Лабораторная работа № 4

Тема: Прототипы и прототипное наследование.

Цель: Освоить работу со свойством prototype, цепочкой прототипов, методами объектов и принципами прототипного наследования в JavaScript.

🔷 Вариант 1.

1. Создайте функцию-конструктор Animal, которая принимает один параметр: species.

Добавьте метод speak в прототип Animal, который выводит сообщение в формате: "The [species] makes a sound.".

Затем создайте экземпляр Animal и вызовите метод speak.

В консоли браузера введите имя созданного экземпляра и посмотрите цепочку свойств.

2. Создайте функцию-конструктор Vehicle, которая принимает параметр brand.

Добавьте метод start в прототип Vehicle, который выводит сообщение в формате: "The [brand] vehicle is starting.".

Затем создайте функцию-конструктор Car, которая наследует от Vehicle и имеет свойство model.

Переопределите метод start в Car, чтобы он выводил сообщение в формате: "The [brand] car model [model] is starting.".

Создайте экземпляр Car и вызовите метод start.

🔷 Вариант 2.

1. Создайте функцию-конструктор Book, которая принимает два параметра: title и author.

Добавьте метод getInfo в прототип Book, который возвращает строку с информацией о книге в формате: "Title: [title], Author: [author]".

Затем создайте экземпляр Book и вызовите метод getInfo.

В консоли браузера введите имя созданного экземпляра и посмотрите цепочку свойств.

2. Создайте функцию-конструктор Shape, которая не имеет параметров.

Добавьте метод area, который возвращает undefined. Затем создайте функцию-конструктор Rectangle, которая наследует от Shape и имеет свойства width и height.

Переопределите метод area в Rectangle, чтобы он возвращал площадь прямоугольника (ширина * высота).

Создайте экземпляр Rectangle и выведите его площадь.

⚡ **Отчет должен содержать (см. образец):**
- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна VSC с исходным кодом программ;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email:
colledge20education23@gmail.com

Шпаргалка по прототипам.
1. Создание функции-конструктора.
Используйте функции для создания объектов.
При создании экземпляров используйте ключевое слово new.
Пример:

```
function Fruit(name, color) {
    this.name = name;  // Свойство name
    this.color = color; // Свойство color
}
```

2. Добавление методов в прототип
Методы, которые должны быть доступны всем экземплярам, добавляются в прототип конструктора.
Пример:

```
Fruit.prototype.describe = function() {
    console.log(`This is a ${this.color} ${this.name}.`);
};
```

3. Создание экземпляра
Создайте экземпляр с помощью new и вызовите методы.
Пример:

```
const apple = new Fruit('Apple', 'red');
apple.describe(); // "This is a red Apple."
```

4. Наследование через прототип
При наследовании вызовите конструктор родителя с параметрами.
Пример:

```
function Citrus(name, color, acidity) {
    Fruit.call(this, name, color); // Вызов конструктора
родителя
    this.acidity = acidity;         // Свойство acidity
}

// Устанавливаем прототип Citrus на Fruit
Citrus.prototype = Object.create(Fruit.prototype);
Citrus.prototype.constructor = Citrus;
```

5. Переопределение методов
Переопределите методы в дочернем конструкторе, чтобы изменить их поведение.
Пример:

```
Citrus.prototype.describe = function() {
    console.log(`This is a ${this.color} ${this.name} with
an acidity level of ${this.acidity}.`);
};
```

6. Создание экземпляра дочернего конструктора
Создайте экземпляр дочернего конструктора и вызывайте переопределенные методы.
Пример:

```
const lemon = new Citrus('Lemon', 'yellow', 'high');
lemon.describe(); // "This is a yellow Lemon with an acidity
level of high."
```