

## ATIVIDADE

### Assunto:

Sobrescrita ou redefinição.

### Orientações:

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

### Regras de criação dos programas:

Crie um novo projeto Java denominado **AtividadeSobrescrita**. As classes devem possuir os nomes informados no texto. Ao final, o projeto deve ser exportado para um arquivo em formato ZIP.

### Nome completo:

**Mac Myller da Silva Carlos**

1. Explique conceitualmente o significado de sobrescrita e suas regras.

**Resposta:** A sobrescrita é útil quando uma subclasse herda um atributo ou método que não é adequado para ela, então é possível modificar o tipo (no caso de atributo) ou o funcionamento de um método, desde que este método tenha o mesmo identificador, argumentos e retorno.

2. Quais as diferenças entre sobrescrita e sobrecarga?

**Resposta:** A sobrecarga é a possibilidade de criar vários funcionamentos de um método de acordo com os parâmetros passados.

A sobrescrita é a possibilidade de implementar um método já criado na superclasse, a fim de adequar o funcionamento à classe filha.

3. Utilizando sua IDE de preferência, implemente o estudo de caso a seguir:

### Sistema hospitalar

Crie um sistema hospitalar simplificado para realizar o cadastro de pacientes e equipe médica. Este sistema deve conter as classes a seguir: Pessoa, Paciente, Médico e Hospital. Observe o uso de composição e herança.

Informações e comportamentos que devem estar contidas em cada classe:

#### Pessoa:

- nome
- endereço
- idade
- cpf
- sexo
- Uma pessoa pode andar

#### Paciente (é uma Pessoa):

- doença;
- medicação

- Um paciente pode sentir dor
- Um paciente pode ter alta

#### Médico (é uma Pessoa):

- crm
- salário
- especialização
- Um médico pode dar plantão

#### Hospital:

- nome
- endereço

Adicionalmente, cada classe deve conter um método chamado `imprimirValores()`, que é responsável por imprimir os atributos da classe. Este método deve ser declarado em `Pessoa` e sobrescrito nas subclasses.

```
public class Hospital {
    String nome;
    String endereco;
    Hospital(String nome, String endereco){
        this.nome = nome;
        this.endereco = endereco;
    }

    public void imprimirValores(){
        System.out.println("Nome: "+this.nome);
        System.out.println("Endereco: "+this.endereco);
    }
}
```

```
public class Pessoa {
    String nome;
    String endereco;
    int idade;
    String cpf;
    String sexo;
    boolean podeAndar; // Uma pessoa pode andar
    Pessoa(){
    }

    Pessoa(String nome, String endereco, int idade, String cpf, String sexo,
boolean podeAndar){
        this.nome = nome;
        this.endereco = endereco;
        this.idade = idade;
        this.cpf = cpf;
        this.sexo = sexo;
        this.podeAndar = podeAndar;
    }

    public void imprimirValores(){
        System.out.println("Nome: "+ this.nome);
        System.out.println("Endereço: "+ this.endereco);
        System.out.println("Idade: "+ this.idade);
        System.out.println("CPF: "+ this.cpf);
    }
}
```

```

        System.out.println("Sexo: " + this.sexo);
        System.out.println("pode andar: " + ( (this.podeAndar) ? " sim" : "nao" ) );
    }
}

```

```

public class Paciente extends Pessoa {
    String doenca;
    String medicacao;
    boolean sentindoDor; // Um paciente pode sentir dor
    boolean alta; // Um paciente pode ter alta
    Paciente(){

    }

    Paciente(String nome, String endereco, int idade, String cpf, String sexo,
boolean podeAndar, String doenca, String medicacao, boolean sentindoDor, boolean
alta ){

        super(nome, endereco, idade, cpf, sexo, podeAndar);
        this.doenca = doenca;
        this.medicacao = medicacao;
        this.sentindoDor = sentindoDor;
        this.alta = alta;
    }

    public void imprimirValores(){
        System.out.println("Nome: " + this.nome);
        System.out.println("Endereço: " + this.endereco);
        System.out.println("Idade: " + this.idade);
        System.out.println("CPF: " + this.cpf);
        System.out.println("Sexo: " + this.sexo);
        System.out.println("pode andar: " + ((this.podeAndar) ? " sim" : "nao" ) );
        System.out.println("Doenca: " + this.doenca);
        System.out.println("Medicacao: " + this.medicacao);
        System.out.println("Sentindo dor: " + ((this.sentindoDor) ? " sim" : "nao")
);
    }
}

```

```

public class Medico extends Pessoa {
    String crm;
    double salario;
    String especializacao;
    boolean podeDarPlantao; // Um médico pode dar plantão
}

```

```

Medico() {

}

Medico(String nome, String endereco, int idade, String cpf, String sexo,
boolean podeAndar, String crm, double salario, String especializacao, boolean
podeDarPlantao ){
    super(nome, endereco, idade, cpf, sexo, podeAndar);
    this.crm = crm;
    this.salario = salario;
    this.especializacao = especializacao;
    this.podeDarPlantao = podeDarPlantao;
}

public void imprimirValores(){
    System.out.println("Nome: " + this.nome);
    System.out.println("Endereço: " + this.endereco);
    System.out.println("Idade: " + this.idade);
    System.out.println("CPF: " + this.cpf);
    System.out.println("Sexo: " + this.sexo);
    System.out.println("pode andar: " + ( (this.podeAndar) ? " sim" : "nao" ) );
    System.out.println("CRM: " + this.crm);
    System.out.println("Salario: " + this.salario);
    System.out.println("Especializacao: " + this.especializacao);
    System.out.println("Pode Dar Plantao: " + ( (this.podeDarPlantao) ? " sim" :
"nao" ) );
}
}

```

Boa sorte!

Prof. Igor.