Федеральное государственное образовательное бюджетное учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Колледж информатики и программирования

ПМ.08 Разработка кода информационных систем

Группа: 3ИСИП-622

УТВЕРЖДАЮ

Председатель предметно-цикловой комиссии информационных систем и программирования

_____ Т.Г. Аксёнова

«____» _____2025 г.

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9

Разработка калькулятора геометрических фигур

Преподаватель

_____Р. Р. Абзалимов

Исполнитель

_____Л. Д. Слепцов

Оценка: _____

«____» _____2025 г.

Москва

2025

Цель работы

Изучить основные операции с объектами в JavaScript, включая создание, доступ, изменение, удаление свойств, а также использование методов дляработы с объектами.

Ход работы

1. Как можно получить значение свойства объекта через точечную нотацию?

Используется object.property:

```
const user = { name: "Alice", age: 25 };
console.log(user.name); // "Alice"
```

Этот способ удобен, если название свойства **известно заранее**.

2. В каких случаях удобнее использовать квадратные скобки для доступа к свойствам объекта?

Квадратные скобки object["property"] удобны, если:

- **Название свойства неизвестно заранее** (переменная):

```
let key = "age";
console.log(user[key]); // 25
```

- **Свойство содержит пробелы или спецсимволы**:

```
const obj = { "full name": "John Doe" };
console.log(obj["full name"]); // "John Doe"
```

- **Свойство является результатом выражения**:

```
const obj = { ["prop" + 1]: "value" };
console.log(obj["prop1"]); // "value"
```

3. Как можно изменить значение существующего свойства объекта?

Просто присвоить новое значение:

```
user.age = 30;
```

```
console.log(user.age); // 30
```

4. Как добавить новое свойство в объект?

Просто присвоить значение новому свойству:

```
user.city = "New York";
console.log(user.city); // "New York"
```

5. Как удалить свойство из объекта?

Используется delete:

```
delete user.age;
console.log(user.age); // undefined
```

6. Как проверить, существует ли свойство в объекте?

Способы:

1. **Оператор in**:

```
console.log("name" in user); // true
console.log("age" in user); // false (если удалили)
```

2. **Проверка на undefined** (если свойство точно не может принимать undefined):

```
console.log(user.age !== undefined); // false
```

3. **Метод hasOwnProperty** (не учитывает унаследованные свойства):

```
console.log(user.hasOwnProperty("name")); // true
```

7. Как перебрать все свойства объекта?

Используется for...in:

```
for (let key in user) {
  console.log(`${key}: ${user[key]}`);
}
```

8. Как получить массив ключей и массив значений объекта?

1. **Ключи (Object.keys)**:

```
console.log(Object.keys(user)); // ["name", "city"]
```

2. **Значения (Object.values)**:

```
console.log(Object.values(user)); // ["Alice", "New York"]
```

3. **Ключи + значения (Object.entries)**:

```
console.log(Object.entries(user)); // [["name", "Alice"], ["city", "New York"]]
```

9. Как создать поверхностную копию объекта?

1. **Spread-оператор (...)**:

```
const userCopy = { ...user };
```

2. **Object.assign**:

```
const userCopy = Object.assign({}, user);
```

3. **Глубокое копирование (JSON.parse(JSON.stringify()))**, если в объекте есть вложенные объекты:

```
const deepCopy = JSON.parse(JSON.stringify(user));
```

10. Что такое деструктуризация и как она используется с объектами?

Деструктуризация — это способ извлечения значений из объекта в переменные:

```
const user = { name: "Alice", age: 25, city: "New York" };
const { name, age } = user;
console.log(name); // "Alice"
console.log(age);  // 25
```

Также можно задать **значения по умолчанию**:

```
const { name, country = "USA" } = user;
console.log(country); // "USA"
```

Можно **переименовать переменные**:

```
const { name: userName } = user;
console.log(userName); // "Alice"
```