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

Класс это шаблон, который описывает свойства и методы, которыми будет обладать любой объект, созданный с помощью этого класса. А объект это объект, в котором содержится уже конкретная коллекция свойств или функций.

Классом может быть, к примеру каталог товаров в онлайн-магазине (свойства - картинка, название, цена, размеры, а методом - добавить в корзину), а каждая отдельная карточка товара это объект (куртка, название бренда куртки, её цена, размеры)

1. Приведите примеры объектно-ориентированного подхода в реальной жизни.  
     
    *Возможный вариант ответа: например, библиотека могла бы содержать классы Книга (название, автор, год выпуска, количество страниц), Сотрудники (ФИО, должность, зарплата, дата начала работы) и Читатели (ФИО, контакты, номер читательского билета).*

Бланк товарного чека (номер чека, позиции, кол-во, цена, сумма, итог) - класс. Заполненный товарный чек - объект. Приведите примеры объектно-ориентированного подхода в программировании.  
  
 *Возможный вариант ответа: например, товары в интернет магазине (название товара, тип, цена), комментарии в соцсети (автор комментария, дата, текст комментария), видео в Youtube (название видео, его адрес, автор и дата загрузки).*

Личный кабинет на сайте - класс (имя, дата рождения, адрес, аватарка), а личный кабинет конкретного пользователя - объект.

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

Это специальный метод, служащий для создания и инициализации объектов, созданных с использованием “class”.

User-defined Constructors - конструкторы, объявленные и определенные программистом для использования в конкретном приложении. (например создание классов, функций с нужными нам свойствами, методами, аргументами).

Built-in Constructors - конструкторы, поставляемые в комплекте со средой выполнения (например Array, Date, и Object.)

1. Что выведет код? Почему именно так?  
     
   

bike(); - пустая строка, тк мы не указали объект

obj1.bike(); - pomidor, тк это name этого объекта

obj2.bike(); - site, тк это name этого объекта

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

Статические свойства и методы определяются для всего класса в целом, а не для отдельного объекта, и вызывать мы их можем только непосредственно из класса. Статические методы применяются, когда нам необходимо создать или клонировать объекты, а свойства полезны для хранения информации, которую нам не нужно реплицировать между экземплярами.

1. Самостоятельно изучите, что такое геттеры и сеттеры, приведите пример класса с их использованием.  
   Геттеры и сеттеры - это специальные методы, доступ к которым осуществляется через связанное с каждым из них свойством. Геттер работает при чтении свойств, а сеттер при присвоении свойству какого-то значения. То есть оба этих метода создают свойство, но работающее через функции.

let obj = {

get someName() {

// геттер, срабатывает при чтении obj.someName

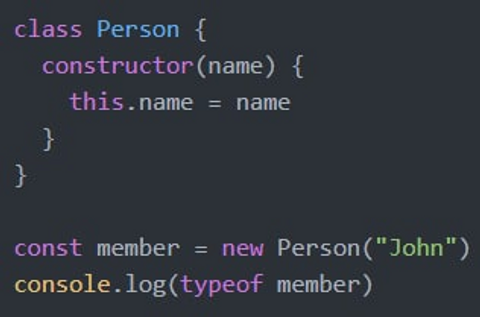
},

set someName(value) {

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

}

};

1. Что выведет код? Почему именно так?  
     
   

Object, потому что это const member создался по классу Person.

1. Что выведет код? Почему именно так?



Выведет свойства объекта, тк person это созданный объект.

{name: ‘Valera’, age: 23, city: ‘Amsterdam’}