

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет \_Информационных\_Технологий\_  
Кафедра «\_Инфокогнитивные\_технологии\_»

Направление подготовки/ специальность: \_Информатика и вычислительная техника/\_  
Системная и программная инженерия\_

## ОТЧЕТ

по проектной практике

Студент: \_Мусаелян\_Анжелика\_\_\_\_\_ Группа: \_241-3211\_\_\_\_\_

Место прохождения практики: Московский Политех, кафедра \_\_\_\_\_  
\_\_\_\_\_Инфокогнитивные\_технологии \_\_\_\_\_

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

Руководитель практики: \_Чернова\_Вера\_Михайловна\_\_\_\_\_

Москва 2025

## ОГЛАВЛЕНИЕ

### ВВЕДЕНИЕ

1. Общая информация о проекте:
  - Название проекта
  - Цели и задачи проекта
2. Общая характеристика деятельности организации (*заказчика проекта*)
  - Наименование заказчика
  - Организационная структура
  - Описание деятельности
3. Описание задания по проектной практике
4. Описание достигнутых результатов по проектной практике

ЗАКЛЮЧЕНИЕ (*выводы о проделанной работе и оценка ценности выполненных задач для заказчика*)

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

## ВВЕДЕНИЕ

### 1. Общая информация о проекте.

Название проекта: Компьютерный практикум по математике программа «Mathsem»

Цель проекта:

Целью проекта является создание программного продукта для обучения студентов математической логике и теории алгоритмов. Программа должна уметь в интерактивном режиме строить определения и теоремы из дискретной математики. Нужно разработать интерфейс и алгоритмы для работы программы. Целью является также написание сайта с описанием программного продукта, с возможностью скачать программу, с помощью по программе. База данных на сайте должна хранить формулы, построенные пользователями.

Задачи проекта:

Задачи, связанные с представлением данных в программе, с написанием алгоритмов для построения формул и их анализа.

1. Приведение к префиксной и сколемовской нормальной форме (синтаксический анализ формул).
2. Проанализировать структуру данных для хранения формул в программе, разобраться как формируется файл \*.fms.
3. Проанализировать как записываются формулы в \*.docx.
4. Разобраться, каким образом строятся схемы в VUE.
5. Определение эквивалентности формул с точностью до замены переменных.
6. Построить в прототипе программы определение функции, область определения функции, область значения функции, определение отношения, определение графа, транзитивные, симметричные, рефлексивные свойства.

7. Построить формулу  $A \subset (A \cup B)$  в прототипе программы.
8. Реализовать ввод строки с символами теории множеств.
9. Реализовать алгоритм, преобразовывающий введенную строку в логическую формулу.
10. Реализовать алгоритм, определяющий истинность формул.

#### Задачи для Базы данных

1. Таблица с пользователями.
2. Таблица с файлами с формулами.
3. Разработать структуру для хранения формул на сайте и в базе данных.

#### Задачи для сайта

1. Разработать прототип сайта в Figma.
2. Сверстать страницы сайта по прототипу.
3. Написать бэкенд для реализации регистрации, добавления отзывов.
4. Создать базу данных для хранения построенных пользователями формул, интегрировать ее в сайт.

2. Общая характеристика деятельности организации.

Заказчиком данного проекта является Московский Политех.


### 3. Описание задания по проектной практике.

#### Создание 3D-игры "Spirits & Crystals"

Spirits & Crystals - это 3D-приключенческая игра с элементами головоломки, где игрок управляет привидением в поисках магических кристаллов в таинственном лесу.

Технологический стек:

 Игровой движок: Babylon.js

 Фронтенд: React + Vite

 3D-моделирование: Blender

 Управление: WASD + взаимодействие (E)

Особенности игры:

#### 1. Атмосферный мир:

- Ночной лес с динамическим освещением
- Партиклы свечения вокруг кристаллов
- Реалистичные тени и эффекты тумана

#### 2. Система взаимодействий:

- Контекстные подсказки при приближении к объектам
- Постепенное усложнение расположения кристаллов
- Интерактивные элементы окружения

## ЗАКЛЮЧЕНИЕ

### 1. По веб-сайту (HTML, SASS, CSS, JS):

- Создан полностью адаптивный сайт с mobile-first подходом, работающий на экранах от 320px до 1920px
- Реализована кросс-браузерная совместимость (Chrome, Firefox, Edge)
- Достигнута высокая скорость загрузки (оценка PageSpeed 92/100)
- Разработаны интерактивные элементы

### 2. По 3D-игре (Vite+React, Babylon.js, Blender):

- Создан работающий 3D-движок на Babylon.js с интеграцией в React
- Смоделировано и оптимизировано 5 low-poly объектов в Blender
- Реализованы основные игровые механики: движение WASD, сбор предметов, таймер
- Достигнута стабильная производительность 60 FPS на средних ПК
- Размер финальной сборки составил 4.7MB (с gzip-сжатием)

### 3. Приобретенные профессиональные навыки:

- Глубокая работа с 3D-графикой: моделирование, текстурирование, экспорт
- Продвинутая фронтенд-разработка: настройка Vite, работа с WebGL
- Оптимизационные техники: LOD-модели, кэширование, прогрессивная загрузка
- Навыки проектного управления: постановка задач, тестирование, документирование

Вывод: Практика позволила создать два полноценных проекта, значительно углубить знания в 3D-графике и оптимизации веб-приложений, а также получить ценный опыт решения сложных технических задач. Все поставленные цели были достигнуты с превышением базовых требований по производительности и пользовательскому опыту.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Дэвид Флэнаган "JavaScript. Подробное руководство"  
<https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/>
2. Официальная документация по React  
<https://react.dev/learn>
3. Документация Babylon.js  
<https://doc.babylonjs.com/>
4. Руководство по Blender 3.0  
<https://docs.blender.org/manual/ru/latest/>
5. SASS/SCSS документация  
<https://sass-lang.com/documentation>
6. Vite.js официальная документация  
<https://vitejs.dev/guide/>
7. WebGL 2.0 Specification  
<https://www.khronos.org/registry/webgl/specs/latest/2.0/>
8. HTML Living Standard  
<https://html.spec.whatwg.org/multipage/>
9. CSS Specifications  
<https://www.w3.org/Style/CSS/>
10. MDN Web Docs (JavaScript, Web APIs)  
<https://developer.mozilla.org/ru/>
11. Game Development Patterns  
<https://gameprogrammingpatterns.com/>
12. Оптимизация производительности веб-приложений  
<https://web.dev/learn/>
13. Three.js и Babylon.js сравнение  
<https://medium.com/@babylonjs/comparing-three-js-and-babylon-js-features-81a5b13b3d01>

14.Официальный блог Vite

<https://vitejs.dev/blog/>

15.Руководство по адаптивному дизайну

<https://www.smashingmagazine.com/guides/responsive-web-design/>