МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

по дисциплине «Разработка приложений для Интернет»

Работа с объектами

Выполнил студент

группы ИП-31

Заяц Н. С.

Проверил преподаватель

Самовендюк Н. В.

Гомель 2021

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

**Вариант 12**

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

| 08 | Мышь | Интерфейс подключения  Назначение  Тип сенсора  Модель сенсора  Максимальное разрешение сенсора |
| --- | --- | --- |

**Решение:**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Zaysts MIkita</title>

</head>

<body>

<script>

function print(...args) {

for (let a of args) {

document.write('<div>');

document.write(a);

document.write('<br></div>');

}

}

function pushResult(results, result, title) {

results.push({

result: result,

title: title,

toString() {

return 'TITLE: ' + title + '<br> RESULT: ' + result;

}

});

}

let elements = [];

let mouseOfNikita = {

sensorModel: 'GF-312',

sensorType: 'Lizer',

title: 'Nikita\'s mouse',

connectionInterface: 'USB 3.9',

dPI: 400,

toString() {

return JSON.stringify(this, null, '<br>');

}

}

pushResult(elements, mouseOfNikita, 'mouse');

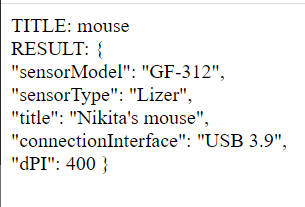
print(...elements);

</script>

</body>

</html>

**Результат:**

****

**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.

**Решение:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Zayats</title>

</head>

<body>

<script>

function Mouse(sensorModel, sensorType, title, connectionInterface, dPI) {

this.\_sensorModel = sensorModel;

this.\_sensorType = sensorType;

this.\_title = title;

this.\_connectionInterface = connectionInterface;

this.\_dPI = dPI;

}

Mouse.prototype = {

get sensorModel() {

return this.\_sensorModel;

},

get sensorType() {

return this.\_sensorType;

},

get title() {

return this.\_title;

},

get connectionInterface() {

return this.\_connectionInterface;

},

get dPI() {

return this.\_dPI;

},

// setters

set sensorModel(sensorModel) {

this.\_sensorModel = sensorModel;

},

set sensorType(sensorType) {

this.\_sensorType = sensorType;

},

set title(title) {

this.\_title = title;

},

set connectionInterface(connectionInterface) {

this.\_connectionInterface = connectionInterface;

},

set dPI(dPI) {

this.\_dPI = dPI;

},

}

Mouse.prototype.realeseDate = '1 октября 2018г';

Mouse.prototype.logInformation = function () {

console.log(`sensorModel: ${this.sensorModel}\nsensorType: ${this.sensorType}\ntitle: ${this.title}`,

`\nconnectionInterface: ${this.connectionInterface}\nДата выхода на рынок: ${this.realeseDate}\ndPI: ${this.dPI}.`)

}

let mouse = new Mouse('МГТ-321', 'Лазер', 'Мышка никиты - 2', 'USB', 321);

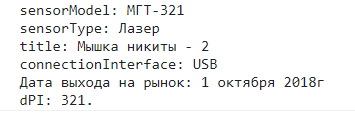
mouse.logInformation();

</script>

</body>

</html>

**Результат:**



**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.

**Решение:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Zayats</title>

</head>

<body>

<script>

class Mouse {

#\_sensorModel;

#\_sensorType;

#\_title;

#\_connectionInterface;

#\_dPI;

constructor(...args) {

if (args[0] instanceof Mouse) {

let mouse = args[0];

this.#\_sensorModel = mouse.sensorModel;

this.#\_sensorType = mouse.sensorType;

this.#\_title = mouse.title;

this.#\_connectionInterface = mouse.connectionInterface;

this.#\_dPI = mouse.dPI;

} else {

this.#\_sensorModel = args[0];

this.#\_sensorType = args[1];

this.#\_title = args[2];

this.#\_connectionInterface = args[3];

this.#\_dPI = args[4];

}

}

// getters

get sensorModel() {

return this.#\_sensorModel;

}

get sensorType() {

return this.#\_sensorType;

}

get title() {

return this.#\_title;

}

get connectionInterface() {

return this.#\_connectionInterface;

}

get dPI() {

return this.#\_dPI;

}

// setters

set sensorModel(sensorModel) {

this.#\_sensorModel = sensorModel;

}

set sensorType(sensorType) {

this.#\_sensorType = sensorType;

}

set title(title) {

this.#\_title = title;

}

set connectionInterface(connectionInterface) {

this.#\_connectionInterface = connectionInterface;

}

set dPI(dPI) {

this.#\_dPI = dPI;

}

toString() {

return JSON.stringify(this);

}

printInConsole(){

console.log('Mouse : ');

console.log(this.toString());

}

};

let mouse = new Mouse('GX-123', 'Лазер', 'Мышь никиты игровая', 'USB', 540);

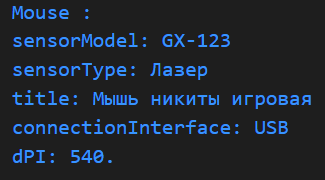
mouse.printInConsole();

</script>

</body>

</html>

**Результат:**



**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.

**Решение:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Task\_2</title>

</head>

<body>

<script>

function print(...args) {

for (let a of args) {

document.write('<div>');

document.write(a);

document.write('<br></div>');

}

}

class Mouse {

#\_sensorModel;

#\_sensorType;

#\_title;

#\_connectionInterface;

#\_dPI;

constructor(options) {

this.#\_sensorModel = options.sensorModel;

this.#\_sensorType = options.sensorType;

this.#\_title = options.title;

this.#\_connectionInterface = options.connectionInterface;

this.#\_dPI = options.dPI;

}

// getters

get sensorModel() {

return this.#\_sensorModel;

}

get sensorType() {

return this.#\_sensorType;

}

get title() {

return this.#\_title;

}

get connectionInterface() {

return this.#\_connectionInterface;

}

get dPI() {

return this.#\_dPI;

}

// setters

set sensorModel(sensorModel) {

this.#\_sensorModel = sensorModel;

}

set sensorType(sensorType) {

this.#\_sensorType = sensorType;

}

set title(title) {

this.#\_title = title;

}

set connectionInterface(connectionInterface) {

this.#\_connectionInterface = connectionInterface;

}

set dPI(dPI) {

this.#\_dPI = dPI;

}

toString() {

return `sensorModel: ${this.sensorModel}.\ndPI: ${this.dPI}.\nconnectionInterface: ${this.connectionInterface}\n.sensorType: ${this.sensorType}\n.`;

}

print(){

console.log(this.toString());

}

};

class MouseWithPrinter extends Mouse{

constructor(options){

super(options);

this.price = options.price;

this.relizeDate = options.relizeDate;

}

print(){

console.log('MouseWithPrinter:');

console.log(this.toString());

}

toString() {

let parrentData = super.toString();

let currentData = `$price: ${this.price}; relizeDate: ${this.relizeDate};`;

return parrentData + currentData;

}

}

let mouseWithPrinter1 = new MouseWithPrinter({

sensorModel: 'IP-213',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 123,

relizeDate: 'todat',

});

let mouseWithPrinter2 = new MouseWithPrinter({

sensorModel: 'IP-213',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 576,

relizeDate: 'todat',

});

let mouseWithPrinter3 = new MouseWithPrinter({

sensorModel: 'IP-213',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 22,

relizeDate: 'todat',

});

let mouses = [mouseWithPrinter1, mouseWithPrinter2, mouseWithPrinter3];

console.log('Индекс объекта с минимальной стоимостью: ' + mouses.map(e => e.price).indexOf(Math.min(...mouses.map(e => e.price))))

console.log('Индекс объекта с максимальной стоимостью: ' + mouses.map(e => e.price).indexOf(Math.max(...mouses.map(e => e.price))))

console.log('Суммарная стоимость всех объектов: ' + mouses.map(e => e.price).reduce((acc, curr) => acc + curr) + '$')

console.log('Средняя стоимость объекта: ' + mouses.map(e => e.price).reduce((acc, curr) => acc + curr) / mouses.length + '$')

let averageCost = mouses.map(e => e.price).reduce((acc, curr) => acc + curr) / mouses.length;

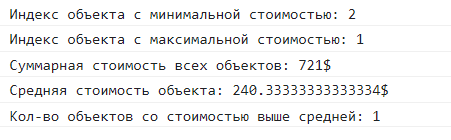
console.log('Кол-во объектов со стоимостью выше средней: ' + mouses.filter(e => e.price > averageCost).length)

</script>

</body>

</html>

**Результат:**

****

**Вывод:** в ходе выполнения лабораторной работы были получены навыки создания объектов, доступа к полям и методам.