Молдавский Государственный Университет
Факультет Математики и Информатики
Департамент Информатики

Отчет по индивидуальной работе №1 по курсу JavaScript & TypeScript

Проверил: Nartea Nichita
Отчет составил: студент 1-го курса
группы I2302 Slavov Constantin

Кишинев, 2024

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:
Этот класс представляет отдельную финансовую транзакцию. Он содержит информацию о ее уникальном идентификаторе, дате, сумме, типе (например, дебетовая или кредитная), описании, торговце и типе карты.

Конструктор:
- Принимает параметры, такие как идентификатор, дата, сумма, тип, описание, торговец и тип карты.
- Инициализирует соответствующие свойства объекта с полученными значениями.

- Метод `toString()`:
- Возвращает строковое представление транзакции в формате JSON.
- Использует метод `JSON.stringify()` для преобразования свойств объекта в строку JSON.

- Класс `TransactionAnalyzer`:
Описание:
Этот класс предоставляет функциональность для анализа и обработки наборов финансовых транзакций. Он содержит методы для работы с данными транзакций, такие как добавление новой транзакции, получение всех транзакций, вычисление общей суммы транзакций, поиск транзакций по различным параметрам и вычисление 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:

1. `addTransaction(transactionData)`: Добавляет новую транзакцию в список транзакций.
2. `getAllTransactions()`: Возвращает все транзакции.
3. `getUniqueTransactionTypes()`: Возвращает массив уникальных типов транзакций.
4. `calculateTotalAmount()`: Вычисляет общую сумму транзакций.
5.`getTransactionByType(type)`: Возвращает транзакции указанного типа.
6.`getTransactionsByMerchant(merchantName)`: Возвращает транзакции указанного торговца.
7.`calculateAverageTransactionAmount()`: Вычисляет среднюю сумму транзакции.
8.`findTransactionById(id)`: Находит транзакцию по ее 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, например, получение общей суммы транзакций, поиск транзакции по идентификатору, фильтрация по типу или торговцу и т. д.

Вывод:

Исходя из проделанной работы, я смог приобрести навыки с новыми элементами языка JavaScript, а данный код является хорошим примером работы с транзакциями и их анализа. Данный подход может служить неплохой основой для создания более сложных приложений для аналитики банковских транзакции на языке JavaScript.

Ссылка на репозиторий GitHub:

https://github.com/kraaddys/JS_and_TS/tree/main/myLab

Ответы на контрольные вопросы:

1. Какие примитивные типы используются в языке JavaScript?

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

2. 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:

1. `map()`: Этот метод был использован для преобразования массива исходных данных транзакций в массив объектов класса `Transaction`. Каждый элемент исходного массива был отображен на экземпляр класса `Transaction`, что позволило легко создать новый массив объектов транзакций с помощью указанной функции-преобразователя.

2. `filter()`: Метод `filter()` был использован для фильтрации массива транзакций по определенным критериям, таким как тип транзакции или имя торговца. Например, можно было легко найти все транзакции заданного типа или транзакции от определенного торговца, просто применив метод `filter()` к массиву транзакций.

3. `reduce()`: Метод `reduce()` использовался для вычисления общей суммы транзакций или общей суммы дебетовых транзакций. Этот метод позволяет выполнить итерацию по массиву и выполнить какое-либо агрегирующее действие, например, суммирование значений, на основе всех элементов массива.

Эти методы позволили 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:

- Chat GPT
- MDN
- YouTube
- Просто Google и форумы
- Современный учебник JavaScript, который был закреплен на Moodle