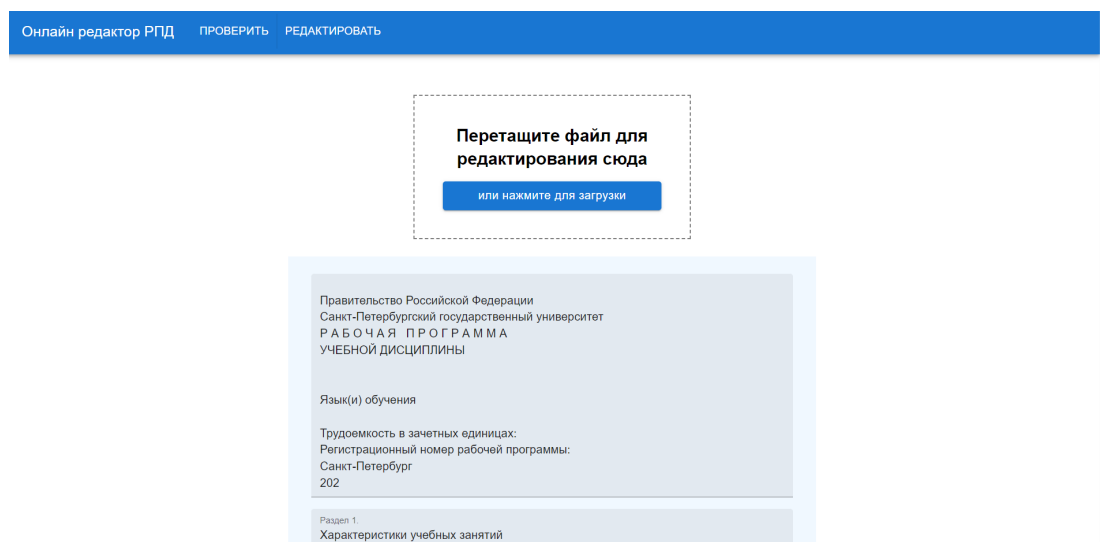


Рис. 6: Пример отображения содержимого файла

Правительство Российской Федерации Санкт-Петербургский государственный университет РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Язык(и) обучения Трудоемкость в зачетных единицах: Регистрационный номер рабочей программы: Санкт-Петербург 202	Раздел 1. Характеристики учебных занятий 1.1. Цели и задачи учебных занятий 1.2. Требования к подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты) 1.3. Перечень результатов обучения (learning outcomes) 1.4. Перечень и объём активных и интерактивных форм учебных занятий Раздел 2. Организация, структура и содержание учебных занятий 2.1. Организация учебных занятий 2.1.1 Основной курс 2.2. Структура и содержание учебных занятий Раздел 3. Обеспечение учебных занятий 3.1. Методическое обеспечение 3.1.1 Методические указания по освоению дисциплины 3.1.2 Методическое обеспечение самостоятельной работы 3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания 3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства) 3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса 3.2. Кадровое обеспечение 3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий 3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом
---	--

Рис. 7: Пример формируемого файла

Созданный компонент загрузки состоит из:

- Области, в которую можно перетащить нужные файлы:
 - при загрузке не DOCX-файлов появится сообщение об ошибке,
 - основан на drag and drop событиях мыши [8].
- Кнопки выбора файла:
 - использует HTML элемент `<input>` [4],
 - можно выбрать один или несколько файлов с расширением .docx.

2.2. Бэкенд-часть

После загрузки файла на бэкенд отправляется POST-запрос [5] с документом, затем вызывается скрипт для парсинга файла. В результате парсинга получается два файла со структурой и содержанием файла

(см. рисунок 8) с помощью которых производится проверка и формирование ответа к запросу.

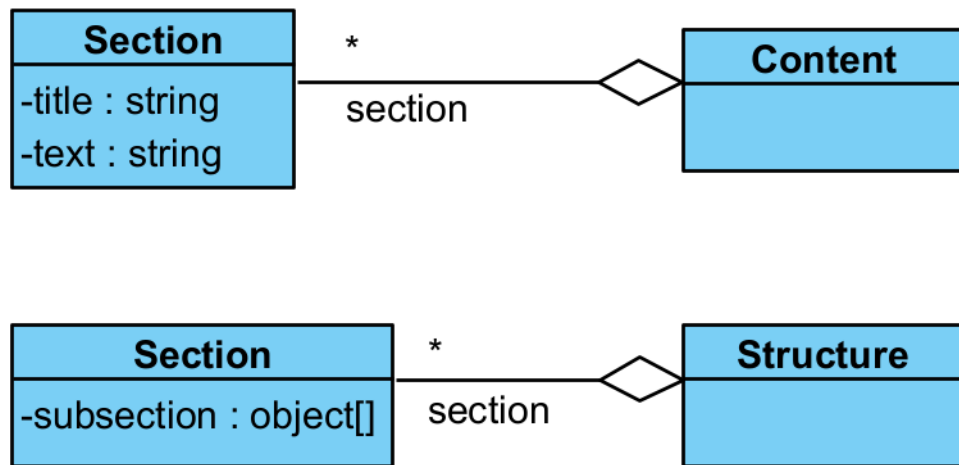


Рис. 8: Диаграмма классов файла с содержанием и со структурой документа

Производятся следующие проверки:

- правильное оформление титульной страницы;
- наличие всех необходимых разделов;
- отсутствие слова «студент» в разделах;
- правильное наименование каждого раздела.

2.3. Разбор DOCX-файла

Для разбора РПД была использована программа, реализованная другой студенткой⁴. Это веб-приложение, написанное на Python, было переделано в скрипт для использования в данной работе. Была удалена клиентская часть, также были исправлены некоторые ошибки, найденные в процессе использования скрипта.

⁴Код приложения на GitHub: <https://github.com/nyutamaksimova/DisciplineWorkProgram> (дата обращения 09.12.2022)

3. Апробация

Апробация проводилась с использованием опросника System Usability Scale [6]. Участникам было предложено воспользоваться сервисом, отправив на проверку различные готовые РПД и другие файлы. В опросе участвовало 3 студента и 2 преподавателя. Общая оценка составила 93,0, что показывает то, что система проста и удобна в использовании. Некоторые недочеты, найденные в процессе апробации, были исправлены.

Заключение

В рамках этой учебной практики были решены следующие задачи:

- Проведен обзор существующих решений.
- Изучены используемые технологии.
- Реализована проверка загруженного файла на соответствие структуре.
- Реализована фронтенд-часть сервиса.
- На странице проверки добавлена возможность загружать несколько файлов.
- Добавлена страница редактирования.
- Реализовано отображение содержимого файла по разделам.
- Реализовано создание и сохранение DOCX-файла с форматированием.
- Контейнеризовано приложение.
- Проведена апробация.

Код доступен в GitHub репозитории⁵.

⁵GitHub репозиторий проекта. Ссылка: <https://github.com/ElenaBakova/DWP-Editor>

Список литературы

- [1] Docker Inc. Официальный сайт Docker. — URL: <https://www.docker.com/> (дата обращения: 08.12.2022).
- [2] Material UI. Официальный сайт библиотеки Material UI. — URL: <https://mui.com/> (дата обращения: 14.11.2022).
- [3] Microsoft. Официальный сайт ASP.NET. — URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения: 14.11.2022).
- [4] Mozilla. Официальный сайт документации MDN. — URL: <https://developer.mozilla.org/en-US/docs/web/html/element/input> (дата обращения: 10.12.2022).
- [5] RFC 2616. — URL: <https://www.rfc-editor.org/rfc/rfc2616#section-9.5> (дата обращения: 10.12.2022).
- [6] Sauro Jeff. Measuring Usability with the System Usability Scale (SUS). — URL: <https://measuringu.com/sus/> (дата обращения: 15.12.2022).
- [7] Говоров А. И. Бабаянц К. П. Говорова М. М. Хлопотов М. В. Деркунская С. Чернышева А. В. Коряков С. А. Артамонова В. Алгоритмы экспертизы и верификации рабочих программ дисциплин для использования в информационных системах // Вестник Астраханского государственного технического университета. — 2021. — С. 6–10.
- [8] Официальная документация по событиям мыши. — URL: <https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent> (дата обращения: 09.12.2022).
- [9] Официальный сайт библиотеки React. — URL: <https://reactjs.org/> (дата обращения: 14.11.2022).
- [10] Официальный сайт библиотеки docx.js. — URL: <https://docx.js.org/#/> (дата обращения: 11.05.2023).

- [11] Официальный сайт с документацией ECMA. — URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-376/> (дата обращения: 11.05.2023).
- [12] Сайт редактора РПД. — URL: <https://web-rpd.spbgasu.ru> (дата обращения: 07.12.2022).
- [13] Факультет ИКТ. Сайт Конструктора ОП. — URL: <https://op.itmo.ru/> (дата обращения: 25.10.2022).