Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Дисциплина:** JavaScript

**Отчёт**

**по лабораторной работе №3**

**Тема:** Продвинутые обьекты в JavaScript

Выполнил: студент группы IA2403

Демченко Юрий

Проверил: преподаватель

A.Murineț

Кишинев, 2025

**Лабораторная работа №3**

**Цель работы:**

Познакомиться с классами и объектами в JavaScript, научиться создавать классы, использовать конструкторы и методы, а также реализовать наследование.

**Условия работы:**

Шаг 1. Создание класса Item

Шаг 2. Создание класса Weapon

Шаг 3. Тестирование

Шаг 4. Создание функции-конструктора

Документирование кода

**Ход работы:**

В рамках лабораторной работы я создал два файла с реализацией классов на языке JavaScript:

lab3.js — в этом файле была реализована структура классов с использованием классов.

lab3v2.js — аналогичная структура, но с использованием функций-конструкторов.

1.Создание класса Item

В первом шаге я создал класс Item, который представляет предмет в игре. В конструкторе класса определяются три свойства:

name — название предмета;

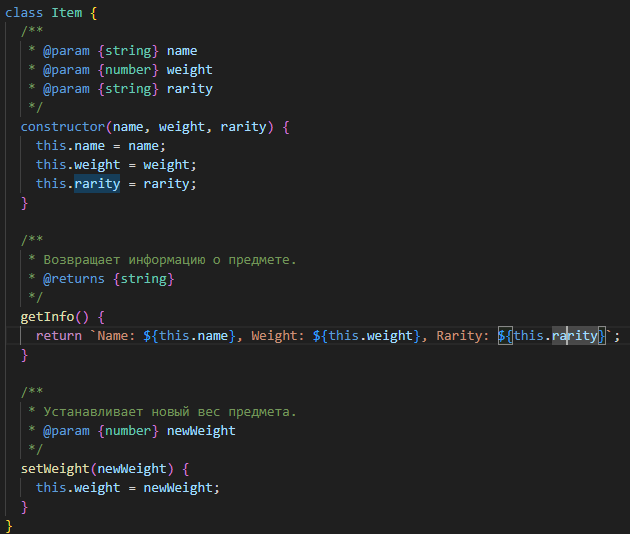
weight — вес предмета;

rarity — редкость предмета (например, обычный, редкий, легендарный).

Также были реализованы два метода:

getInfo — возвращает строковое представление информации о предмете.

setWeight — позволяет изменить вес предмета.



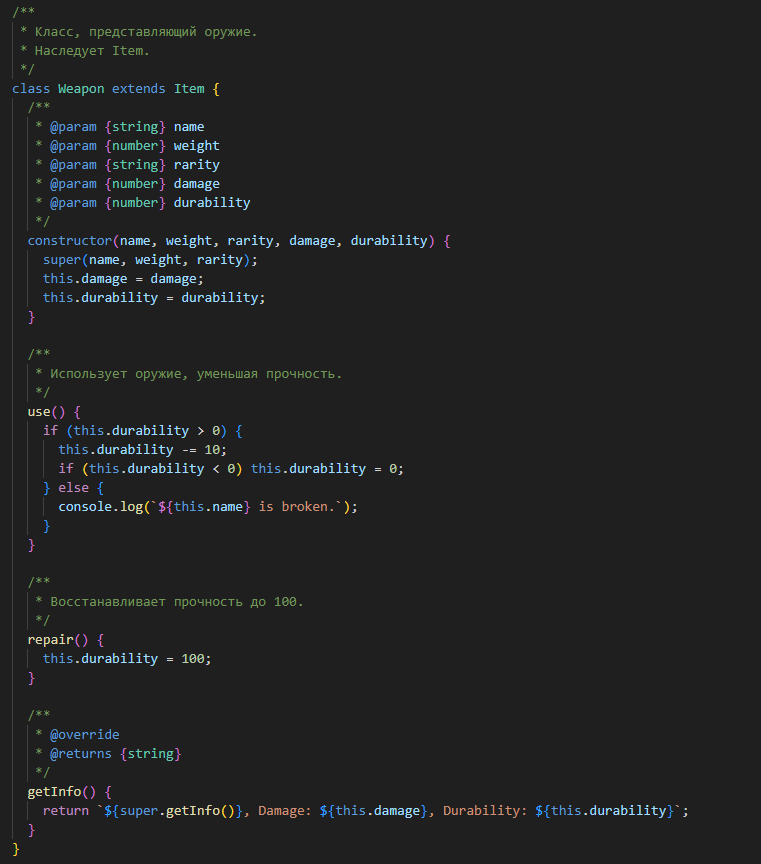
2. Создание класса Weapon

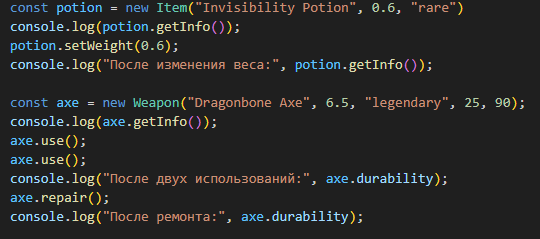
Затем я создал класс Weapon, который наследуется от Item и добавляет дополнительные характеристики, такие как damage (урон) и durability (прочность). Для оружия были реализованы два метода:

use —уменьшает прочность оружия при использовании.

repair — восстанавливает прочность оружия до максимума.

Кроме того, я переопределил метод getInfo, чтобы добавить информацию об уроне и прочности.

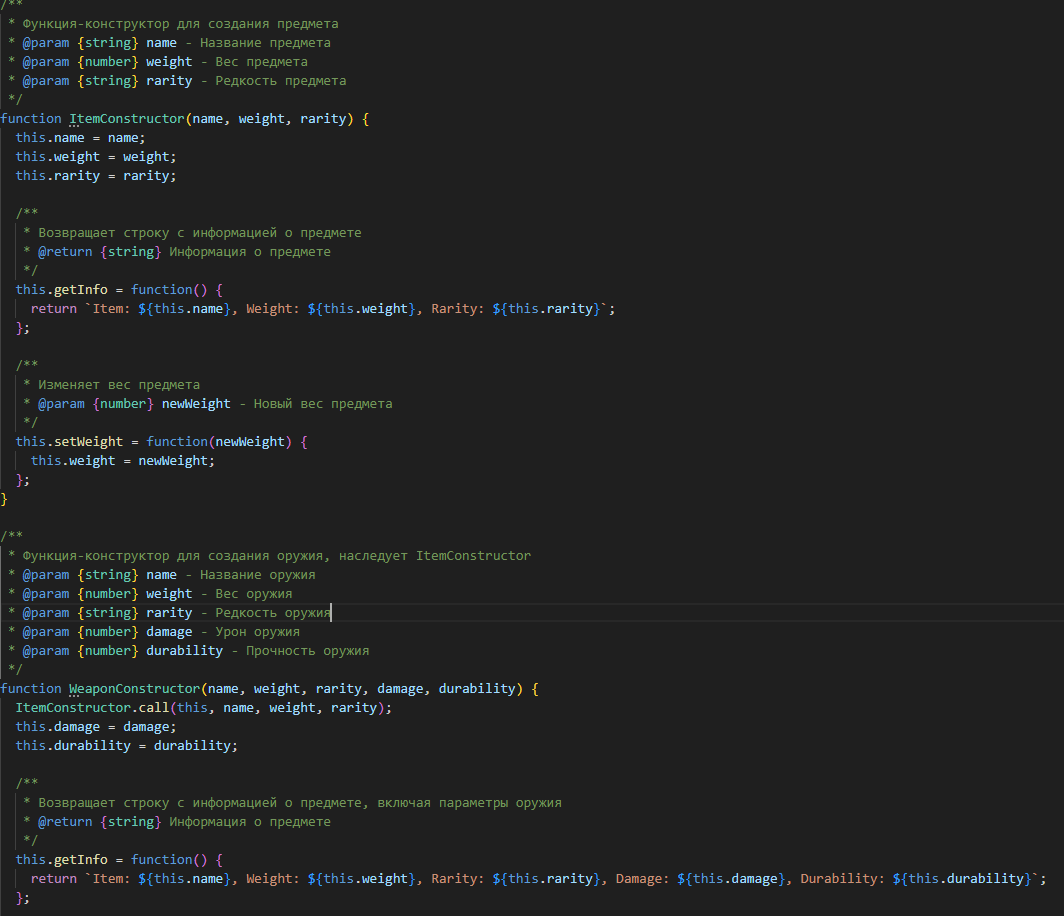




3. Аналогичную функциональность я реализовал во втором файле, но с использованием функций-конструкторов. В этом случае я определил два конструктора:

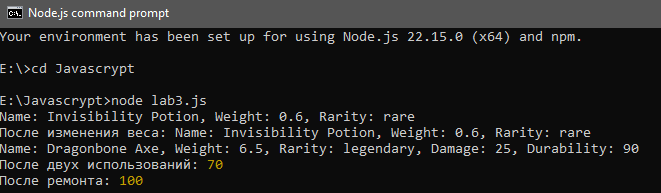
ItemConstructor — для создания простых предметов.

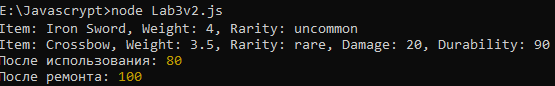
WeaponConstructor — для создания оружия, который наследует свойства и методы от ItemConstructor.





Результат:





**Выводы**

В ходе выполнения лабораторной работы №3, я научился использовать объекты и методы для моделирования реальных сущностей, таких как предметы и оружие, с их характеристиками и функциональностью. Реализация различных методов для изменения состояния объектов позволила мне на практике понять, как эффективно управлять состоянием объектов и работать с их поведением.

**Ответы на контрольные вопросы**

1. Какое значение имеет this в методах класса?

Ключевое слово this в методах класса ссылается на текущий экземпляр объекта, который был создан с помощью оператора new. Это позволяет методам обращаться к другим свойствам и методам этого объекта. Например, в методе getInfo() класса Item, выражение this.name указывает на поле name именно того объекта, для которого вызван метод.

1. Как работает модификатор доступа # в JavaScript?

Модификатор # используется для объявления приватных полей и методов. Такие свойства не могут быть доступны или изменены вне класса. Это повышает инкапсуляцию и безопасность кода. Например, если объявить #durability в классе Weapon, то получить к нему доступ напрямую (например, через object.#durability) снаружи будет невозможно — возникнет ошибка.

1. В чём разница между классами и функциями-конструкторами?

Классы (class) — это синтаксическая надстройка над прототипным наследованием в JavaScript, введённая в ES6. Они обеспечивают более читаемый и современный способ определения объектов и наследования. Функции-конструкторы использовались до появления классов и требуют явной работы с прототипами.

Основные различия:

-Синтаксис классов проще и чище.

-Классы поддерживают ключевые слова extends и super() для наследования.

-В классах можно использовать приватные поля (#).

-Функции-конструкторы используют Function.prototype для добавления методов.

**Библиография**

1. <https://github.com/MSU-Courses/javascript>
2. <https://moodle.usm.md/course/JavaScript>