**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

##### Группа 73 ТП

## Отчет

## по лабораторной работе № 12

**"Создание объектов и методов объектов. Работа со свойствами объектов"**

учебный предмет

“Программные средства создания Интернет приложений”

**Исполнитель: А.А. Артюшевский**

**Руководитель: Е.П. Сушкевич**

**Минск, 2025**

**Цель работы:**

Сформировать умение использовать объекты в сценариях на языке JavaScript

**Задания:**

1. Создайте объект car с тремя свойствами (марка, модель, год выпуска) и методом, который выводит информацию об автомобиле.
2. Добавьте объекту user свойство isAdmin и метод toggleAdmin, который меняет значение этого свойства на противоположное.
3. Реализуйте функцию-конструктор Book, которая создает объекты с полями title, author, year и методом getSummary. Создайте три книги и выведите их краткое описание.
4. Создайте объект calculator с методами add, subtract, multiply, divide, которые работают с двумя внутренними свойствами a и b. Реализуйте методы для установки значений этих свойств.
5. Напишите функцию, которая принимает объект и имя свойства, и возвращает true, если свойство существует, и false — если нет.
6. Создайте объект student с методом addGrade(grade), который добавляет оценку в массив оценок, и методом getAverage(), возвращающим средний балл.
7. Реализуйте объект bankAccount с приватным свойством balance (используйте замыкание), методами deposit, withdraw и getBalance.
8. Создайте объект rectangle с двумя свойствами (ширина и высота) и методами для вычисления площади и периметра.
9. Напишите метод для объекта person, который возвращает строку с перечислением всех его свойств и их значений.
10. Добавьте к объекту shop метод addProduct(name, price), который добавляет товар (объект с двумя свойствами) в массив товаров, и метод getTotalPrice(), возвращающий сумму цен всех товаров.

Решение:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        // 1

        const car = {

            brand: "Toyota",

            model: "Corolla",

            year: 2020,

            getInfo() {

                return `${this.brand} ${this.model}, ${this.year} г.в.`;

            }

        };

        // 2

        const user = {

            isAdmin: false,

            toggleAdmin() {

                this.isAdmin = !this.isAdmin;

            }

        };

        // 3

        function Book(title, author, year) {

            this.title = title;

            this.author = author;

            this.year = year;

            this.getSummary = function () {

                return `${this.title} (${this.year}), автор — ${this.author}`;

            };

        }

        const book1 = new Book("1984", "Джордж Оруэлл", 1949);

        const book2 = new Book("Мастер и Маргарита", "Михаил Булгаков", 1966);

        const book3 = new Book("Над пропастью во ржи", "Дж. Д. Сэлинджер", 1951);

        // 4

        const calculator = {

            a: 0,

            b: 0,

            setValues(x, y) {

                this.a = x;

                this.b = y;

            },

            add() { return this.a + this.b; },

            subtract() { return this.a - this.b; },

            multiply() { return this.a \* this.b; },

            divide() { return this.b !== 0 ? this.a / this.b : "Деление на ноль!"; }

        };

        // 5

        function hasProperty(obj, prop) {

            return prop in obj;

        }

        // 6

        const student = {

            grades: [],

            addGrade(grade) {

                this.grades.push(grade);

            },

            getAverage() {

                const sum = this.grades.reduce((a, b) => a + b, 0);

                return this.grades.length ? sum / this.grades.length : 0;

            }

        };

        // 7

        function createBankAccount() {

            let balance = 0;

            return {

                deposit(amount) { balance += amount; },

                withdraw(amount) { if (amount <= balance) balance -= amount; },

                getBalance() { return balance; }

            };

        }

        const bankAccount = createBankAccount();

        // 8

        const rectangle = {

            width: 5,

            height: 10,

            getArea() {

                return this.width \* this.height;

            },

            getPerimeter() {

                return 2 \* (this.width + this.height);

            }

        };

        // 9

        const person = {

            name: "Артём",

            age: 30,

            city: "Минск",

            describe() {

                return Object.entries(this)

                    .map(([key, value]) => `${key}: ${value}`)

                    .join(", ");

            }

        };

        // 10

        const shop = {

            products: [],

            addProduct(name, price) {

                this.products.push({ name, price });

            },

            getTotalPrice() {

                return this.products.reduce((sum, p) => sum + p.price, 0);

            }

        };

    </script>

</body>

</html>

Контрольные вопросы:

1. Какие существуют основные способы создания объектов в JavaScript? Приведите примеры использования литеральной нотации и функции-конструктора.

Ответ:

**а. Литеральная нотация** – самый распространённый способ:

const car = {

brand: "Toyota",

model: "Corolla",

year: 2020

};

**б. Функция-конструктор:**

function Car(brand, model, year) {

this.brand = brand;

this.model = model;

this.year = year;

}

const myCar = new Car("Honda", "Civic", 2022);

**в. Классы (синтаксический сахар поверх конструктора):**

class Car {

constructor(brand, model, year) {

this.brand = brand;

this.model = model;

this.year = year;

}

}

**г. Метод** Object.create()**:**

const prototype = { greet() { return "Hello"; } };

const obj = Object.create(prototype);

1. Как добавить метод к объекту, созданному с помощью литеральной нотации? Как внутри метода обратиться к свойствам этого же объекта?

Ответ:

const user = {

name: "Артём",

age: 30,

greet() {

return `Привет, меня зовут ${this.name}`;

}

};

1. Что такое функция-конструктор в JavaScript и как с её помощью создавать несколько объектов с одинаковой структурой? Как используется ключевое слово this внутри конструктора?

Ответ:

Функция-конструктор создаёт новые объекты с одинаковой структурой:

javascript

function User(name, age) {

this.name = name;

this.age = age;

this.sayHi = function() {

return `Привет, я ${this.name}`;

};

}

const u1 = new User("Иван", 28);

const u2 = new User("Ольга", 35);

1. Чем отличаются объекты, созданные с помощью литерала {}, конструктора new Object() и метода Object.create()? В каких случаях предпочтительно использовать каждый из этих способов?

Ответ:

|  |  |  |
| --- | --- | --- |
| **Способ** | **Пример** | **Особенности** |
| Литерал | {} | Кратко и читаемо. Используется в 90% случаев. |
| new Object() | new Object() | Эквивалент {}, но реже используется. |
| Object.create(proto) | Object.create(obj) | Позволяет задать прототип объекта вручную. Подходит для глубокого контроля над наследованием. |

1. Как получить, изменить и удалить свойства объекта? Какие существуют способы обращения к свойствам объекта (через точку и через квадратные скобки) и в каких случаях применяются?

Ответ:

const person = {

name: "Лена",

age: 25

};

// Получение:

console.log(person.name); // через точку

console.log(person["age"]); // через скобки

// Изменение:

person.name = "Елена";

// Удаление:

delete person.age;

const prop = "name";

console.log(person[prop]); // "Елена"