**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине: «Современные технологии создания Web-систем»

на тему: **«**Работа с объектами»

Выполнил: студент гр. ИТИ-21

Манько С.Д.

Принял: ассистент

Карась О.В.

Гомель 2025

**Цель работы:** получить навыки создания объектов, доступа к полям и методам, ознакомится с прототипным наследованием.

**Краткие теоретические сведения**

На сегодняшний день объектно-ориентированное программирование (ООП) является одной из господствующих парадигм в разработке приложений. JavaScript предоставляет возможности ООП, но имеет некоторые особенности.

Основным понятием ООП является объект – сложная комплексная структура, объединяющая в себе данные и методы их обработки.

Для работы с подобными структурами в JavaScript используются объекты. Каждый объект может хранить свойства, которые описывают его состояние, и методы, которые описывают его поведение.

**Порядок выполнения работы**

Задание **1**. В соответствии со своим вариантом определить объект, с заданными полями и методом вывода информации об объекте. Объект создается с помощью объектного литерала.

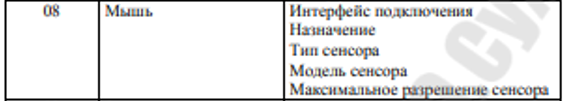


Рисунок 1 – Выбранный вариант задания.

Задание **2.** Написать функцию-конструктор для создания объектов в соответствии с вариантом. Определить геттеры и сеттеры для получения доступа к свойствам. В прототип объектов добавить свойство – дата выхода на рынок, метод отображения информации об объекте.

Задание **3.** Описать класс, описывающий объекты в соответствии с вариантом. Предусмотреть конструктор с параметрами, геттеры и сеттеры для получения доступа к свойствам, метод отображения информации об объекте.

Задание **4.** Описать класс, являющийся наследником класса задания 3. В производном классе добавить свойства: дата выхода на рынок, стоимость. Переопределить метод вывода информации об объекте.

Задание **5.** На основе разработанного класса задания 4 создать массив объектов. Определить объекты с максимальной и минимальной стоимостью, суммарную стоимость всех объектов, вычислить среднюю стоимость объекта, подсчитать количество объектов со стоимостью выше средней. Для выполнения задания использовать встроенные методы массивов и объект Math.

**Ход работы**

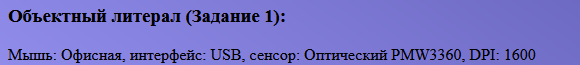


Рисунок 2 –Выполнение задания 1.

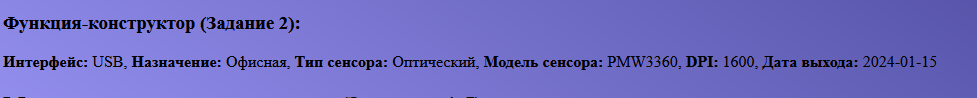


Рисунок 2 –Выполнение задания 2.

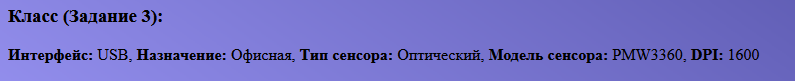


Рисунок 2 –Выполнение задания 3.

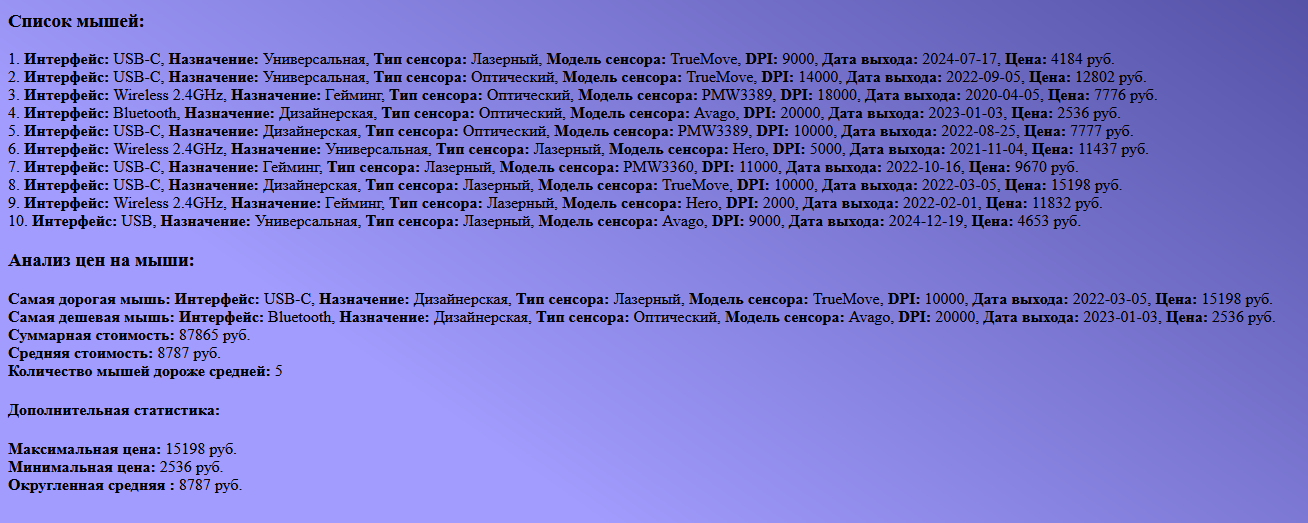


Рисунок 3 –Выполнение задания 4 и 5.

**Выводы:** На лабораторной работе получены навыки создания объектов, доступа к полям и методам, ознакомится с прототипным наследованием.

**ПРИЛОЖЕНИЕ А**

**Файл 1.js**

function createMouseLiteral() {

const mouse = {

connectionInterface: document.getElementById('interfaceInput').value || "USB",

purpose: document.getElementById('purposeInput').value || "Офисная",

sensorType: document.getElementById('sensorTypeInput').value || "Оптический",

sensorModel: document.getElementById('sensorModelInput').value || "Standard",

maxDPI: parseInt(document.getElementById('maxDPIInput').value) || 1600,

displayInfo: function() {

return `Мышь: ${this.purpose}, интерфейс: ${this.connectionInterface}, сенсор: ${this.sensorType} ${this.sensorModel}, DPI: ${this.maxDPI}`;

}

};

return mouse;

}

function MouseConstructor(connectionInterface, purpose, sensorType, sensorModel, maxDPI) {

this.\_connectionInterface = connectionInterface;

this.\_purpose = purpose;

this.\_sensorType = sensorType;

this.\_sensorModel = sensorModel;

this.\_maxDPI = maxDPI;

}

MouseConstructor.prototype.getConnectionInterface = function() {

return this.\_connectionInterface;

};

MouseConstructor.prototype.setConnectionInterface = function(value) {

this.\_connectionInterface = value;

};

MouseConstructor.prototype.getPurpose = function() {

return this.\_purpose;

};

MouseConstructor.prototype.setPurpose = function(value) {

this.\_purpose = value;

};

MouseConstructor.prototype.getSensorType = function() {

return this.\_sensorType;

};

MouseConstructor.prototype.setSensorType = function(value) {

this.\_sensorType = value;

};

MouseConstructor.prototype.getSensorModel = function() {

return this.\_sensorModel;

};

MouseConstructor.prototype.setSensorModel = function(value) {

this.\_sensorModel = value;

};

MouseConstructor.prototype.getMaxDPI = function() {

return this.\_maxDPI;

};

MouseConstructor.prototype.setMaxDPI = function(value) {

if (value > 0) this.\_maxDPI = value;

};

MouseConstructor.prototype.releaseDate = "Не указана";

MouseConstructor.prototype.displayInfo = function() {

return `<b>Интерфейс:</b> ${this.getConnectionInterface()}, <b>Назначение:</b> ${this.getPurpose()}, <b>Тип сенсора:</b> ${this.getSensorType()}, <b>Модель сенсора:</b> ${this.getSensorModel()}, <b>DPI:</b> ${this.getMaxDPI()}, <b>Дата выхода:</b> ${this.releaseDate}`;

};

class Mouse {

constructor(connectionInterface, purpose, sensorType, sensorModel, maxDPI) {

this.\_connectionInterface = connectionInterface;

this.\_purpose = purpose;

this.\_sensorType = sensorType;

this.\_sensorModel = sensorModel;

this.\_maxDPI = maxDPI;

}

get connectionInterface() { return this.\_connectionInterface; }

get purpose() { return this.\_purpose; }

get sensorType() { return this.\_sensorType; }

get sensorModel() { return this.\_sensorModel; }

get maxDPI() { return this.\_maxDPI; }

set connectionInterface(value) { this.\_connectionInterface = value; }

set purpose(value) { this.\_purpose = value; }

set sensorType(value) { this.\_sensorType = value; }

set sensorModel(value) { this.\_sensorModel = value; }

set maxDPI(value) { if (value > 0) this.\_maxDPI = value; }

displayInfo() {

return `<b>Интерфейс:</b> ${this.connectionInterface}, <b>Назначение:</b> ${this.purpose}, <b>Тип сенсора:</b> ${this.sensorType}, <b>Модель сенсора:</b> ${this.sensorModel}, <b>DPI:</b> ${this.maxDPI}`;

}

}

class MouseCommercial extends Mouse {

constructor(connectionInterface, purpose, sensorType, sensorModel, maxDPI, date, price) {

super(connectionInterface, purpose, sensorType, sensorModel, maxDPI);

this.\_date = date;

this.\_price = price;

}

get date() { return this.\_date; }

get price() { return this.\_price; }

set date(value) { this.\_date = value; }

set price(value) { if (value > 0) this.\_price = value; }

getMouseInfo() {

return `<b>Интерфейс:</b> ${this.connectionInterface}, <b>Назначение:</b> ${this.purpose}, <b>Тип сенсора:</b> ${this.sensorType}, <b>Модель сенсора:</b> ${this.sensorModel}, <b>DPI:</b> ${this.maxDPI}, <b>Дата выхода:</b> ${this.date}, <b>Цена:</b> ${this.price} руб.`;

}

}

const connectionInterfaces = ["USB", "Bluetooth", "USB-C", "Wireless 2.4GHz"];

const purposes = ["Офисная", "Гейминг", "Дизайнерская", "Универсальная"];

const sensorTypes = ["Оптический", "Лазерный", "Оптико-лазерный"];

const sensorModels = ["PMW3360", "PMW3389", "Hero", "TrueMove", "Avago"];

let mice = [];

function createRandomMouse() {

const connectionInterface = connectionInterfaces[Math.floor(Math.random() \* connectionInterfaces.length)];

const purpose = purposes[Math.floor(Math.random() \* purposes.length)];

const sensorType = sensorTypes[Math.floor(Math.random() \* sensorTypes.length)];

const sensorModel = sensorModels[Math.floor(Math.random() \* sensorModels.length)];

const maxDPI = Math.floor(Math.random() \* 20) \* 1000 + 1000;

const date = `${Math.floor(Math.random() \* 5) + 2020}-${(1 + Math.floor(Math.random() \* 12)).toString().padStart(2, '0')}-${(Math.floor(Math.random() \* 28) + 1).toString().padStart(2, '0')}`;

const price = Math.floor(Math.random() \* 20000) + 1000;

return new MouseCommercial(connectionInterface, purpose, sensorType, sensorModel, maxDPI, date, price);

}

function initializeMice() {

mice = [];

for(let i = 0; i < 10; i++) {

mice.push(createRandomMouse());

}

}

function printMice() {

const miceElement = document.getElementById("mice");

miceElement.innerHTML = "<h3>Список мышей:</h3>";

mice.forEach((mouse, index) => {

miceElement.innerHTML += `<div>${index + 1}. ${mouse.getMouseInfo()}</div>`;

});

}

function checkMice() {

if (mice.length === 0) {

initializeMice();

}

const prices = mice.map(mouse => mouse.price);

const maxPrice = Math.max(...prices);

const minPrice = Math.min(...prices);

const sumPrice = prices.reduce((sum, price) => sum + price, 0);

const avgPrice = sumPrice / mice.length;

const maxPriceMouse = mice.find(mouse => mouse.price === maxPrice);

const minPriceMouse = mice.find(mouse => mouse.price === minPrice);

const countAboveAvg = mice.filter(mouse => mouse.price > avgPrice).length;

const miceInfoElement = document.getElementById("miceInfo");

miceInfoElement.innerHTML = `

<h3>Анализ цен на мыши:</h3>

<div><b>Самая дорогая мышь:</b> ${maxPriceMouse.getMouseInfo()}</div>

<div><b>Самая дешевая мышь:</b> ${minPriceMouse.getMouseInfo()}</div>

<div><b>Суммарная стоимость:</b> ${sumPrice} руб.</div>

<div><b>Средняя стоимость:</b> ${Math.round(avgPrice)} руб.</div>

<div><b>Количество мышей дороже средней:</b> ${countAboveAvg}</div>

<h4>Дополнительная статистика:</h4>

<div><b>Максимальная цена:</b> ${maxPrice} руб.</div>

<div><b>Минимальная цена:</b> ${minPrice} руб.</div>

<div><b>Округленная средняя :</b> ${Math.round(avgPrice)} руб.</div>

`;

}

function demonstrateLiteral() {

const mouse = createMouseLiteral();

document.getElementById("demoResult").innerHTML = `

<h3>Объектный литерал (Задание 1):</h3>

<div>${mouse.displayInfo()}</div>

`;

}

function demonstrateConstructor() {

const mouse = new MouseConstructor(

document.getElementById('interfaceInput').value || "USB",

document.getElementById('purposeInput').value || "Офисная",

document.getElementById('sensorTypeInput').value || "Оптический",

document.getElementById('sensorModelInput').value || "Standard",

parseInt(document.getElementById('maxDPIInput').value) || 1600

);

mouse.releaseDate = "2024-01-15";

document.getElementById("demoResult").innerHTML = `

<h3>Функция-конструктор (Задание 2):</h3>

<div>${mouse.displayInfo()}</div>

`;

}

function demonstrateClass() {

const mouse = new Mouse(

document.getElementById('interfaceInput').value || "USB",

document.getElementById('purposeInput').value || "Офисная",

document.getElementById('sensorTypeInput').value || "Оптический",

document.getElementById('sensorModelInput').value || "Standard",

parseInt(document.getElementById('maxDPIInput').value) || 1600

);

document.getElementById("demoResult").innerHTML = `

<h3>Класс (Задание 3):</h3>

<div>${mouse.displayInfo()}</div>

`;

}

document.addEventListener('DOMContentLoaded', function() {

initializeMice();

printMice();

checkMice();

});