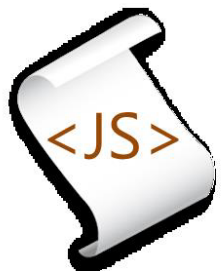


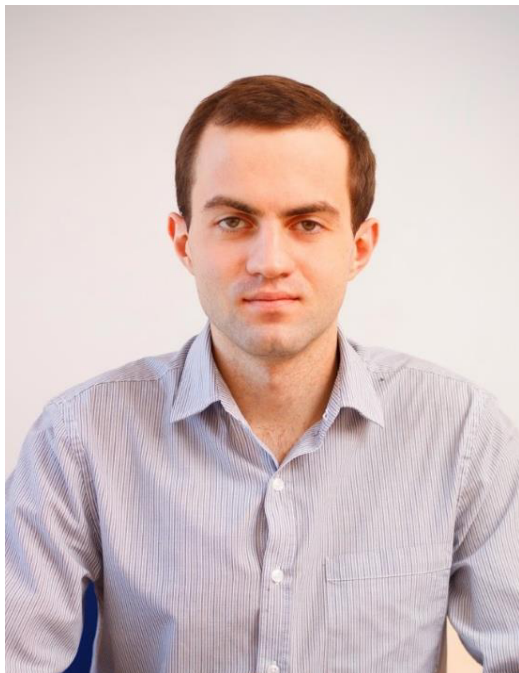
JavaScript. Шаблоны



Шаблоны создания объектов

JavaScript. Шаблоны

Автор курса



Дмитрий Охрименко
МСТ



DMYTRO OKHRIMENKO

Has successfully completed the requirements to be recognized as a Trainer.

Date of achievement: 03/19/2013
Certification number: E207-8354
Inactive Date: 05/31/2014

Steven A. Ballmer
Chief Executive Officer

Microsoft
CERTIFIED
Trainer

MCID: 9210561

JavaScript. Шаблоны

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство Вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Шаблоны создания объектов

Пространства имен

Namespaces

В JavaScript нет специальной конструкции для определения пространства имен. Для того, чтобы избежать конфликтов имен, необходимо использовать **глобальный объект**.

Антишаблон

```
var count = 1;

function Car() { /*....*/ }
function Bus() { /*....*/ }

var module = {};
module.arr = [1, 2, 3];
```

 - глобальная переменная

Одна глобальная переменная

```
var App = {};

App.count = 1;

App.Car = function () { /*....*/ }
App.Bus = function () { /*....*/ }

App.module = {};
App.module.arr = [1, 2, 3];
```

Шаблон модуль

Module

Шаблон «модуль» – позволяет структурировать и организовывать программный код по мере увеличения его объема.

```
App.util.calc = (function () {  
    // закрытые члены  
    var x, y;  
  
    return {  
        // открытые члены  
        add: function () {  
            return x + y;  
        },  
        setX: function (val) {  
            x = val;  
        },  
        setY: function (val) {  
            y = val;  
        }  
    }  
})();
```

Повторное использование кода

Code Reuse

Классическое наследование:

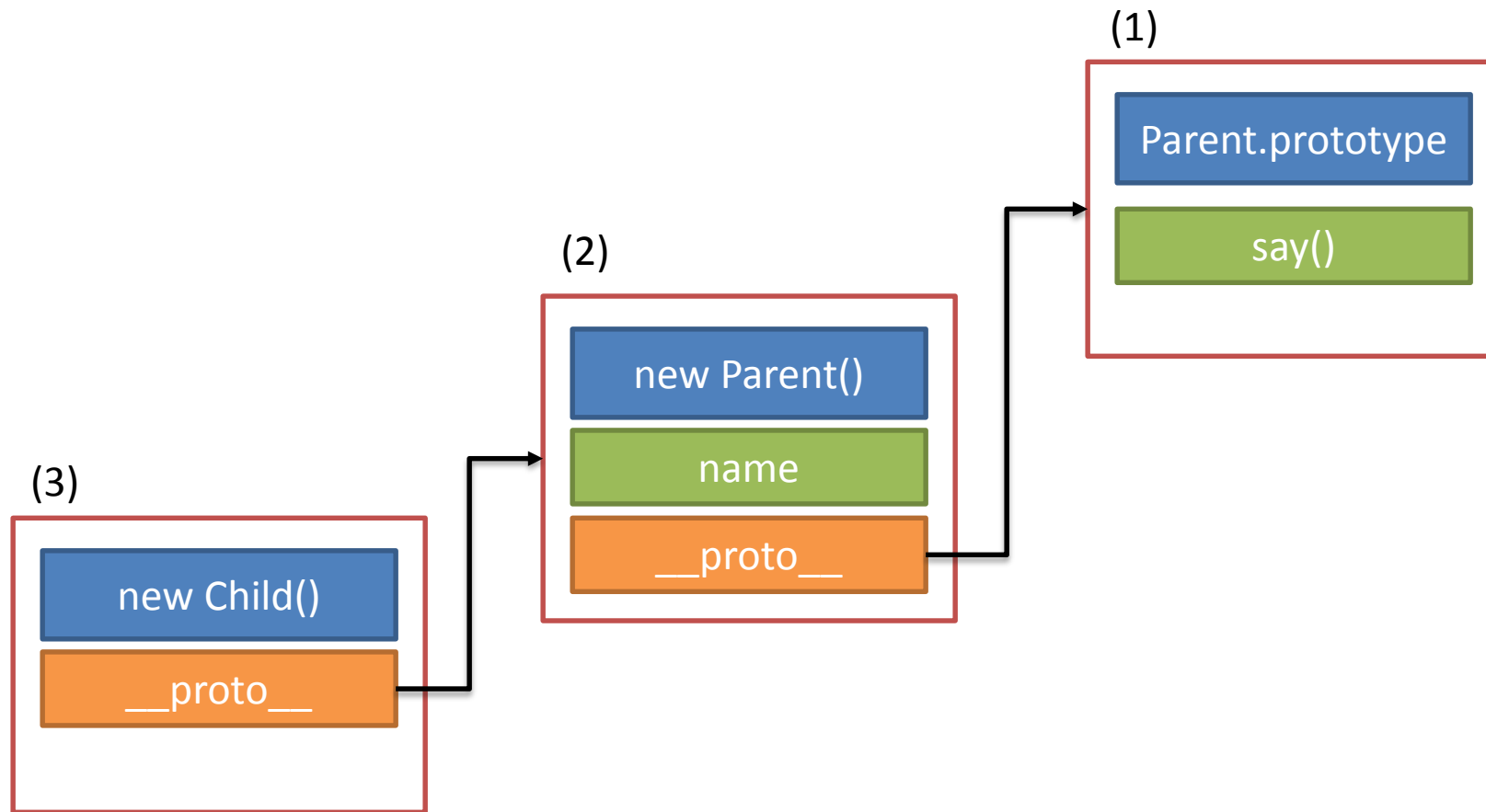
- Цепочка прототипов
- Заимствование конструкторов
- Заимствование конструктора и установка прототипа
- Совместное использование прототипа
- Временный конструктор

Современные шаблоны повторного использования кода:

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

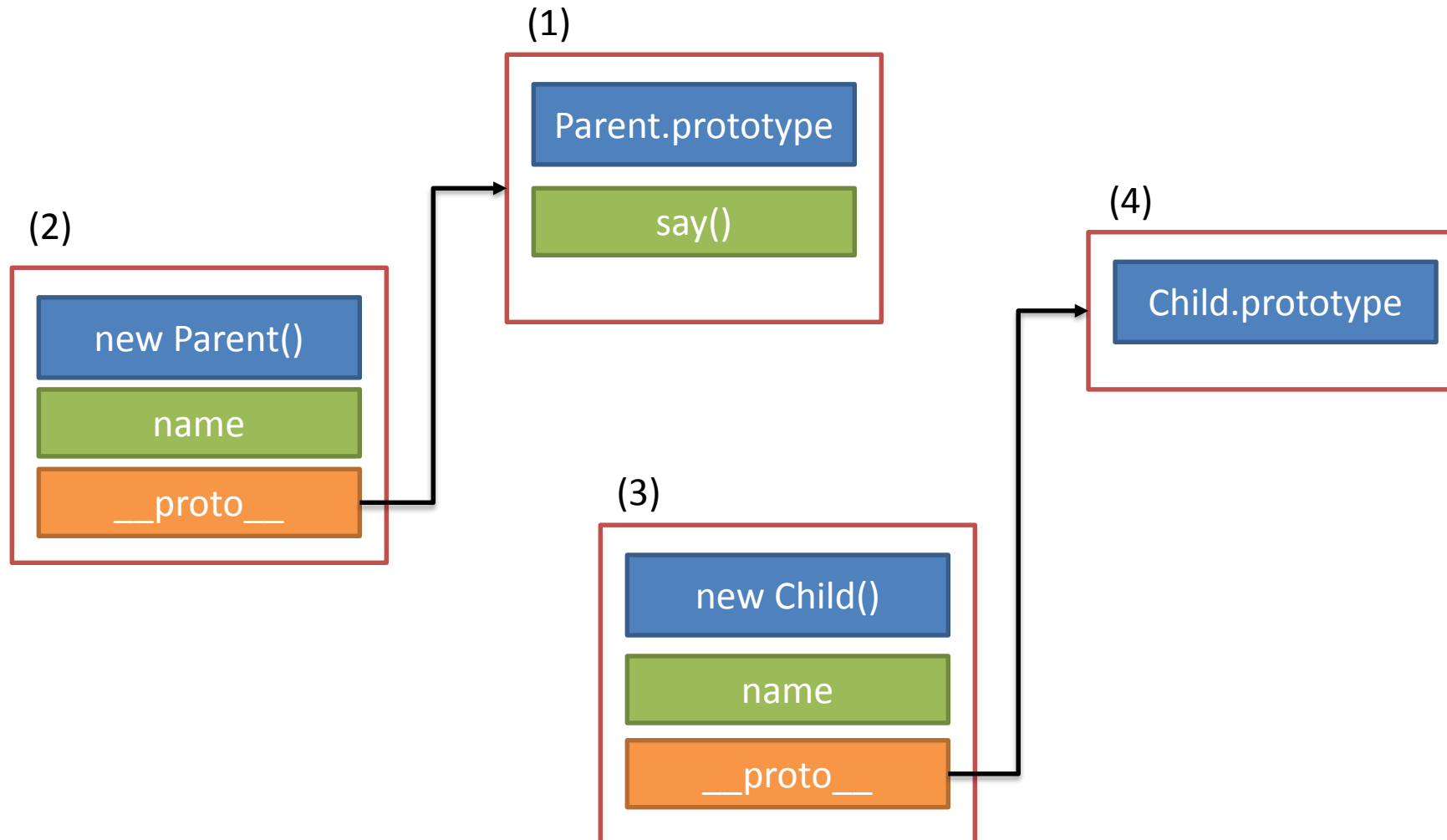
Наследование. Шаблон №1

Цепочка прототипов



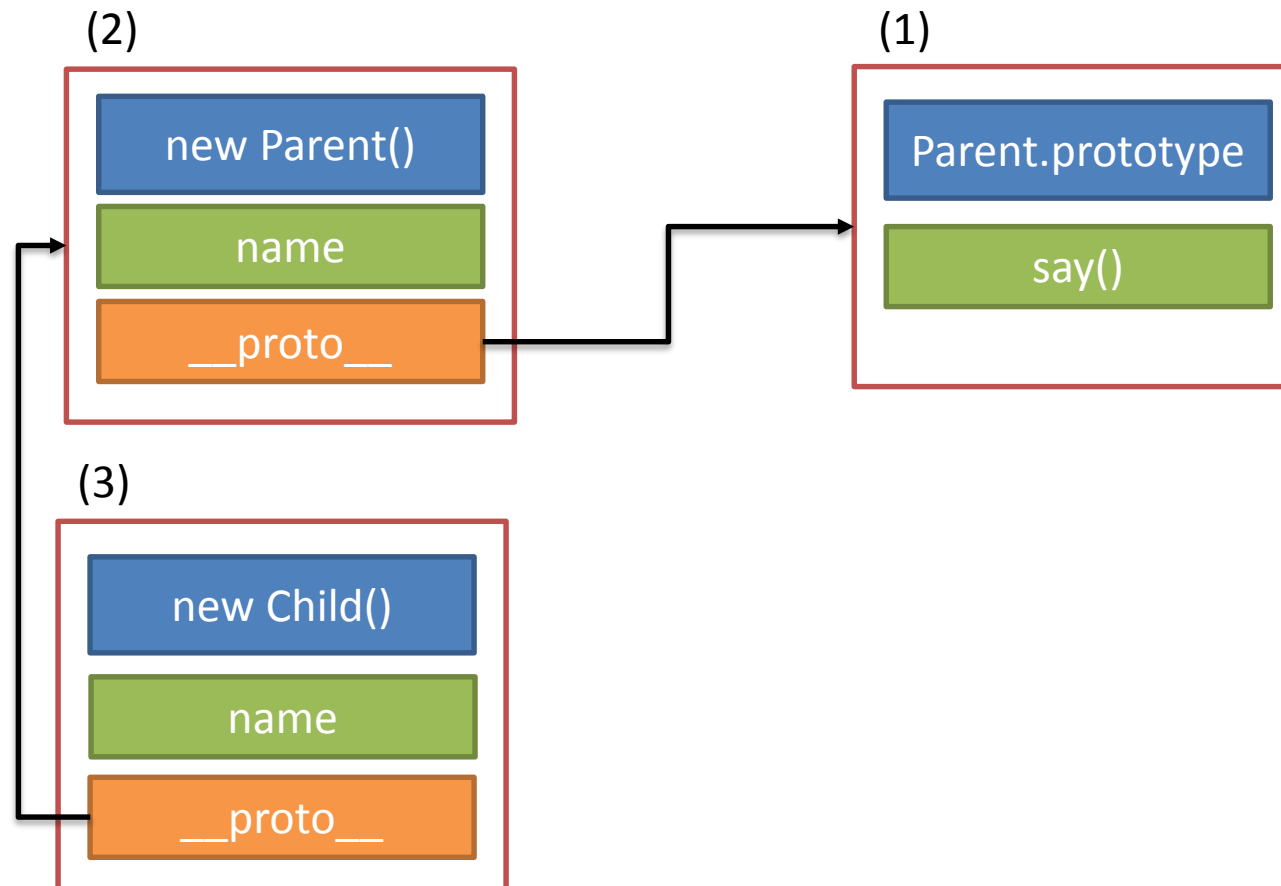
Наследование. Шаблон №2

Заимствование конструктора



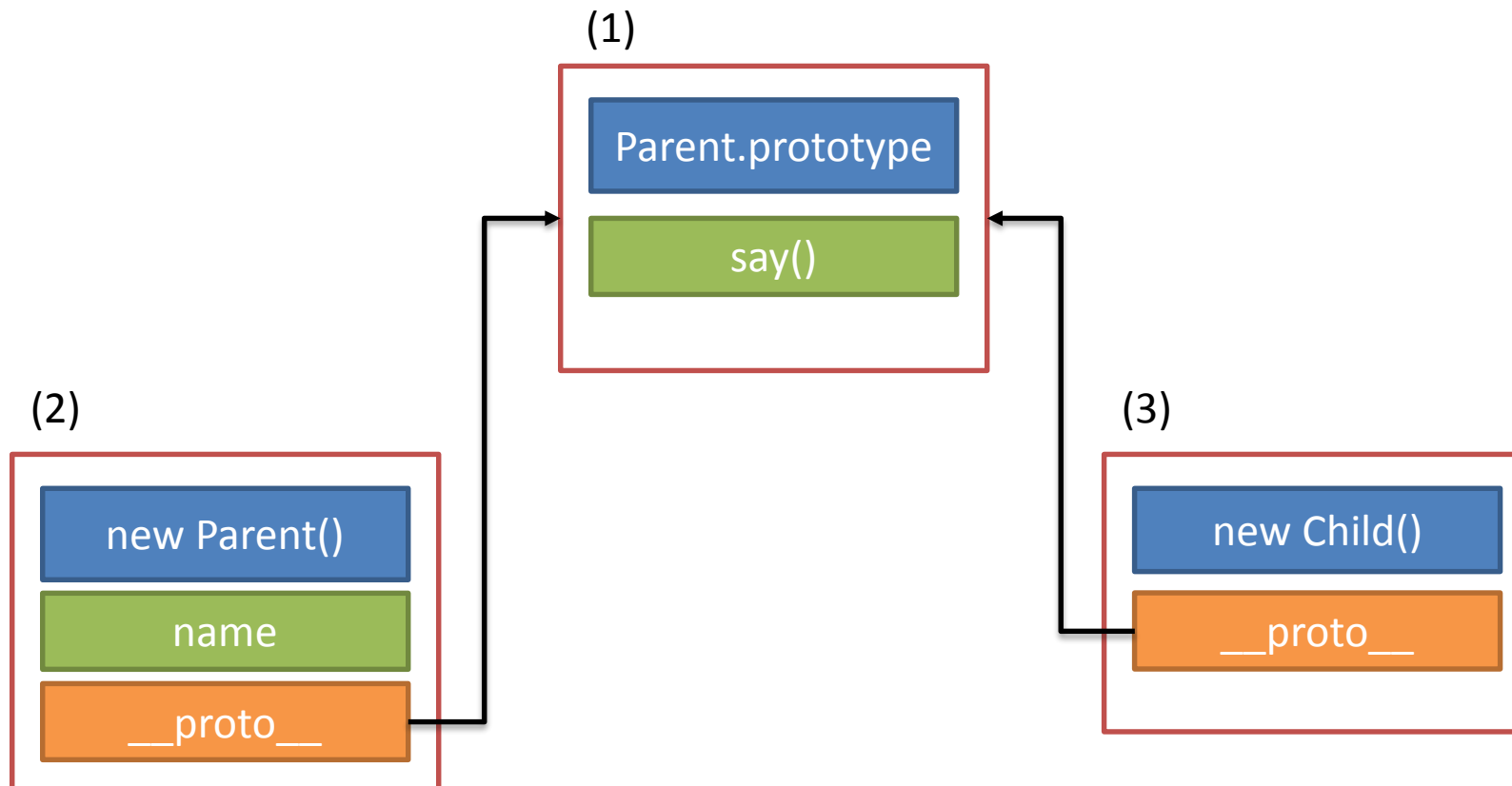
Наследование. Шаблон №3

Заимствование и установка прототипа



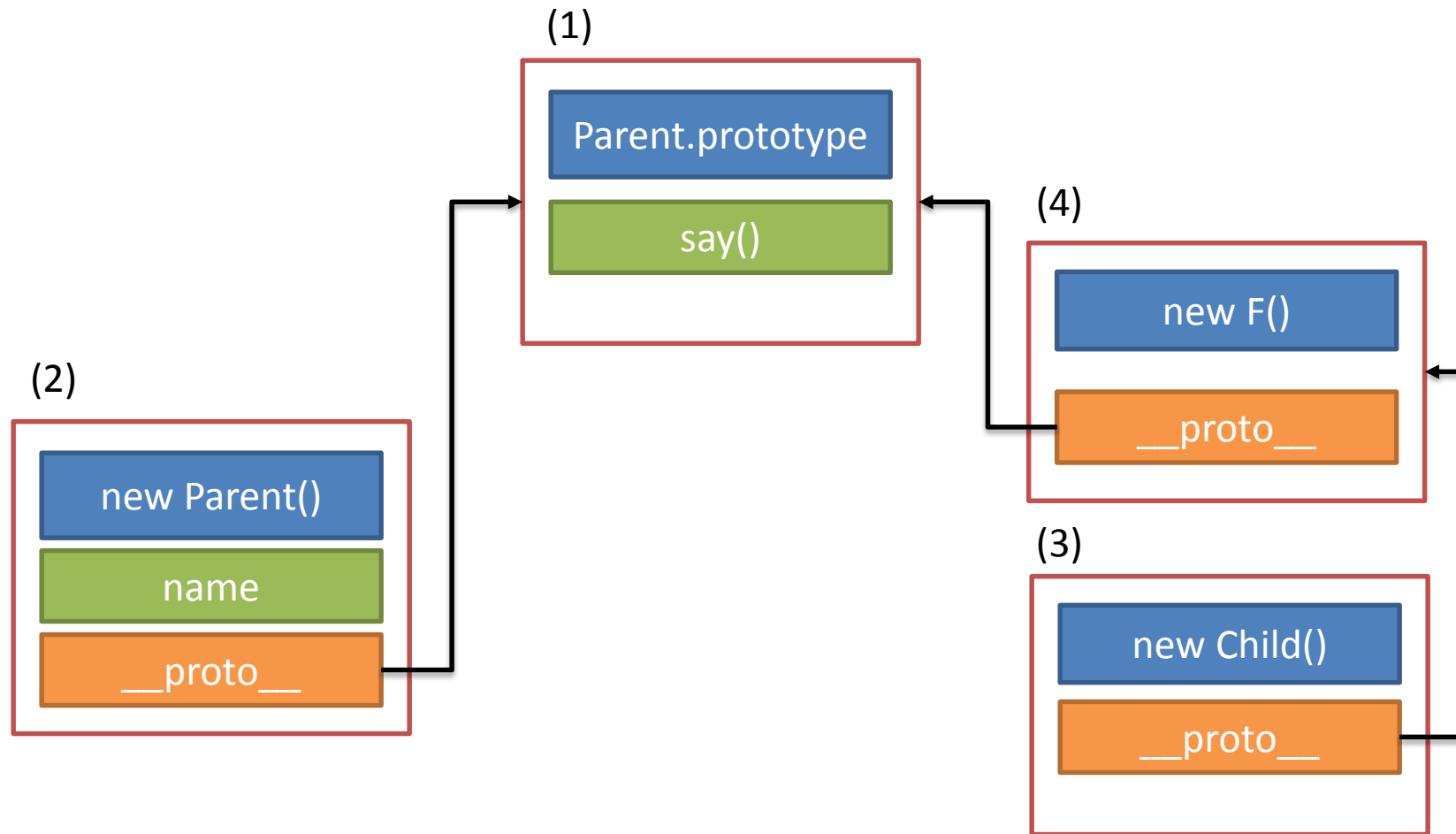
Наследование. Шаблон №4

Совместное использование прототипа



Наследование. Шаблон №5

Временный конструктор



Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

