

1. Чем отличается класс от объекта? Приведите примеры и того, и другого, отличные от указанных в уроке.

Ответ: Класс представляет описание объекта, его состояния и поведения, а объект является конкретным воплощением или экземпляром класса

```
class Hero {  
  
    constructor(name, level) {  
  
        this.name = name;  
  
        this.level = level;  
  
    }  
  
}  
  
const hero1 = new Hero('Varg', 1);  
const hero2 = new Mage('Lejon', 2, 'Magic Missile');
```

1. Приведите примеры объектно-ориентированного подхода в реальной жизни.
Ответ: *Магазин :Сотрудники (ФИО, должность, зарплата, дата начала работы) и Клиенты (ФИО, контакты), Товары (категория , стоимость, вес)*
Приведите примеры объектно-ориентированного подхода в программировании.
Клиенты в интернет магазине (ФИО , контакты , товары в корзине), пользователь приложения (дата регистрации, логин , почта), персонаж в игре(имя , уровень, класс).

2. Что такое конструктор? Самостоятельно изучите и напишите, какие бывают виды конструкторов.

Ответ:

Конструктор - это специальная функция, задача которой заполнить пустой объект свойствами и методами. Иными словами, конструктор - это функция, которая конфигурирует объект для дальнейшего использования.

Встроенные конструкторы (Built-In Constructors)

```
new String()      // A new String object  
new Number()     // A new Number object  
new Boolean()    // A new Boolean object  
new Object()     // A new Object object  
new Array()      // A new Array object  
new RegExp()     // A new RegExp object  
new Function()   // A new Function object  
new Date()       // A new Date object
```

3. Что выведет код? Почему именно так?

```
function bike() {  
    console.log(this.name)  
}  
  
var name = "ninja"  
var obj1 = { name: "pomidor", bike: bike }  
var obj2 = { name: "site", bike: bike }  
  
bike()  
obj1.bike()  
obj2.bike()
```

Ответ: Выведет : ninja

pomidor

site

П/ч сначала обращается к глобальной области видимости, а потом уже к локальным по очереди

4. Чем статические свойства и методы отличаются от нестатических? В каких ситуациях они применяются?

Ответ: Методы и свойства класса, которые не привязаны к конкретному объекту и могут существовать без создания экземпляра и слова **new**, называются *статическими*.

В классе такие методы обозначаются ключевым словом **static**

Обычно статические методы используются для реализации функций, принадлежащих классу, но не к каким-то конкретным его объектам

5. Самостоятельно изучите, что такое геттеры и сеттеры, приведите пример класса с их использованием.

Ответ:

Есть два типа свойств объекта.

Первый тип это *свойства-данные (data properties)*. Мы уже знаем, как работать с ними. Все свойства, которые мы использовали до текущего момента, были свойствами-данными.

Второй тип свойств мы ещё не рассматривали. Это *свойства-аксессоры (accessor properties)*. По своей сути это функции, которые используются для присвоения и получения значения, но во внешнем коде они выглядят как обычные свойства объекта.

Геттеры и сеттеры

Свойства-аксессоры представлены методами: «геттер» – для чтения и «сеттер» – для записи. При литеральном объявлении объекта они обозначаются `get` и `set`:

```
et user = {
  name: "John",
  surname: "Smith",

  get fullName() {
    return `${this.name} ${this.surname}`;
  },

  set fullName(value) {
    [this.name, this.surname] = value.split(" ");
  }
};

// set fullName запустится с данным значением
user.fullName = "Alice Cooper";

alert(user.name); // Alice
alert(user.surname); // Cooper
```

6. Что выведет код?
Почему именно так?

Ответ: object

п.ч. Это экземпляр класса

```
class Person {  
  constructor(name) {  
    this.name = name  
  }  
}  
  
const member = new Person("John")  
console.log(typeof member)
```

7. Что выведет код? Почему именно так?

```
const person = {  
  name: "Valera",  
  age: 23  
}  
  
let city = person.city  
city = "Amsterdam"  
  
console.log(person)
```

Ответ: object , п.ч. Это экземпляр класса