Lab 8 – Interfaces e Classes Abstratas

### **Interface** é uma característica da Linguagem de Programação Java que permite flexibilidade e agilidade no desenvolvimento de aplicações. Neste laboratório você irá explorar o conceito de **interfaces** e classes **abstratas**. Ressaltamos que os exemplos deste laboratório dependem de outras listagens de código desenvolvidas nos laboratórios anteriores.

Sugerimos que estes exemplos sejam feitos com uso da IDE Eclipse.

**Duração prevista: 50 minutos**

***Exercícios***

**Exercício 1**: Classes Abstratas (25 minutos)

**Exercício 2**: Interface como um tipo (25 minutos)

# *Exercício 1 – Classes Abstratas*

### As classes do projeto Banco precisam de um padrão de comportamento que identifique que instancias dessas classes são referentes ao projeto, será criada uma entidade que irá ser a classe pai de todas as classes do projeto.

1. Crie uma classe **EntidadeBanco.java** abstrata, nesta crie um método que retorne o identificador da classe, como na **Listagem-7.1**.

**public abstract class** EntidadeBanco {

**public abstract** Long getIdentificador();

}

*Listagem 7.1 – Classe EntidadeBanco.java*

1. Agora modifique as classes **Pessoa.java**, **Conta.java** e **Transacao.java,** de forma que essas classes se tornem sub-classe de EntidadeBanco.java.

**Observação:** Como a classe **EntidadeBanco.java** é abstrata é contém um método abstrato **getIdentificador(),** esta te forçará a criar um atributo **identificador** do tipo Long e implementar os métodos **Get** e **Set**. Este procedimento será feito em todas as classes modificadas.

**private** Long identificador;

@Override

**public** Long getIdentificador() {

**return** identificador;

}

**public void** setIdentificador(Long identificador) {

**this**.identificador = identificador;

}

# *Exercício 2 – Interface como um tipo*

1. Crie a **Interface Entidade.java** como definida abaixo:

**public interface** Entidade {

Long getIdentificador();

***}***

*Listagem 7.2 – Classe Entidade.java*

1. Implemente a classe **EntidadeBanco.java** de forma que implemente essa interface entidade.

**public abstract class** EntidadeBanco **implements** Entidade{...}

1. Crie a **Interface IExtrato.java** como definida abaixo:

**public interface** IExtrato {

/\*\*

\* Formata o movimento como string.

\*/

**public** String formatar() ;

*}*

Listagem 7.3 – Interface Extrato

1. Agora crie a classe **ExtratoTXT.java** que implementa a **interface IExtrato.java** e implemente o método **formartar()** conforme abaixo:

**import** java.util.Iterator;

**public class** ExtratoTXT **implements** IExtrato {

**protected** Conta conta;

**public** ExtratoTXT(Conta conta) {

**this**.conta = conta;

}

**public** String formatar() {

String newLine = System.*getProperty*("line.separator");

String resultado = "Extrato de conta " + newLine; resultado += String.*format*("%-20.20s", "Data") + " "

+ String.*format*("%7.7s", "Debito") + " "

+ String.*format*("%7.7s", "Credito") + " "

+ String.*format*("%15.15s", "Valor") + " "

+ String.*format*("%s", "Descricao") + newLine; Iterator it = conta.getMovimento().iterator(); **while** (it.hasNext()) {

Transacao t = (Transacao) it.next();

**if**(t.getTipoTransacao() == EnumTipoTransacao.*TRANSFERENCIA*){

resultado += String.*format*("%-20.20s", UtilData.*DDMMAAAAHHMM*(t.get-

Data()))

+ " "

+ String.*format*("%7d", t.getContaDebito().getNumero())

+ " "

+ String.*format*("%7d", t.getContaCredito().getNumero())

+ " " + String.*format*("%15.15s", t.getValor()) + " "

+ String.*format*("%s", t.getDescricao()) + newLine;

}

}

**return** resultado;

}

}

Listagem 7.4 – ExtratoTXT.java

1. Crie a classe **ExtratoHTML.java** que implementa a **interface IExtrato.java**.

**import** java.util.Iterator;

**public class** ExtratoHTML **implements** IExtrato {

**protected** Conta conta;

**public** ExtratoHTML( Conta conta ) {

**this**.conta = conta;

}

**public** String formatar() {

String newLine = System.*getProperty*("line.separator"); String resultado = "<html>" + newLine;

resultado += "<head>" + newLine;

resultado += "<title>Extrato de Conta</title>" + newLine; resultado += "<style type=\"text/css\">" + newLine; resultado += "<!--" + newLine;

resultado += "body { font-family: Verdana, Arial,Helvetica," + "sans-serif; font-weight: normal; font-variant: normal}" + newLine;

resultado += ".clsIndex { }" + newLine;

resultado += ".clsTitle { background-color: #CCCCCC;" + "text-align: center }" + new-

Line;

resultado += "td { font-size: 9pt; font-family: Verdana, Arial," + "Helvetica, sans-serif;

background-color: #EEEEEE}" + newLine;

resultado += "-->" + newLine; resultado += "</style>" + newLine; resultado += "</head>" + newLine; resultado += "<body>" + newLine;

resultado += "<h2>Extrato de conta</h2>" + newLine; resultado += "<TABLE CLASS=\"clsIndex\">" + newLine; resultado += "<tr>" + newLine;

resultado += "<TD CLASS=\"clsTitle\"><B>Data</B></TD>" + newLine; resultado += "<TD CLASS=\"clsTitle\"><B>Debito</B></TD>" + newLine; resultado += "<TD CLASS=\"clsTitle\"><B>Credito</B></TD>" + newLine; resultado += "<TD CLASS=\"clsTitle\"><B>Valor</B></TD>" + newLine; resultado += "<TD CLASS=\"clsTitle\"><B>Descricao</B></TD>" + newLine; resultado += "</tr>" + newLine;

Iterator it = conta.getMovimento().iterator();

**while** (it.hasNext()) {

Transacao t = (Transacao) it.next();

**if**(t.getTipoTransacao() == EnumTipoTransacao.*TRANSFERENCIA*){ resultado += "<tr>" + newLine;

resultado += "<TD align=\"left\">" + UtilData.*DDMMAAAAHHMM*(t.getDa-

ta()) + "</TD>" + newLine;

resultado += "<TD align=\"right\">" + t.getContaDebito().getNumero() +

"</TD>" + newLine; Line;

}

resultado += "<TD align=\"right\">" + t.getContaCredito().getNumero() +

resultado += "<TD align=\"right\">" + t.getValor() + "</TD>" + newLine; resultado += "<TD align=\"left\">" + t.getDescricao() + "</TD>" + new-

resultado += "</tr>" + newLine;

}

resultado += "</table>" + newLine; resultado += "</body>" + newLine; resultado += "</html>" + newLine; **return** resultado;

}

}

Listagem 7.5 – ExtratoHTML.java

1. Crie a classe **ExtratoContaCorrente.java** para imprimir as movimentações da classe **Conta.java**.

**public class** ExtratoContaCorrente {

**public static void** main(String[] args) {

// Cria uma instância de ContaService onde está presente as operações para Objeto Conta ContaService operacoesConta = **new** ContaService();

Conta correntista1 = **new** Conta("Aluno", 1001); Conta correntista2 = **new** Conta("Professor", 2002);

// faz deposito operacoesConta.depositar(correntista1, 1000);

// faz transferência de correntista1 para correntista2 e salva em memoria a transação operacoesConta.transferir(correntista1, 450.00, correntista2);

// faz transferência de correntista1 para correntista2 e salva em memoria a transação operacoesConta.transferir(correntista2, 50.00, correntista1);

//IExtrato movimento = **new** ExtratoTXT(correntista1);

//System.*out*.println(movimento.formatar());

IExtrato movimento1 = **new** ExtratoHTML(correntista1); System.*out*.println(movimento1.formatar());

}

}

Listagem 7.6 – Imprime extrato Conta corrente.

1. Modifique a **Listagem-7.6**, para usando **ExtratoHTML.java**, **Listagem-7.5**, no lugar de

## ExtratoTXT.java.

### Melhore o extrato imprimindo no cabeçalho o nome do cliente, numero da conta e data de impressão.

resultado += "Titular: " + conta.getTitular() + " Conta: " + conta.getNumero() + " "+ newLine; resultado += "Data de Impressao: " + UtilData.*DDMMAAAAHHMM*(UtilData.*data*()) + " "+ newLine;