Universidade do Trabalho Digital

PROGRAMAÇÃO ORIENTADA A OBJETOS

Herança, polimorfismo e interfaces

CENTEC
INSTITUTO CENTRO DE ENSINO TECNOLÓGICO

GOVERNO DO ESTADO DO CEARÁ
Secretaria da Ciência, Tecnologia e Educação Superior

Classe abstrata.

- Uma classe abstrata é uma super classe que não pode ser instanciada em nenhum objeto. Ela passa a ser uma super classe.

```java
public abstract class Pessoa {

    private String nome;
    private int idade;

    public Pessoa(String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
    }

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public int getIdade() {
        return idade;
    }
    public void setIdade(int idade) {
        this.idade = idade;
    }

}
```

Herança com extends

- A palavra-chave extends faz com que o objeto herde os métodos e atributos da superclasse.

- A classe só pode ser herdeira, pelo extends, de apenas uma super classe.

```
public class Cliente extends Pessoa{

    private String cadastro;

    public Cliente(String nome, int idade,String cadastro) {
        super(nome, idade);
        this.cadastro = cadastro;
    }

    public String getCadastro() {
        return this.cadastro;
    }
}
```

Classe exemplo: Cliente.

- Observe a classe Cliente.
- Aqui podemos ver os atributos e métodos getters e setters.

```java
public class Cliente {

    private String nome;
    private int idade;

    public String getNome() {
        return this.nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }
}
```

Implementando a herança.

```java
public class Cliente extends Pessoa{

    private String cadastro;

    public Cliente(String nome, int idade,String cadastro) {
        super(nome, idade);
        this.cadastro = cadastro;
    }

    public String getCadastro() {
        return this.cadastro;
    }
}
```

```java
package apostila;

public class Funcionario extends Pessoa{

    private String funcao;

    public Funcionario(String nome, int idade, String funcao) {
        super(nome, idade);

        this.funcao = funcao;
    }

    public String getFuncao() {
        return funcao;
    }

    public void setFuncao(String funcao) {
        this.funcao = funcao;
    }

}
```

Interfaces e herança múltipla.

```
public interface Funcionalidades {

    public void Cadasttro();

    public double Desconto();

}
```

- Interfaces são utilizadas na linguagem Java para fazer com que as classes tenham métodos específicos, uma vez que a classe tenha implementado a interface ela é obrigada a implementar e sobrescrever os métodos dela.

Polimorfismo

```java
public class Funcionario extends Pessoa implements Funcionalidades{

    private String funcao;

    public Funcionario(String nome, int idade, String funcao) {
        super(nome, idade);
        this.funcao = funcao;
    }

    public String getFuncao() {
        return funcao;
    }

    public void setFuncao(String funcao) {
        this.funcao = funcao;
    }

    public void Cadasttro() {
        System.out.println("Funcionário Cadastrado no RH");
    }

    public double Desconto() {

        double desconto = 15.0;
        return desconto;
```

```java
public class Cliente extends Pessoa implements Funcionalidades{

    private String cadastro;

    public Cliente(String nome, int idade,String cadastro) {
        super(nome, idade);
        this.cadastro = cadastro;
    }

    public String getCadastro() {
        return this.cadastro;
    }

    public void Cadasttro() {
        System.out.println("Cliente cadastrado no banco de dados");
    }

    public double Desconto() {
        Double desconto = 5.0;
        return desconto;
    }
}
```