

Factory pattern

Factory Pattern là pattern sử dụng phương thức factory để tạo object mà không cần chỉ định chính xác class hoặc constructor function.

Factory Pattern được sử dụng để tạo các object mà không "làm lộ" logic khởi tạo. Pattern này có thể được sử dụng khi chúng ta cần tạo một object khác dựa vào một điều kiện được chỉ rõ từ trước. Ví dụ:

```
class Car {
  constructor(options) {
    this.doors = options.doors || 4
    this.state = options.state || 'brand new'
    this.color = options.color || 'white'
  }
}
class Truck {
  constructor(options) {
    this.doors = options.doors || 4
    this.state = options.state || 'used'
    this.color = options.color || 'black'
  }
}
class VehicleFactory {
  createVehicle(options) {
    if (options.vehicleType === 'car') {
      return new Car(options)
    } else if (options.vehicleType === 'truck') {
      return new Truck(options)
    }
  }
}
```

Ở đây mình tạo một class **Car** và **Truck** (với một số giá trị mặc định), 2 class này được sử dụng để tạo object **car** và **truck**. Và mình định nghĩa một class **VehicleFactory** để tạo và return một object mới dựa trên **vehicleType** nhận được từ object **options** như một đối số.

```
const factory = new VehicleFactory()
const car = factory.createVehicle({
  vehicleType: 'car',
  doors: 4,
  color: 'silver',
  state: 'Brand New'
})
const truck = factory.createVehicle({
  vehicleType: 'truck',
  doors: 2,
  color: 'white',
  state: 'used'
})
```

```
})  
// Prints Car {doors: 4, state: "Brand New", color: "silver"}  
console.log(car)  
// Prints Truck {doors: 2, state: "used", color: "white"}  
console.log(truck)
```

Mình đã tạo một object mới là `factory` từ `VehicleFactory` class. Sau đó chúng ta có thể tạo một object `Car` hoặc `Truck` mới bằng cách gọi `factory.createVehicle` và truyền một object options vào với thuộc tính `vehicleType` là `car` hoặc `truck`.