

UNIVERSITATEA DE STAT DIN MOLDOVA

Facultatea de Matematică și Informatică

Departamentul de Informatică

JavaScript

Отчет по индивидуальной работе №1

Тема “Анализ транзакций с использованием JavaScript”

Выполнил студент:

Smirnov Vladislav

Группа: IAFR2403

Проверил преподаватель:

asis. universitar Nartea Nichita

Кишинев

2025

Теоретическая часть

Формулировка задачи

Разработать консольное приложение на языке JavaScript для анализа транзакций.

Приложение должно включать:

1. Класс TransactionAnalyzer для обработки транзакций.
2. Методы анализа данных о транзакциях.
3. Загрузку данных из JSON-файла.

Описание цели и основные этапы работы

Целью работы является изучение основ работы с JavaScript, включая работу с классами, методами, JSON-файлами и функциями обработки массивов.

Основные этапы работы:

1. Создание структуры проекта.
2. Разработка классов Transaction и TransactionAnalyzer.
3. Реализация методов анализа транзакций.
4. Тестирование работы кода.

Практическая часть

Краткое описание особенностей реализации

1. Класс Transaction:

- a. Представляет отдельную транзакцию.
- b. Содержит метод `string()`, возвращающий строковое представление транзакции в формате JSON.

```
// класс представляет собой отдельную транзакцию и преобразует исходный
// объект (с полученной информацией) в экземпляр, у которого есть свойства и
// метод возвращать строковое представление транзакции в формате JSON
class Transaction {
    constructor({
        transaction_id,
        transaction_date,
        transaction_amount,
        transaction_type,
        transaction_description,
        merchant_name,
        card_type }) {
        this.transaction_id = transaction_id;
        this.transaction_date = transaction_date;
        this.transaction_amount = transaction_amount;
        this.transaction_type = transaction_type;
        this.transaction_description = transaction_description;
        this.merchant_name = merchant_name;
        this.card_type = card_type;
    }
    string() {
        return JSON.stringify(this, null, 2);
    }
}
```

1. Класс **TransactionAnalyzer**:

- a. Хранит массив транзакций.
- b. Реализует методы для анализа данных

Методы для анализа данных:

1. Получение списка всех транзакций (`getAllTransactions()`).
2. Фильтрация по типу (`getTransactionByType(type)`),
3. Фильтрация по диапазону дат (`getTransactionsInDateRange(startDate, endDate)`),
4. Фильтрация по сумме (`getTransactionsByAmountRange(minAmount, maxAmount)`).
5. Подсчет общей суммы (`calculateTotalAmount()`),
6. Подсчет средней суммы (`calculateAverageTransactionAmount()`),
7. Поиск наиболее частых типов (`mostTransactionTypes()`).
8. Анализ транзакций в разрезе месяцев (`findMostTransactionsMonth()`, `findMostDebitTransactionMonth()`).

Пример загрузки данных и создания экземпляра класса:

```
// Импортируем данные из файла 'transaction.json', который содержит массив
// объектов транзакций
const data = require('./transaction.json');

// Создаем новый экземпляр класса TransactionAnalyzer, передавая в
// конструктор массив данных из файла
const analyzerData = new TransactionAnalyzer(data);
```

Вывод

В ходе работы было создано консольное приложение на JavaScript, реализующее функциональность анализа транзакций с использованием классов и методов обработки массивов. Все методы, указанные в задании, были реализованы и протестированы.

Ссылка на репозиторий Git: <https://github.com/ofdoamne/JS>

Ответы на контрольные вопросы

1. **Какие примитивные типы данных существуют в JavaScript?**
 - string, number, boolean, null, undefined, symbol, bigint.
2. **Какие методы массивов использовались в работе?**
 - map(), filter(), reduce(), find(), Set(), join(), push().
3. **В чем состоит роль конструктора класса?**
 - Конструктор класса используется для создания и инициализации объектов с заданными свойствами.
4. **Как создать новый экземпляр класса в JavaScript?**
 - `const analyzer = new TransactionAnalyzer(data);`

Список использованных источников

1. MDN Web Docs: <https://developer.mozilla.org/>
2. JavaScript.info: <https://learn.javascript.ru/>
3. Официальная документация Node.js: <https://nodejs.org/>