# Week 19

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

*Объект представляет собой уникальный набор ключей и свойств, а класс – подобие шаблона с помощью которого можно быстро создавать объекты.*

1. Приведите примеры объектно-ориентированного подхода в реальной жизни.

*Университет. Классы – Группа (факультет, кафедра, номер курса, направление обучения, колич-во студентов, кол-во должников, ФИО старосты); Преподаватели (ФИО, предмет, стаж работы, степень, кол-во опубликованных статей); Факультет (Название факультета, ФИО декана, ФИО заместителя, кол-во студентов)*

Приведите примеры объектно-ориентированного подхода в программировании.

*заметки (заголовок, тело заметки, дата создания)*, *закладка в панели меню (иконка, название, ссылка)*

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

*Конструктор используется. когда нужно создать множество однотипных объектов с помощью оператора new. Конструкторы не имеют оператора return. Их задача – записать все необходимое в this, и это автоматически становится результатом. Если все же в конструкторе записан return, то он вернет объект return, если return пустой или возвращает примитив, то функция вернет this.*

1. Что выведет код? Почему именно так? *– bike() – ninja*, *obj1.bike() выведет pomidor, obj2.bike() выведет site. Если используется строгий режим, то функция bike() вывело бы undefined, если не строгий режим, то this в таком случае будет глобальный объект window, а у него есть свойство name со значением ninja.*

Изображение выглядит как текст

Автоматически созданное описание

1. Чем статические свойства и методы отличаются от нестатических? В каких ситуациях они применяются? - *Статические методы используются для функциональности, принадлежат классу «в целом», а не относятся к конкретному объекту класса. Статические свойства используются в тех случаях, когда мы хотели бы сохранить данные на уровне класса, а не какого-то одного объекта.*
2. Самостоятельно изучите, что такое геттеры и сеттеры, приведите пример класса с их использованием. - *иногда нужно, чтобы свойство устанавливалось только при создании объекта и после этого никогда не изменялось, тогда используется геттер*.

Сеттер используется когда нужно ограничить доступ на изменение свойства объекта.

class CoffeeMachine {

\_waterAmount = 0;

set waterAmount(value) {

if (value < 0) throw new Error("Отрицательное количество воды");

this.\_waterAmount = value;

}

get waterAmount() {

return this.\_waterAmount;

}

constructor(power) {

this.\_power = power;

}

}

1. Что выведет код? Почему именно так? – *объект*

Изображение выглядит как текст

Автоматически созданное описание

1. Что выведет код? Почему именно так? – *результатом кода будет объект person с первоначальными свойствами, так как присваивается значение объекта с ключом city, а не наоборот. Меняется значение примитивной переменной city.*

Изображение выглядит как текст

Автоматически созданное описание