# **Краткое описание Factory Pattern**

Фабричный метод (англ. Factory Method также известен как Виртуальный конструктор (англ. Virtual Constructor)) — порождающий шаблон проектирования, предоставляющий подклассам интерфейс для создания экземпляров некоторого класса. В момент создания наследники могут определить, какой класс создавать. Иными словами, Фабрика делегирует создание объектов наследникам родительского класса. Это позволяет использовать в коде программы не специфические классы, а манипулировать абстрактными объектами на более высоком уровне.

# **Factory**

Собственно, основной задачей фабрики в статически типизируемых языках является создание разных объектов с одинаковым интерфейсом, в зависимости от ситуаций, в JavaScript этак проблема так остро не стоит, так что появляется вопрос — зачем эта фабрика тут вообще нужна?

Все просто — помимо этой, первой, цели, у нее есть еще и вторая — фабрика может проводить какую-то первичную инициализацию объектов.

Например, предположим, у нас есть объекты Daddy, Mammy, и lad, создавая их с помощью фабрики мы можем просто сказать — familyfactory.createLad(); familyfactory.createDaddy(), а уж то, что они оба рыжие и 210см. роста, за нас решит фабрика — эти параметры мы не задаем.

Собственно, для того чтобы фабрика могла создавать какие-то объекты, для них сначала неплохо бы задать конструкторы (в этом примере объекты, к сожалению, не такие интересные как несколькими строками выше ):

var Shapes =

{

Circle: function (param)

{

console.log("new " + param.color + " circle created with radius " + param.radius + "px");

},

Square: function (param)

{

console.log("new " + param.color + " square created with " + param.side + "px on a side ");

},

Triangle: function (param)

{

console.log("new " + param.color + " triangle created with " + param.side + "px on a side ");

}

}

А теперь можно сделать и саму фабрику — выглядеть она может так:

function ShapeFactory(size, color)

{

this.size = size;

this.color = color;

}

ShapeFactory.prototype =

{

constructor: ShapeFactory,

makeCircle: function () { return new Shapes.Circle({ radius: this.size / 2, color: this.color }); },

makeSquare: function () { return new Shapes.Square({ side: this.size, color: this.color }); },

makeTrinagle: function () { return new Shapes.Triangle({ side: this.size, color: this.color }); }

}

Пишем скромненький тест:

var factory = new ShapeFactory(100, "red")

factory.makeSquare();

factory.makeSquare();

factory.makeTrinagle();

factory.makeCircle();

factory.makeTrinagle();

**Источники**

<https://carldanley.com/js-factory-pattern/>

<https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B1%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)>

<http://habrahabr.ru/post/132472/>