Aula 16

Orientação a objeto

Orientação a objetos é uma abordagem de programação que organiza o código de forma a representar objetos do mundo real, como pessoas, carros e animais, de uma maneira mais fácil de entender e manipular. Imagine que você tem uma classe chamada "Carro" que define como um carro deve se comportar, com métodos para ligar o motor, acelerar e frear, e propriedades como cor, marca e modelo. Quando você cria um objeto a partir dessa classe, você está criando uma instância única desse carro, com suas próprias características. Assim, a orientação a objetos permite criar programas mais organizados, onde cada objeto tem seu papel definido e pode interagir com outros objetos de forma controlada, facilitando a construção de sistemas complexos de maneira mais intuitiva.

Para criar uma classe de objetos, precisamos do seguinte comando:

```
class Nome{

constructor(variáveis){

//criar uma variável para ele

this.nome_da_variável = variável_citada_no_constructor

}

//criar um método

nome_do_método(){
```

```
código
}
}
```

Exemplo simples:

```
class Pessoa {
  constructor(nome, idade) {
    this.nome = nome;
    this.idade = idade;
  }
  saudacao() {
    console.log(`Olá, meu nome é ${this.nome} e tenho ${this.idade} anos.`);
  }
  envelhecer() {
    this.idade++;
    console.log(`Agora tenho ${this.idade} anos.`);
  }
}
let pessoa1 = new Pessoa("João", 30);
pessoa1.saudacao();
```

Exemplo complexo:

```
class Carro {
  constructor(marca, modelo, cor) {
```

```
    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.velocidade = 0;
    this.ligado = false;
}
ligar() {
    this.ligado = true;
    console.log("O carro está ligado.");
}

desligar() {
    this.ligado = false;
    console.log("O carro está desligado.");
}

acelerar() {
    if (this.ligado) {
        this.velocidade += 10;
        console.log("Acelerando. Velocidade atual: " + this.velocidade + " km/h");
    } else {
        console.log("Não é possível acelerar. O carro está desligado.");
    }
}
frear() {
    if (this.velocidade > 0) {
        this.velocidade -= 10;
        console.log("Freando. Velocidade atual: " + this.velocidade + " km/h");
```

```
        } else {
            console.log("O carro já está parado.");
        }
    }
}
let carro1 = new Carro("Chevrolet", "Onix", "Prata");
let carro2 = new Carro("Volkswagen", "Gol", "Branco");
let carro3 = new Carro("Fiat", "Uno", "Vermelho");
let carro4 = new Carro("Toyota", "Corolla", "Preto");
let carro5 = new Carro("Honda", "Civic", "Azul");
carro1.ligar()
```

Exercícios:

Exercício 1: A partir do código abaixo, crie mais 5 alunos, de turmas diferentes e diga se estão aprovados ou não.

```
class Aluno {
constructor(nome, idade, turma) {
        this.nome = nome;
        this.idade = idade;
        this.turma = turma;
        this.aprovado = false;
```

```
        }
        setAprovado(aprovado) {
                this.aprovado = aprovado;
        }
        getInfo() {
                return `${this.nome}, ${this.idade} anos, Turma: ${this.turma}, Aprovado: ${this.aprovado ? "Sim" : "Não"}`;
        }
        }
        //Dois exemplos de como criar
        let aluno1 = new Aluno("João", 17, "9A");
        let aluno2 = new Aluno("Maria", 16, "9B");
        aluno1.setAprovado(true);
        aluno2.setAprovado(false);
        console.log(aluno1.getInfo());
        console.log(aluno2.getInfo());
```

Exercício 2: Agora, ao invés deles serem variáveis, faça eles serem posições numa lista, para que assim nós possamos só chamar uma posição e teremos os itens.

Exercício 3: Agora crie uma classe animal que contenha quantidade de patas, se é mamífero, e o que come, e adicione numa lista todos os itens, para depois fazer um teste nele.

Exercício 4: Então, após a lista completa, faça uma série de ifs, que define quais animais a pessoa pode estar se referindo, por exemplo, se escrever 4 patas, mamífero e que come carne, pode ser gato, cachorro,

lobo ou leão, mas agora, se escrever bípede mamífero que come vegetais, Canguru pode ser uma opção.

Exercício 5: Pegando o código abaixo como exemplo de fazer um objeto sem usar classe, crie agora um objeto chamado, Cleber, com idade de 200 anos, quer tenha cpf 0293847213, e não tem emprego, além de ter os seguintes métodos:

Contar até sua idade atual, no caso começa em 200, **desperdiçar a vida**, onde ele fica 1 ano mais velho, e **resolver ser útil**, onde ele arranja um emprego, muda de nome pra “Cleber o agricultor”, se ele tiver um emprego já, essa função só escreve, “parabéns Cleber, tu se deu bem”.

Exercício 6: Criar um jogo onde o protagonista tem tem uma série de características, **vida**, **fome**, **sanidade**, e você tbm tem no inventário uma quantia de comida, uma de dinheiro, seu personagem tem os métodos, **trabalhar**, que consome sanidade mas te dá dinheiro, **comer**, que te ajuda na fome mas te consome itens de alimento, e **descansar**, que recupera sanidade, e **comprar comida** para aumentar seu estoque.

A cada turno, fome e sanidade diminuem em 5 pontos, caso tu tenha, sanidade abaixo de 20%, comece a perder vida, se fome zerar também reduz pontos de vida.

Exercício 6.2: Adicione elementos aleatórios que dificultem o jogo. Patrão atrasar pagamento, comida estragada, vizinho fazer barulho, etc.

Coisas que podem acontecer aleatoriamente e que vão atrapalhar no jogo.