1. **Объект** — это набор собранных внутри одной переменной свойств и/или функций. Каждое свойство представляет собой пару имя-значение.

К примеру, объект йогурт.

Есть коллекция свойств, которые определяют данный объект:

Дата производства: 01.10.2022

Срок хранения:3 недели

Название: «Вкусняшка»

Состав: молоко, кисломолочные бактерии

**Класс** — это шаблон, описывающий свойства и методы, которыми будет обладать любой объект, созданный на основе этого класса.

К примеру, класс человек, к которому принадлежат различные объекты(профессии): объект учитель, объект программист, объект строитель.

1. Пример объектно-ориентированного подхода в реальной жизни:

Университет содержит классы факультет(номер корпуса, название факультета, преподавательский состав), преподаватели(ФИО, степень, количество рабочих часов), студенты(ФИО, возраст, название факультета, название специальности).

Пример объектно-ориентированного подхода в программировании: пост в инстаграм (дата публикации, тема поста, автор)

1. Функция-конструктор описывает шаблон, по которому будут создаваться объекты (экземпляры). В основе такого шаблона лежит перечисление свойств и методов, которыми будет обладать объект, созданный на основе единой структуры (класса). Бывают

* встроенные конструкторы,
* конструктор с параметрами;
* [конструктор по умолчанию](https://www.wikiwand.com/ru/Конструктор_по_умолчанию), не принимающий аргументов;
* именованный конструктор — функция, предполагающая явный вызов по имени, работающая как конструктор
* [конструктор копирования](https://www.wikiwand.com/ru/Конструктор_копирования) — конструктор, принимающий в качестве аргумента объект того же класса (или ссылку из него);
* конструктор преобразования — конструктор, принимающий один аргумент (эти конструкторы могут вызываться автоматически для преобразования значений других типов в объекты данного класса).

1. Через консоль лог выводим ninja, pomidor, site
2. Статические поля хранят состояния класса в целом, а не отдельного объекта. Перед названием статического поля ставится ключевое слово static. К статическому полю нельзя обратиться через this.
3. **геттеры** — это функции, которые возвращают значения закрытых переменных-членов класса;

**сеттеры** — это функции, которые позволяют присваивать значения закрытым переменным-членам класса.

let obj = {

*get propName()* {

// геттер, срабатывает при чтении obj.propName

},

*set propName(value)* {

// сеттер, срабатывает при записи obj.propName = value

}

};

1. Выведет Object, так как мы используем оператор typeof, а он позволяет определить тип параметра.
2. Выведет имя и возраст персоны