



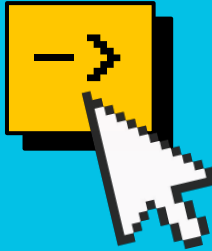
COMPETEC





Aula nº09

Orientação a Objetos



Assuntos da aula

- >Elementos da P00
- >Classes
- >Estrutura de classe
- >Elementos de classe
- >Como usar a classe

Ok

Cancel

Options<<



A Programação Orientada a Objetos (POO) ou Computação Orientada a Objetos (COO) é uma técnica de programação mais avançada em Java. Ela é uma forma de facilitar o entendimento do código, a manutenção dele e a reutilizar códigos e funções sem repetições, assim como com métodos.

Ela é formada por alguns elementos. Dentre os principais, destaca-se para esta aula: Métodos, Classes, Objetos, Atributos, Construtores.

Classes são como um roteiro para o que serão os nossos objetos. Eles que ditarão quais serão as principais propriedades e comportamentos dos objetos derivados daquela classe.

Por exemplo, uma classe poderia ser “Cachorro”, que pode latir, correr, pular, etc., e os objetos dessa classe poderiam ser “Totó”, “Marley”, etc.

Ao mesmo tempo, a classe Cachorro tem como propriedades “peso”, “idade” e “raca”, que darão algumas particularidades a cada objeto Cachorro, afinal, cada cachorro tem um valor diferente para cada uma dessas propriedades.

Para criar uma nova classe basta colocar:

Visibilidade: public, private, protected, default (sem nada);

Palavra reservada: class;

Nome: nome que identifica a classe, normalmente é o mesmo nome do arquivo sem a extensão .java e começa com letra maiúscula;

Complementos: informações sobre heranças, etc.;

Chaves: coloque as chaves para delimitar o início e fim da classe.

```
public class Cachorro {  
    //todo o código aqui  
}
```

Dentro das chaves temos basicamente 3 elementos:

Atributos: propriedades do objeto;

Construtores: método invocado quando criamos um novo objeto;

Métodos: são as ações possíveis que o objeto pode fazer;

Atributos são variáveis que guardarão informações sobre o objeto, como nome, idade, peso, poder, força, etc.

Para declarar precisamos colocar a visibilidade, o tipo de variável e o nome:

public String nome;

private String senha;

protected long CPF;

```
class Cachorro{  
    public String nome;  
    public String raca;  
    private float peso;  
    private int idade;  
}
```


Agora com os Construtores. Eles são os métodos que serão requisitados quando um novo objeto for criado e servem para instanciar os atributos (variáveis) desse novo objeto.

Para criá-lo basta colocar a visibilidade (normalmente public), o nome do método deve ser o mesmo da classe e os parâmetros necessários.

Lembrando que uma classe pode ter vários construtores, a diferença estará principalmente na quantidade e tipos de parâmetros requisitados (assinatura do método).

```
class Cachorro{
    public String nome;
    public String raca;
    private float peso;
    private int idade;

    public Cachorro(String nome, float peso){
        this.nome = nome;    //A palavra this indica que estamos
        this.peso = peso;    //nos referindo a este objeto.
        raca = " ";
        idade = -1;
    }
}
```

Por fim, os Métodos são as ações que o nosso objeto poderá fazer.

Esses são criados da mesma forma que vimos em outras aulas. Visibilidade, tipo de retorno, nome, parâmetros e as chaves {}.

```
class Cachorro{
    [...] (atributos e construtor)

    public latir( ){
        System.out.println(nome + “ ” + nome);
    }
    public boolean setIdade(int idade){
        if(idade < 0 || idade > 20) return false;
        this.idade = idade; return true;
    }
}
```

Os getters e setters são os métodos responsáveis por possibilitar a mudança do valor dos atributos. São usados para aqueles atributos não são public e que poderão ser mudados durante a execução do programa, mesmo que com restrições, como número de caracteres, ou, como no exemplo, valor máximo e mínimo.

```
class Cachorro{  
    [...] (atributos, construtor e funções)  
    public getPeso( ){  
        return peso;  
    }  
    public boolean setIdade(int idade){  
        if(idade < 0 || idade > 20)  
            return false;  
  
        this.idade = idade;  
        return true;  
    }  
}
```

Para usar a classe que criamos, precisamos de outras classes, que farão uso desta classe inicial.

Para usar, temos que instanciar um novo objeto. Assim como quando usamos o Scanner, por exemplo.

```
Scanner sc = new Scanner(System.in);  
Cachorro rex = new Cachorro("Rex", 20.4f);
```

Colocamos o nome da classe, o nome que queremos dar ao objeto e instanciamos com new (novo) NomeDaClasse e por fim os parâmetros do construtor que escolhermos seguir.

Agora que criamos o novo objeto, podemos usar todas as funcionalidades que fizemos anteriormente, colocando o nome do objeto.

Exemplos:

```
rex.lair( ); //imprime: Rex Rex
```

```
if(rex.setIdade(5)){  
    System.out.println("Idade cadastrada");  
} else {  
    System.out.println("Erro ao cadastrar idade");  
}  
//imprime: Idade cadastrada
```

COMPETEC



Obrigado pela
atenção! :)

Don't show this message again

Ok

