ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

LINGUAGEM JAVA

Orientação a Objetos
Herança e Sobrescrita

Tópicos

- Herança
- Sobrescrita

ELABORATA
INFORMÁTICA

Herança

Herança

- A herança é uma técnica da programação orientada a objetos que possibilita uma classe herdar membros de uma outra classe.

Classe Pai e Classe Filha

- A classe que possui os membros que serão herdados é chamada de classe base, classe pai ou ainda classe principal. Já a classe que herdará os membros é chamada de classe filha ou classe secundária.

Membros Herdados

- Os membros herdados sempre serão os membros que utilizam o modificador de acesso public. Para a classe filha ter acesso aos atributos private da classe pai será necessário a utilização dos métodos getters e setters.

Cenário

- Foi identificado as seguintes classes, atributos e métodos para um possível sistema de controle bancário:

Classe: Conta Poupança
Atributos: Banco, Agência, Número, Saldo
Métodos: Depositar, Sacar e Consultar Saldo

Classe: Conta Corrente
Atributos: Banco, Agência, Número, Saldo, Limite, Gerente
Métodos: Depositar, Sacar e Consultar Saldo

Repetição de dados

- No levantamento das classes para o sistema de controle bancário houve uma repetição de alguns atributos e métodos nas classes Conta Poupança e Conta Corrente.

Eliminando a repetição de dados

- A herança pode ser utilizada para eliminar a redundância, facilitando a implementação das classes.

Exemplo

```
public class ContaBancaria {

  private String banco;
  private String agencia;
  private String numero;
  private double saldo;

  public void depositar(double valor) {
    this.saldo += valor;
  }

  public boolean sacar(double valor) {
    if (this.saldo >= valor) {
      this.saldo -= valor;
      return true;
    } else {
      return false;
    }
  }
  // Métodos getters e setters
}
```

Exemplo (cont.)

```
public class ContaPoupanca extends ContaBancaria {

  private Date dataAniversario;

  public ContaPoupanca() {
    this.dataAniversario = new Date();
  }

  // Métodos getters e setters

}
```

Exemplo (cont.)

```
public class ContaCorrente extends ContaBancaria {

  private int limite;
  private String gerente;

  // Métodos getters e setters

}
```

Exercício 01

- Dado o levantamento inicial abaixo, aplicar a herança de tal modo que nenhum atributo fique repetido nas classes.

Classe Cliente: Nome, Data nascimento, Gênero, CPF, RG, Telefone, Email

Classe Funcionario: Nome, Data nascimento, Gênero, CPF, RG, Telefone, Email, Salário

Classe Fornecedor: Razão Social, Nome Fantasia, CNPJ, IE, Telefone, Email

Exercício 02

- Dado o levantamento inicial abaixo, aplicar a herança de tal modo que nenhum atributo fique repetido nas classes.

Classe CD: Titulo, Músico, Gravadora, Ano Lançamento, Gênero Musical

Classe DVD: Título, Músico, Gravadora, Ano Lançamento, Gênero Musical

Classe Livro: Título, Subtítulo, Autor, Editora, Ano Lançamento, Gênero Literário, Páginas

Classe Revista: Título, Subtítulo, Editora, Ano, Número

ELABORATA INFORMÁTICA

Sobrescrita

Sobrescrita

- A sobrescrita de métodos é um recurso da programação orientada a objetos que permite um método na classe filha ter o seu comportamento alterado.

Exemplo

```
public class ContaCorrente extends ContaBancaria {

  private int limite;
  private String gerente;

  @Override
  public boolean sacar(double valor) {
    if (this.getSaldo() + this.limite >= valor) {
      this.saldo -= valor;
      return true;
    } else {
      return false;
    }
  }

  // Métodos getters e setters

}
```

Modificador de acesso 'protected'

- O modificador de acesso protected permite que a classe filha acesse diretamento os atributos da classe pai sem a necessidade do uso dos getters e setters.

Exemplo

```
public class ContaBancaria {
  protected String banco;
  protected String agencia;
  protected String numero;
  protected double saldo;
  ...
}
```

Exemplo (cont.)

```
public class ContaCorrente extends ContaBancaria {

  private int limite;
  private String gerente;

  @Override
  public boolean sacar(double valor) {
    if (this.saldo + this.limite >= valor) {
      this.saldo -= valor;
      return true;
    } else {
      return false;
    }
  }

  // Métodos getters e setters

}
```

Exercício 01

- Criar um programa para reajustar salários de funcionários.

OBRIGADO!

ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

www.**elaborata**.com.br

Horário de Atendimento Comercial
Segunda à sexta – das 9:00h às 19:30h e
Sábado - das 8:00h às 15:00h.

Rua Monsenhor Celso, 256 - 1º Andar
Centro - Curitiba - PR

41.3324.0015
41.99828.2468

cursos@elaborata.com.br