**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине: «Современные технологии создания Web-систем»

на тему: **«**Наследование в js»

Выполнил: студент гр. ИТИ-21

Беловский С.С.

Принял: ассистент

Карась О.В.

Гомель 2025

**Цель работы:**

Реализовать базовый класс Point, который характеризуется положением на осях координат (свойства для координат x, y, z) и условным обозначением (имя объекта-точки). Определите класс-наследник ColorPoint, который наследует свойства родительского класса и имеет дополнительную характеристику (с дополнительными полем "цвет" ). Создать коллекцию разработанных объектов и создать методы класса для доступа к значениям его полей. Написать функцию, которая осуществляет поиск по набору точек и ищет ближайюшую для заданных координат.

**Порядок выполнения работы**

**Задание**

Реализован базовый класс Point, который характеризуется положением на осях координат (свойства для координат x, y, z) и условным обозначением (имя объекта-точки). Определен класс-наследник ColorPoint, который наследует свойства родительского класса и имеет дополнительную характеристику (с дополнительными полем "цвет" ). Создана коллекция разработанных объектов и методы класса для доступа к значениям его полей. Написана функция, которая осуществляет поиск по набору точек и ищет ближайюшую для заданных координат. Результат выполнения задания представлен на рисунке 1.

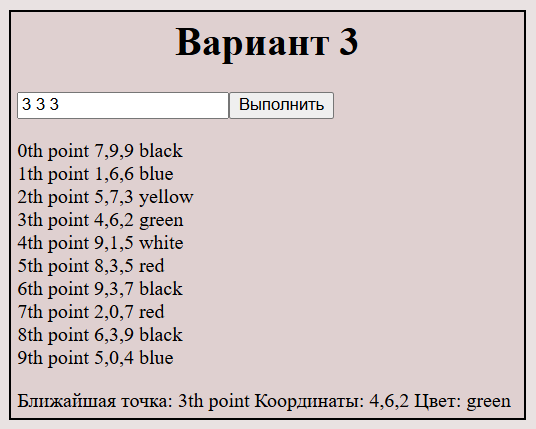


Рисунок 1 – Результат выполнения задания.

**Выводы:** На лабораторной работе изучено наследование в js.

**ПРИЛОЖЕНИЕ А**

**Файл 1.js**

class Point {

constructor(name, x, y, z) {

this.name = name;

this.x = x;

this.y = y;

this.z = z;

}

getName() {

return this.name;

}

getCoordinates() {

return [this.x, this.y, this.z];

}

}

class ColorPoint extends Point {

constructor(name, x, y, z, color) {

super(name, x, y, z);

this.color = color;

}

getColor() {

return this.color;

}

}

let points = [];

let colors= ["red", "green", "blue", "yellow", "black", "white"];

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

points.push(new ColorPoint(i + "th point ", Math.floor(Math.random() \* 10), Math.floor(Math.random() \* 10), Math.floor(Math.random() \* 10), colors[Math.floor(Math.random() \* 6)]));

}

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

document.getElementById("result").innerHTML += points[i].getName();

document.getElementById("result").innerHTML += points[i].getCoordinates() + " ";

document.getElementById("result").innerHTML += points[i].getColor();

document.getElementById("result").innerHTML += "<br>";

}

document.getElementById("button1").onclick = function () {

let x, y, z;

let cords = document.getElementById("cords").value;

let coordinates = cords.split(" ");

x = Number(coordinates[0]);

y = Number(coordinates[1]);

z = Number(coordinates[2]);

if(isNaN(x) || isNaN(y) || isNaN(z)){

alert("Введите корректные координаты");

return;

}

getNearbyestPoint(points, x, y, z);

}

function getNearbyestPoint(points, x, y, z) {

let nearbyestPoint = points[0];

for(let i = 1; i < points.length; i++){

let currentPoint = points[i];

if(Math.abs(currentPoint.x - x) + Math.abs(currentPoint.y - y) + Math.abs(currentPoint.z - z) < Math.abs(nearbyestPoint.x - x) + Math.abs(nearbyestPoint.y - y) + Math.abs(nearbyestPoint.z - z)){

nearbyestPoint = currentPoint;

}

}

document.getElementById("answer").innerHTML = "Ближайшая точка: " + nearbyestPoint.getName() + " Координаты: " + nearbyestPoint.getCoordinates() + " Цвет: " + nearbyestPoint.getColor();

}