JavaScript Базовый

Классы. Наследование

Introduction

Охрименко Дмитрий

MCT

_okhrimenko

dmitriy.okhrimenko

dokhrimenko

MCID: 9210561

Тема урока

Классы. Наследование

План урока

Что такое наследование
Ключевое слово extends
Конструкторы и ключевое слово super
Переопределение методов
Статические методы и свойства

Наследование

Наследование – концепция объектно-ориентированных языков программирования, согласно которой один тип данных (класс), может наследовать данные и функциональность другого типа данных (класса).

Наследование способствует повторному использованию существующего кода.

Прототипное наследование – механизм наследования, поддерживаемый в JavaScript, который базируется на построении цепочки прототипов и совместном использовании функций между объектами.

Ключевые слова **class** и **extends** являются синтаксическим сахаром прототипно-ориентированной модели наследования.

Повторное использование кода

Шаблоны наследования с использованием прототипа

- Цепочка прототипов
- Заимствование конструкторов
- Заимствование конструктора и установка прототипа
- Совместное использование прототипа
- Временный конструктор

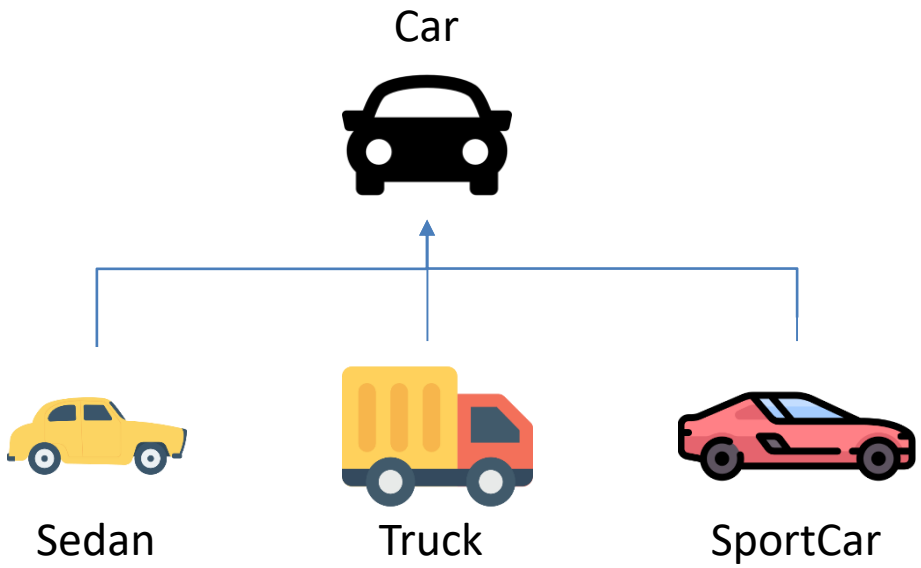
Шаблоны повторного использования кода

- Наследование через прототип
- Наследование копированием свойств
- Смешивание
- Заимствование методов (call, apply)

JavaScript Шаблоны
Урок №4. Шаблоны создания объектов и повторное использование кода

https://itvdn.com/ru/video/javascript-patterns/create-patterns

Наследование классов



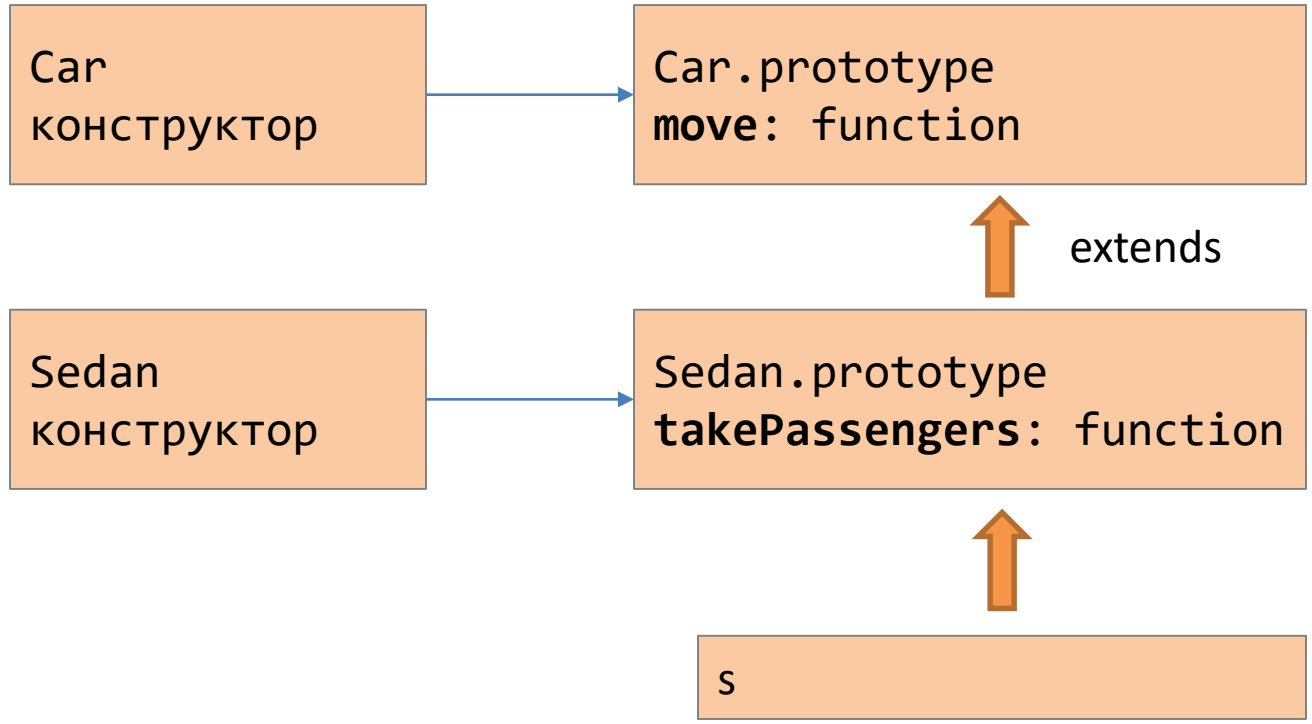

```
class Car {
    move() {}
}

class Sedan extends Car {
    takePassengers() {}
}

class Truck extends Car {
    takeCargo() {}
}

class SportCar extends Car {
    goFast() {}
}
```

Наследование классов

```
class Car {
    move() {}
}
```

```
class Sedan extends Car {
    takePassengers() {}
}
```

```
let s = new Sedan();
s.takePassengers();
s.move();
```



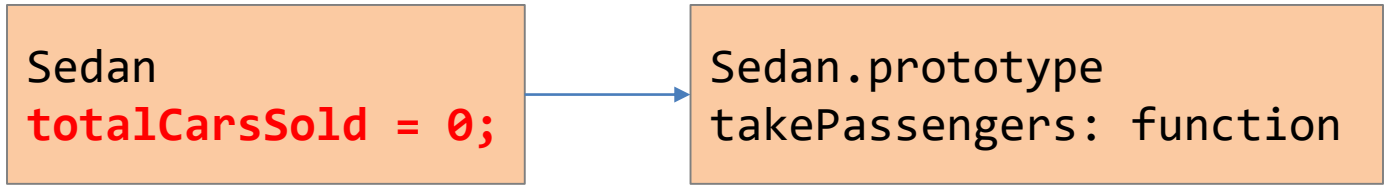
static

static – ключевое слово для определения статических полей и методов. Статический член класса существует в одном экземпляре для всего приложения.

Статические свойство или метод создаются в функции конструкторе, а не в каждом экземпляре.

Для получения доступа к статическим членам, необходимо выполнять обращение через имя класса, а не через конкретный экземпляр.

```
class Sedan extends Car {
    static totalCarsSold = 0;
    takePassengers() {}
}

Sedan.totalCarsSold ++;
```



Итоги

JavaScript использует **прототипно-ориентированную** модель наследования. Наследование упрощает **повторное использование** кода.

Если класс A наследуется от класса B – прототип класса A использует в качестве своего прототипа прототип класса B.

Наследование может быть основано напрямую через работу с прототипом функций конструкторов или через использование ключевых слов **class** и **extends.**

Ключевое слово **super** позволяет получить доступ к членам базового класса (к прототипу конструктора класса, который установлен как базовый).

Если в производном классе есть конструктор, в этом конструкторе надо обязательно вызвать конструктор родительского класса используя **super(параметры).**

Статическое свойство или метод – свойство или метод функции конструктора.

Спасибо за внимание! До новых встреч!

Охрименко Дмитрий
MCT

MCID: 9210561

Информационный видеосервис для разработчиков программного обеспечения