

LAB 7: OOP WITH JAVASCRIPT

MỤC TIÊU

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hướng đối tượng (OOP) trong Javascript: Constructor Functions and the new Operator, Setters & Getters, Encapsulation,...

NỘI DUNG

LAB7.1: (SECTION 14: OBJECT-ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT > CODING CHALLENGE #1)

1. Sử dụng constructor function để khởi tạo đối tượng ô tô (Car). Ô tô có 2 thuộc tính: make & speed. Speed là tốc độ hiện tại của ô tô tính bằng km/h
2. Thực hiện một phương thức (Method) "accelerate" để tăng speed hiện tại của ô tô thêm 10 và in ra speed mới của ô tô;
3. Thực hiện một phương thức (Method) "brake" giảm speed hiện tại của ô tô đi 5 và in ra speed mới của ô tô;
4. Khởi tạo 2 đối tượng ô tô (Car) và thử nghiệm gọi 2 method "brake" và "accelerate" nhiều lần trên mỗi đối tượng;

Test case

Dữ liệu ô tô 1: BMW đang chạy với tốc độ 120km/h

Dữ liệu ô tô 2: Mercedes đang chạy với tốc độ 95km/h

LAB7.2: (SECTION 14: OBJECT-ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT > CODING CHALLENGE #2)

1. Tạo lại đối tượng Car của bài 1, nhưng sử dụng ES6 class;
2. Thêm một getter 'speedUS' trả về speed hiện tại với đơn vị mi/h (chia cho 1.6, 1mi = 1.60934km);
3. Thêm một setter 'speedUS' để thay đổi tốc độ hiện tại tính bằng mi/h (nhưng trước khi lưu sẽ đổi nó sang đơn vị km/h bằng cách mi/h*1.6)
4. Khởi tạo đối tượng ô tô (Car) và thực hiện 2 phương thức (method) 'accelerate' & 'brake', 2 getter & setter;

Test case

DỮ LIỆU XE 1: Xe ô tô Ford đang di chuyển với vận tốc 120km/h

LAB7.3: (SECTION 14: OBJECT-ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT > CODING CHALLENGE #3)

1. Sử dụng constructor function để khởi tạo một đối tượng ô tô điện (ElectricCar) có tên EV như một lớp con của Car (ElectricCar extends(Kế thừa) Car), ngoài 'make' & 'speed' ElectricCar còn có mức pin hiện tại tính bằng % ('charge' property);
2. Khởi tạo phương thức (method) 'chargeBattery' nhận một đối số là 'chargeTo' và thay đổi mức pin hiện tại bằng 'chargeTo';
*Gọn: chargeBattery là một phương thức dùng để thay đổi mức pin hiện tại của ElectricCar;
3. Khởi tạo phương thức (method) 'accelerate' để tăng tốc độ ô tô thêm 20, đồng thời giảm mức pin (charge) đi 1%. Sau đó in ra dòng log: 'Tesla going at 140 km/h, with a charge of 22%';
*Gọn: accelerate thực hiện 3 việc: (speed + 20) & (charge - 1) & console.log('Tesla going at 140 km/h, with a charge of 22%');
4. Khởi tạo đối tượng ElectricCar và thực hiện các phương thức (method) 'accelerate', 'brake' và 'chargeBattery' (mức pin là 90%)
*Chú ý: Vấn đề khi thực hiện method 'accelerate', Hãy đọc lại về 'Tính đa hình của javascript' (JavaScript Polymorphism);

Test case

DỮ LIỆU XE 1: Ô tô 'Tesla' đang di chuyển với vận tốc 120 km/h, có mức pin là 23%;

LAB7.4: (SECTION 14: OBJECT-ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT > ANOTHER CLASS EXAMPLE)

Thực hiện code như mẫu sau:

```
// 1) Public fields
// 2) Private fields
// 3) Public methods
// 4) Private methods
// (there is also the static version)
class Account {
  // 1) Public fields (instances)
```

```
locale = navigator.language;
// 2) Private fields (instances)
#movements = [];
#pin;
constructor(owner, currency, pin) {
  this.owner = owner;
  this.currency = currency;
  this.#pin = pin;
  // Protected property
  // this._movements = [];
  // this.locale = navigator.language;
  console.log(`Thanks for opening an account, ${owner}`);
}
// 3) Public methods
// Public interface
getMovements() {
  return this.#movements;
}
deposit(val) {
  this.#movements.push(val);
  return this;
}
withdraw(val) {
  this.deposit(-val);
  return this;
}
requestLoan(val) {
  // if (this.#approveLoan(val)) {
  if (this._approveLoan(val)) {
    this.deposit(val);
    console.log(`Loan approved`);
    return this;
  }
}
static helper() {
  console.log('Helper');
}
// 4) Private methods
// #approveLoan(val) {
_approveLoan(val) {
  return true;
}
```

```

    }
  }
  const acc1 = new Account('Jonas', 'EUR', 1111);
  // acc1._movements.push(250);
  // acc1._movements.push(-140);
  // acc1.approveLoan(1000);
  acc1.deposit(250);
  acc1.withdraw(140);
  acc1.requestLoan(1000);
  console.log(acc1.getMovements());
  console.log(acc1);
  Account.helper();
  // console.log(acc1.#movements);
  // console.log(acc1.#pin);
  // console.log(acc1.#approveLoan(100));
  // Chaining
  acc1.deposit(300).deposit(500).withdraw(35).requestLoan(25000).withdraw(4000);
  console.log(acc1.getMovements());

```

LAB7.5: (SECTION 14: OBJECT-ORIENTED PROGRAMMING (OOP) WITH JAVASCRIPT > CODING CHALLENGE #4)

1. Khởi tạo đối tượng bài 3, nhưng sử dụng ES6 classes: khởi tạo 'EVCL' là con của 'CarCl' (EVCL extends CarCl);
3. Để property 'charge' là private;
3. Triển khai 2 method 'accelerate' và 'chargeBattery' ở đối tượng này và cập nhập lại method 'brake' trong đối tượng 'CarCl'. ???

Test case

DỮ LIỆU Ô TÔ 1: Ô tô 'Rivian' di chuyển với tốc độ 120 km/h, với mức pin là 23%

--- Hết ---