Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Отчет по индивидуальной работе №1 по курсу “JavaScript & TypeScript”**

Проверил: Nartea Nichita

Отчет составил: Pitropov Alexandr

***Цель индивидуальной работы:***

Ознакомить студентов с основными функциями и с синтаксисом JavaScript на основе консольного приложения для анализа транзакций.

***Теоретическая часть:***

Задача данного проекта: предоставление определенных данных о транзакциях, которые запрашивает пользователь. Программа должна отображать данные о транзакциях, их сумму, дату их совершения, с какой карты была совершена транзакция и т.д. То есть, поиск конкретных транзакции.

***Описание цели и основные этапы работы:***

- Создание программы, которая поможет пользователю получить данные про определенную транзакцию, которая была совершена в какое-либо время.

* Создание классов, конструкторов классов с параметрами, а также использование методов.
* Тестирование работы программы и системы анализа транзакции.
* Вывод информации на экран и еще анализ.

***Краткое описание особенностей реализации:***

Дан файл transactions.json, содержащий некоторое кол-во транзакции. Целью задания было создать программу, которая с помощью определенных методов обрабатывает данный json файл и выдает запрошенную информацию на экран. Я ориентировался по заданиям, которые были заданы.

Этот код состоит из двух классов: `Transaction` и `TransactionAnalyzer`, которые представляют финансовые транзакции и обеспечивают их анализ соответственно.

***Класс Transaction:***

Описание: Представляет отдельную финансовую транзакцию, включающую уникальный идентификатор, дату, сумму, тип (например, дебетовая или кредитная), описание, информацию о торговце и типе карты.

***Конструктор:***

Принимает параметры, такие как идентификатор, дата, сумма, тип, описание, торговец и тип карты, и инициализирует соответствующие свойства объекта.

***Метод toString():***

Возвращает строковое представление транзакции в формате JSON, используя метод JSON.stringify() для преобразования свойств объекта в строку JSON.

***Класс TransactionAnalyzer:***

Описание:

Предоставляет функциональность для анализа и обработки наборов финансовых транзакций, включая методы для добавления новой транзакции, получения всех транзакций, вычисления общей суммы транзакций, поиска транзакций по различным параметрам и вычисления статистики.

***Конструктор:***

Принимает массив исходных данных транзакций, создает экземпляры класса Transaction на основе данных из массива и инициализирует свойство transactions массивом этих транзакций.

***Основные методы:***

Включают методы для добавления транзакции, получения всех транзакций, получения уникальных типов транзакций, вычисления общей суммы транзакций, получения транзакций по типу и торговцу, вычисления средней суммы транзакции и поиска транзакции по идентификатору.

***Дополнительные методы:***

В классе также присутствуют методы для выполнения различных операций с данными транзакций, такие как фильтрация по дате или сумме, вычисление статистики по типам транзакций и т. д.

***Пример использования:***

1.Создание экземпляра TransactionAnalyzer на основе данных о транзакциях.

2.Использование методов для анализа и обработки этих данных, таких как получение общей суммы транзакций, поиск транзакции по идентификатору, фильтрация по типу или торговцу и т. д.

***Вывод:***

Проведенная работа позволила мне улучшить навыки в использовании новых элементов языка JavaScript, а представленный код стал отличным примером работы с транзакциями и их анализом. Такой подход может стать основой для разработки более сложных приложений для анализа банковских транзакций на JavaScript.

***Ответы на контрольные вопросы:***

1. **Какие примитивные типы используются в языке**

**JavaScript?**

Ответ: String, Number, Boolean, Undefined, Null, Symbol.

1. **Какие методы массивов вы использовали для обработки и анализа данных в вашем приложении, и как они помогли в выполнении задачи?**

Ответ: В приложении были использованы следующие методы массивов для обработки и анализа данных:

**Метод map()** был использован для преобразования исходного массива данных о транзакциях в массив объектов класса Transaction. Каждый элемент исходного массива был отображен на экземпляр класса Transaction, что позволило легко создать новый массив объектов транзакций при помощи указанной функции-преобразователя.

**Метод filter()** применялся для фильтрации массива транзакций по различным критериям, таким как тип транзакции или имя торговца. Например, это позволило легко найти все транзакции определенного типа или транзакции от определенного торговца, просто применив метод filter() к массиву транзакций.

**Применение метода reduce()** позволило вычислить общую сумму всех транзакций или сумму всех дебетовых транзакций. Этот метод позволяет итерироваться по массиву и выполнить агрегирующее действие, например, суммирование значений, на основе всех элементов массива.

Такое использование методов позволило мне эффективно обрабатывать и анализировать данные о транзакциях, выполняя различные операции, такие как преобразование данных, фильтрация по критериям и вычисление агрегированных значений.

**3. В чем состоит роль конструктора класса?**

Ответ: Конструктор класса в JavaScript используется для инициализации объектов, создаваемых этим классом. Он выполняет следующие функции:

- Инициализация свойств: Конструктор устанавливает начальные значения свойств объекта, используя переданные параметры. - Создание нового объекта: При вызове конструктора с ключевым словом new создается новый экземпляр объекта, который затем можно использовать в приложении.

**4. Каким образом вы можете создать новый экземпляр класса в JavaScript?**

Ответ: Новый экземпляр класса можно создать при помощи оператора new. В моем случае, оператор был использован так: new Transaction (...), который передает определенные аргументы конструктору класса. Экземпляр сохраняется в переменной transaction для последующего использования.

***Список использованных источников:***

* Официальная документация MDN.
* Обучающие материалы на YouTube.
* Интернет-форумы и поиск в Google.
* Учебный материал по JavaScript на платформе Moodle.

***Ссылка на репозиторий GitHub:***

https://github.com/alexunderpitropov/individualka