**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования  
 «Кемеровский государственный университет»

Институт цифры

Кафедра ЮНЕСКО по информационным вычислительным технологиям

**ОТЧЕТ**

О выполнении семестровой работы по дисциплине «Языки программирования»

проект «Space Object Collections»

(название проекта)

студента 2 курса

Козырева Андрея Денисовича

(ФИО полностью)

направления подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

направленность (профиль) подготовки «Информатика и компьютерные науки»,

Руководитель практики:

канд. физ.-мат. наук, доцент

К.С. Иванов

(ученая степень, звание, ФИО)

канд. физ.-мат. наук, доцент

А. И. Зимин

(ученая степень, звание, ФИО)

Работа защищена:

Козырев А.Д. « » 2023 г.

(ФИО)

Кемерово 2023

Оглавление

[1. Тема проекта 3](#_Toc138055785)

[2. Архитектура 4](#_Toc138055786)

[3. Технологии 6](#_Toc138055787)

[4. Демонстрация работы 7](#_Toc138055788)

# Тема проекта

Основная идея: Создание веб-приложения, которое предоставляет информацию о космических объектах и позволяет пользователю взаимодействовать с ними.

Используемый HTTP-API:

NASA API для получения данных о космических объектах:

* APOD: Астрономическое изображение дня
* Asteroids NeoWs: Веб-сервис ближайших объектов к Земле
* Mars Rover Photos: Изображения, собранные роверами NASA Curiosity, Opportunity и Spirit на Марсе

Требования к функционалу:

1. После ввода имени и нажатия кнопки "Далее" на странице появляется приветственная картинка, полученная через сервис APOD (Astronomy Picture of the Day). Далее на странице отображаются два раздела, которые можно раскрыть и скрыть.
2. Раздел "Астероиды":
   1. При нажатии кнопки "Создать таблицу" происходит загрузка списка астероидов с помощью метода getAsteroids() из API NASA и отображение их в таблице.
   2. Возможность отфильтровать астероиды по опасности с помощью флажка "Показать только опасные астероиды".
   3. Возможность отсортировать астероиды по имени или размеру с помощью выпадающего списка "Сортировать по".
3. Раздел "Галерея фотографий Марса":
   1. При нажатии кнопки "Далее" происходит загрузка фотографий Марса с помощью метода getMarsPhotos(count) из API NASA и отображение их в виде миниатюр.
   2. При нажатии на миниатюру открывается модальное окно с увеличенным изображением фотографии.
   3. Возможность пролистывать фотографии с помощью клавиш стрелок влево и вправо или с помощью кнопок в модальном окне.

# Архитектура

Код данного проекта имеет модульную архитектуру, которая разделяет функциональность на отдельные модули и классы для достижения модульности и повторного использования кода. Ниже приведены основные модули, классы и функции, используемые в проекте:

* api.js: содержит функции для взаимодействия с API для получения информации о фотографиях космоса, астероидах и фотографиях Марса.
* utils.js: содержит вспомогательные функции, используемые в других модулях, например, функцию getRandomElements, которая возвращает случайные элементы из массива.
* asteroids-ui.js: определяет класс AsteroidsUI, который управляет взаимодействием секции "Астероиды" в веб-приложении. Он использует функции из api.js и utils.js для получения данных об астероидах, их сортировки и фильтрации, а затем отображает эти данные в таблице на веб-странице.
* mars-ui.js: определяет класс MarsPhotoGallery, который управляет взаимодействием секции "Галерея фотографий Марса". Он также использует функции из api.js для получения фотографий Марса и отображения их в виде миниатюр на веб-странице.
* ui.js: содержит функцию astronomyPictureUI, которая отвечает за взаимодействие секции "Astronomy Picture of the Day" на веб-странице.
* Файлы index.html и styles.css содержат разметку и стили для веб-страницы соответственно.

Основные функции:  
displayAsteroids(): Этот метод класса AsteroidsUI отображает данные о близких к Земле астероидах на веб-странице. Он фильтрует и сортирует астероиды в соответствии с выбранными пользователем настройками и создает таблицу с отсортированными данными. Затем он вставляет созданную таблицу в указанный элемент на странице.

fetchAndDisplayMarsPhotos(): Этот метод класса MarsPhotoGallery выполняет запрос к API NASA, используя функцию getMarsPhotos(count), для получения фотографий с поверхности Марса. Затем он отображает полученные фотографии в виде миниатюр на веб-странице и назначает обработчики событий для открытия модального окна с увеличенной фотографией при клике на миниатюру.

displayAstronomyPictureContent(name, data): Этот метод класса AstronomyPictureUI отображает информацию о астрономическом контенте дня на веб-странице. Он принимает имя пользователя name и объект data, содержащий данные об астрономическом контенте, и генерирует соответствующие элементы HTML для отображения изображения или видео, заголовка и объяснения контента.

В этих функциях используются различные приёмы и подходы для взаимодействия с пользователем, загрузки данных из API и их отображения на веб-странице. Некоторые из этих приёмов включают:

1. Асинхронные запросы: Использование ключевого слова async и функции await позволяет асинхронно загружать данные из API, чтобы не блокировать основной поток выполнения веб-страницы.
2. Обработчики событий: Установка обработчиков событий позволяет реагировать на действия пользователя, такие как клики на кнопки, изменения флажков и т.д.
3. Динамическое создание и изменение элементов HTML: Функции создают, изменяют и вставляют элементы HTML на веб-страницу в зависимости от полученных данных и действий пользователя.
4. Фильтрация и сортировка данных: Использование функций filterObjects() и sortObjects() из модуля utils.js позволяет фильтровать и сортировать массивы объектов на основе определенных критериев.
5. Модальные окна: Создание модального окна для отображения увеличенных фотографий Марса, а также обработка событий для его открытия и закрытия.

# Технологии

Функции - используются для выполнения определенных действий, например, получение данных об астероидах (fetchAsteroids()) и отображение списка астероидов на веб-странице (displayAsteroids()). Обработчики событий привязаны к определенным функциям для реагирования на действия пользователя (handleAsteroidsInteraction()).

Массивы - используются для хранения и манипуляции данными об астероидах, такими как исходные данные (this.asteroidsData), отфильтрованные астероиды (filteredAsteroids) и отсортированные астероиды (sortedAsteroids).

Классы – в коде присутствует несколько классов, один из них - класс AsteroidsUI, служит для организации функциональности, связанной с работой с астероидами, в одном месте. Он позволяет создать экземпляр класса для управления взаимодействием пользователя с астероидами на веб-странице.

Модули - используются для разделения кода на логически связанные модули, такие как api.js и utils.js, и импорта необходимых функций и данных из других модулей.

Асинхронность - в коде несколько примеров асинхронности, один из них – запросы к API.

Общий подход в коде состоит в том, чтобы разделить функциональность на отдельные функции и классы, чтобы облегчить чтение, понимание и сопровождение кода. Массивы используются для хранения и обработки данных, а модули позволяют импортировать и экспортировать функции и классы между файлами для организации кода. Асинхронность используется для работы с асинхронными операциями без блокировки основного потока выполнения.

# Демонстрация работы











