

Orientação a Objetos 1

Paramétrico

Prof. Dr. Vinícius Camargo Andrade

vcandrade@utfpr.edu.br

Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná

Exemplo

Exemplo

*Em casos que se necessita armazenar valores em listas, como por exemplo, arrays, e **não se sabe** de modo antecipado **quantos índices serão necessários** nessa lista, pode-se utilizar, por exemplo, a interface **List** e a classe **ArrayList** que a implementa.*

Exemplo

Neste caso, a lista cresce conforme a necessidade.

Class ArrayList

- *add (elemento)* – adiciona um elemento à lista;
- *get (índice)* – retorna o elemento do índice específico;
- *remove (índice)* – remove o elemento do índice específico;
- *remove (objeto)* – remove a primeira ocorrência do elemento especificado;
- *size ()* – retorna o número de elementos da lista;
- *clear ()* – remove todos os elementos da lista.

Class ArrayList

```
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         List listaGenerica = new ArrayList();
11
12         listaGenerica.add("João da Silva");
13         listaGenerica.add("Maria de Oliveira");
14         listaGenerica.add("Pedro Carvalho");
15         listaGenerica.add("José Souza");
16
17         for(int i = 0; i < listaGenerica.size(); i++) {
18
19             System.out.println(listaGenerica.get(i));
20         }
21     }
22 }
```

<

Console Problems Debug Shell

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v202008

João da Silva
Maria de Oliveira
Pedro Carvalho
José Souza

Class ArrayList

```
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         List listaGenerica = new ArrayList();
11
12         listaGenerica.add("João da Silva");
13         listaGenerica.add("Maria de Oliveira");
14         listaGenerica.add("Pedro Carvalho");
15         listaGenerica.add("José Souza");
16
17         for(int i = 0; i < listaGenerica.size(); i++) {
18
19             System.out.println(listaGenerica.get(i));
20         }
21     }
22 }
```

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v202008

João da Silva
Maria de Oliveira
Pedro Carvalho
José Souza

Class ArrayList

Neste exemplo, são adicionados à lista 4 nomes de pessoas (strings), porém, o objeto da classe ArrayList recebe qualquer tipo de elemento, seja ele primitivo ou objeto.

Class ArrayList

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List listaGenerica = new ArrayList();
14
15         listaGenerica.add("João da Silva"); // String
16         listaGenerica.add(1); // int
17         listaGenerica.add(25.99); // double
18         listaGenerica.add(v1); // objeto de Veiculo
19         listaGenerica.add(p1); // objeto de Pessoa
20
21         for(int i = 0; i < listaGenerica.size(); i++) {
22
23             System.out.println(listaGenerica.get(i));
24         }
25     }
26 }
```

Console Problems Debug Shell

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\

João da Silva

1

25.99

Veiculo [modelo=Cruze, marca=Chevrolet, placa=ABC-1234, valorDiaria=150.0]

Pessoa [nome=Joaquim Rodrigues, idade=35, peso=75.8]

Problema

Problema

*Como são armazenados vários tipos de dados, ao recuperar um elemento por meio do método `get()`, o mesmo retorna um **objeto genérico**, visto que o interpretador não tem condições de “saber” o que está armazenado naquele índice, **é necessário fazer um cast**.*

Problema

```
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List listaGenerica = new ArrayList();
14
15         listaGenerica.add("João da Silva"); // String
16         listaGenerica.add(1);                // int
17         listaGenerica.add(25.99);            // double
18         listaGenerica.add(v1);               // objeto de Veiculo
19         listaGenerica.add(p1);               // objeto de Pessoa
20
21         String nome = (String) listaGenerica.get(0);
22         Integer numeroInteiro = (Integer) listaGenerica.get(1);
23         Double numeroDecimal = (Double) listaGenerica.get(2);
24         Veiculo objetoVeiculo = (Veiculo) listaGenerica.get(3);
25         Pessoa objetoPessoa = (Pessoa) listaGenerica.get(4);
26     }
27 }
```

Paramétrico

Polimorfismo

Paramétrico

Por si só, a definição de um elemento é incompleta.

Neste contexto, é necessário parametrizar o tipo para que ele exista.

Paramétrico

*A partir do Java 5.0, pode-se utilizar o recurso de **Generics** para **restringir as listas** a um determinado tipo primitivo ou objeto.*

```
List<restrição> nomeLista = new ArrayList<restrição>();
```

Paramétrico

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List<String> listaGenerica = new ArrayList<String>();
14
15         listaGenerica.add("João da Silva");           // String
16         listaGenerica.add("Maria de Oliveira");       // String
17         listaGenerica.add("José Rodrigues");          // String
18         listaGenerica.add(1);                          // int
19         listaGenerica.add(25.99);                     // double
20         listaGenerica.add(v1);                        // objeto de Veiculo
21         listaGenerica.add(p1);                        // objeto de Pessoa
22
23         String nome1 = listaGenerica.get(0);
24         String nome2 = listaGenerica.get(1);
25         String nome3 = listaGenerica.get(2);
26     }
27 }
28
```


Paramétrico

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List<String> listaGenerica = new ArrayList<String>();
14
15         listaGenerica.add("João da Silva");           // String
16         listaGenerica.add("Maria de Oliveira");       // String
17         listaGenerica.add("José Rodrigues");          // String
18         listaGenerica.add(1);                          // int
19         listaGenerica.add(25.99);                     // double
20         listaGenerica.add(v1);                        // objeto de Veiculo
21         listaGenerica.add(p1);                        // objeto de Pessoa
22
23         String nome1 = listaGenerica.get(0);
24         String nome2 = listaGenerica.get(1);
25         String nome3 = listaGenerica.get(2);
26     }
27 }
28
```

Paramétrico

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List<String> listaGenerica = new ArrayList<String>();
14
15         listaGenerica.add("João da Silva");           // String
16         listaGenerica.add("Maria de Oliveira");       // String
17         listaGenerica.add("José Rodrigues");          // String
18         listaGenerica.add(1);                          // int
19         listaGenerica.add(25.99);                     // double
20         listaGenerica.add(v1);                        // objeto de Veiculo
21         listaGenerica.add(p1);                        // objeto de Pessoa
22
23         String nome1 = listaGenerica.get(0);
24         String nome2 = listaGenerica.get(1);
25         String nome3 = listaGenerica.get(2);
26     }
27 }
28
```

Paramétrico

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ArrayListTeste {
7
8     public static void main(String[] args) {
9
10         Veiculo v1 = new Veiculo("Cruze", "Chevrolet", "ABC-1234", 150.00);
11         Pessoa p1 = new Pessoa("Joaquim Rodrigues", 35, 75.8);
12
13         List<String> listaGenerica = new ArrayList<String>();
14
15         listaGenerica.add("João da Silva");           // String
16         listaGenerica.add("Maria de Oliveira");       // String
17         listaGenerica.add("José Rodrigues");          // String
18         listaGenerica.add(1);                          // int
19         listaGenerica.add(25.99);                     // double
20         listaGenerica.add(v1);                        // objeto de Veiculo
21         listaGenerica.add(p1);                        // objeto de Pessoa
22
23         String nome1 = listaGenerica.get(0);
24         String nome2 = listaGenerica.get(1);
25         String nome3 = listaGenerica.get(2);
26     }
27 }
28
```

Collections

Linguagem Java

Collections

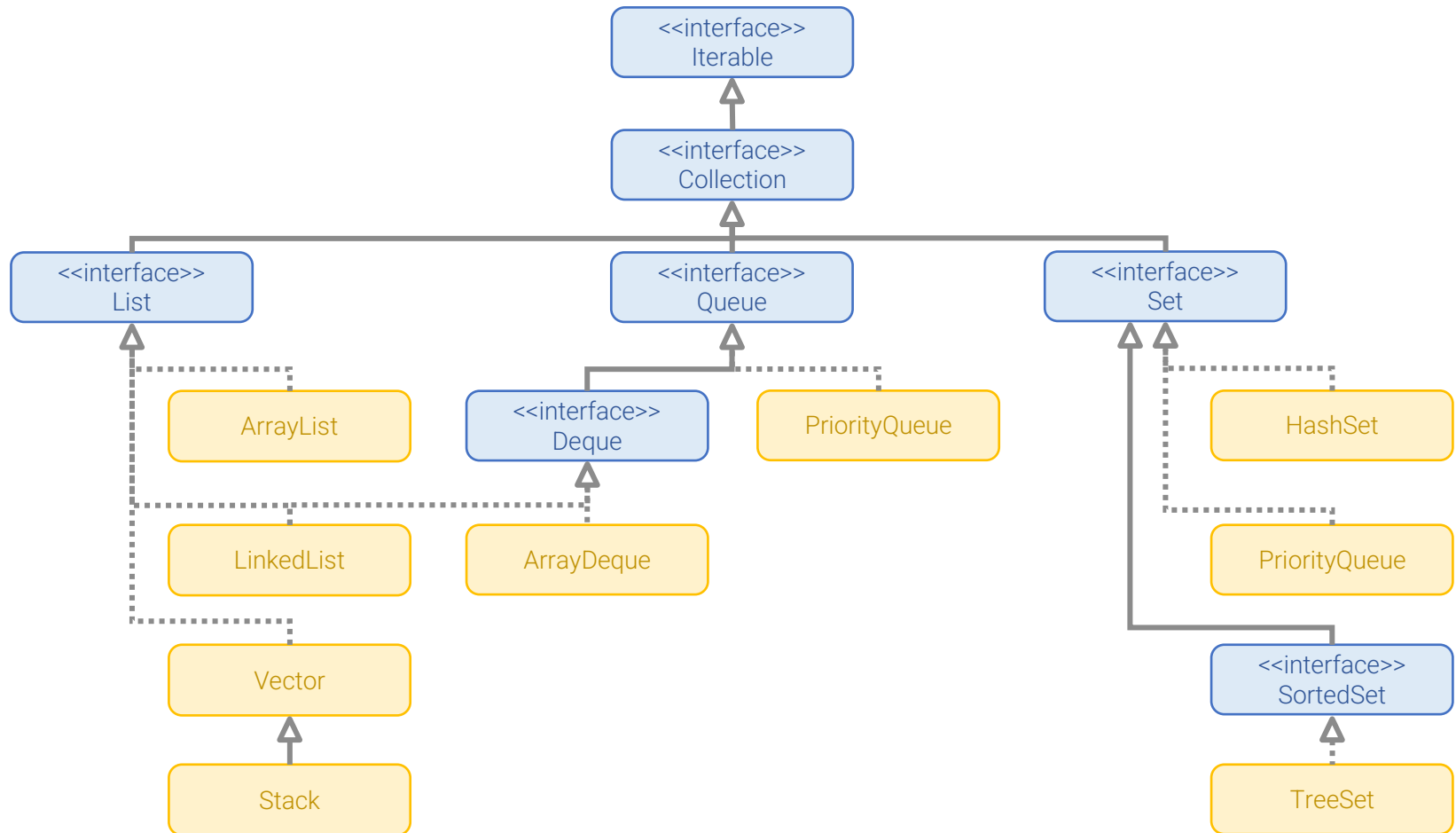
*Uma coleção (*Collection*) é um objeto que pode agrupar vários elementos. Como por exemplo: *fila*, *pilha*, *lista*, *conjunto*.*

Collections

Java possui uma arquitetura para representar e manipular as coleções:

- *Interfaces:* permite manipular coleções de forma independente de suas implementações;
- *Implementações:* implementam uma ou mais interfaces;
- *Algoritmos:* métodos que realizam operações (sort, reverse, entre outros.) sobre as coleções.

Hierarquia Collections



Interface List

- *Coleção ordenada;*
- *Índice do primeiro elemento é **zero**;*
- *Pode conter valores **duplicados**;*
- *Manipulação dos elementos baseada no **índice numérico** da lista. É possível **percorrer sequencialmente** a lista, além de efetuar busca e remoções)*

Interface Queue

- Utilizada para manter uma *fila* de elementos;
- Define métodos para coleções do tipo *FIFO* (*First-in-First-out*).

Interface Set

- *Collection de elementos únicos (não duplicados);*
- *Implementações de Set:*
 - *HashSet: armazena elementos em uma tabela hash;*
 - *TreeSet: armazena elementos em uma árvore binária.*

Percorrer Coleções

Percorrer Coleções

*Quando utiliza-se coleções, há uma maneira facilitada para declarar **laços de repetições**, chamada de estrutura **foreach**.*

Percorrer Coleções

A estrutura foreach não possui um contador explícito como laços de repetições padrão, porém define a lógica para todos os itens da coleção.

```

8 public static void main(String[] args) {
9
10     Pessoa p1 = new Pessoa("João da Silva", 30, 75.8);
11     Pessoa p2 = new Pessoa("Maria de Oliveira", 27, 51.2);
12     Pessoa p3 = new Pessoa("José Rodrigues", 42, 68.1);
13     Pessoa p4 = new Pessoa("Joaquim Rodrigues", 39, 92.7);
14
15     List<Pessoa> listaPessoas = new ArrayList<Pessoa>();
16
17     listaPessoas.add(p1);
18     listaPessoas.add(p2);
19     listaPessoas.add(p3);
20     listaPessoas.add(p4);
21
22     for (int i = 0; i < listaPessoas.size(); i++) {
23
24         Pessoa pessoa = listaPessoas.get(i);
25
26         System.out.println("Nome: " + pessoa.getNome());
27         System.out.println("Idade: " + pessoa.getIdade());
28         System.out.println("Peso: " + pessoa.getPeso());
29         System.out.println("-----");
30     }

```

Problems @ Javadoc Declaration Console

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

```

Nome: João da Silva
Idade: 30
Peso: 75.8
-----
Nome: Maria de Oliveira
Idade: 27
Peso: 51.2
-----
Nome: José Rodrigues
Idade: 42
Peso: 68.1
-----
Nome: Joaquim Rodrigues
Idade: 39
Peso: 92.7

```

```

8 public static void main(String[] args) {
9
10     Pessoa p1 = new Pessoa("João da Silva", 30, 75.8);
11     Pessoa p2 = new Pessoa("Maria de Oliveira", 27, 51.2);
12     Pessoa p3 = new Pessoa("José Rodrigues", 42, 68.1);
13     Pessoa p4 = new Pessoa("Joaquim Rodrigues", 39, 92.7);
14
15     List<Pessoa> listaPessoas = new ArrayList<Pessoa>();
16
17     listaPessoas.add(p1);
18     listaPessoas.add(p2);
19     listaPessoas.add(p3);
20     listaPessoas.add(p4);
21
22     for (Pessoa pessoa : listaPessoas) {
23
24         System.out.println("Nome: " + pessoa.getNome());
25         System.out.println("Idade: " + pessoa.getIdade());
26         System.out.println("Peso: " + pessoa.getPeso());
27         System.out.println("-----");
28     }
29 }

```

Problems @ Javadoc Declaration Console

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

```

Nome: João da Silva
Idade: 30
Peso: 75.8
-----
Nome: Maria de Oliveira
Idade: 27
Peso: 51.2
-----
Nome: José Rodrigues
Idade: 42
Peso: 68.1
-----
Nome: Joaquim Rodrigues
Idade: 39
Peso: 92.7

```

```

8 public static void main(String[] args) {
9
10     Pessoa p1 = new Pessoa("João da Silva", 30, 75.8);
11     Pessoa p2 = new Pessoa("Maria de Oliveira", 27, 51.2);
12     Pessoa p3 = new Pessoa("José Rodrigues", 42, 68.1);
13     Pessoa p4 = new Pessoa("Joaquim Rodrigues", 39, 92.7);
14
15     List<Pessoa> listaPessoas = new ArrayList<Pessoa>();
16
17     listaPessoas.add(p1);
18     listaPessoas.add(p2);
19     listaPessoas.add(p3);
20     listaPessoas.add(p4);
21
22     for (int i = 0; i < listaPessoas.size(); i++) {
23
24         Pessoa pessoa = listaPessoas.get(i);
25
26         System.out.println("Nome: " + pessoa.getNome());
27         System.out.println("Idade: " + pessoa.getIdade());
28         System.out.println("Peso: " + pessoa.getPeso());
29         System.out.println("-----");
30     }

```

Problems @ Javadoc Declaration Console

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\jre\bin\java.exe

```

Nome: João da Silva
Idade: 30
Peso: 75.8
-----
Nome: Maria de Oliveira
Idade: 27
Peso: 51.2
-----
Nome: José Rodrigues
Idade: 42
Peso: 68.1
-----
Nome: Joaquim Rodrigues
Idade: 39
Peso: 92.7

```

```

8 public static void main(String[] args) {
9
10     Pessoa p1 = new Pessoa("João da Silva", 30, 75.8);
11     Pessoa p2 = new Pessoa("Maria de Oliveira", 27, 51.2);
12     Pessoa p3 = new Pessoa("José Rodrigues", 42, 68.1);
13     Pessoa p4 = new Pessoa("Joaquim Rodrigues", 39, 92.7);
14
15     List<Pessoa> listaPessoas = new ArrayList<Pessoa>();
16
17     listaPessoas.add(p1);
18     listaPessoas.add(p2);
19     listaPessoas.add(p3);
20     listaPessoas.add(p4);
21
22     for (Pessoa pessoa : listaPessoas) {
23
24         System.out.println("Nome: " + pessoa.getNome());
25         System.out.println("Idade: " + pessoa.getIdade());
26         System.out.println("Peso: " + pessoa.getPeso());
27         System.out.println("-----");
28     }
29

```

Problems @ Javadoc Declaration Console

<terminated> ArrayListTeste [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\jre\bin\java.exe

```

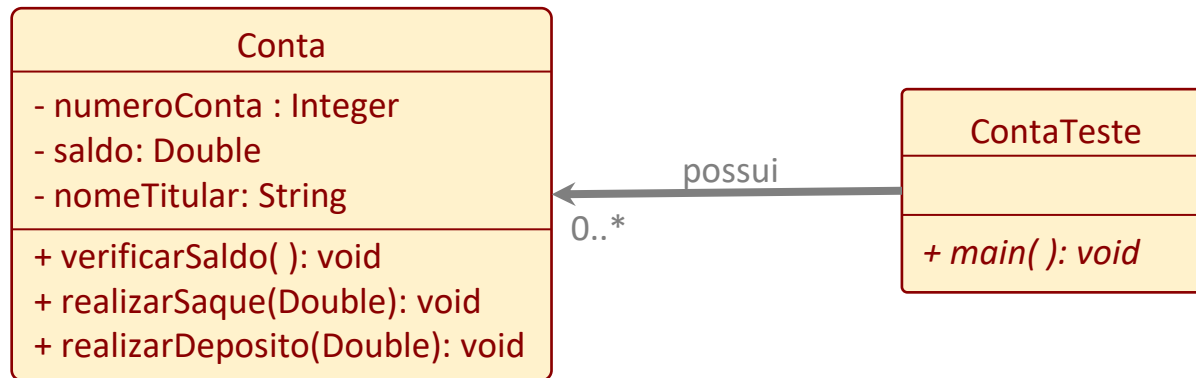
Nome: João da Silva
Idade: 30
Peso: 75.8
-----
Nome: Maria de Oliveira
Idade: 27
Peso: 51.2
-----
Nome: José Rodrigues
Idade: 42
Peso: 68.1
-----
Nome: Joaquim Rodrigues
Idade: 39
Peso: 92.7

```

Contas Bancárias

Cenário de Aplicação

Contas Bancárias



Contas Bancárias

```
3 public class Conta {
4
5     private Integer numeroConta;
6     private Double saldo;
7     private String nomeTitular;
8
9     public Conta(Integer numeroConta, String nomeTitular) {
10
11         this.numeroConta = numeroConta;
12         this.saldo = 0.00;
13         this.nomeTitular = nomeTitular;
14     }
15
16     public void verificarSaldo() {
17
18         System.out.println("Conta: " + this.numeroConta + ".\n"
19                             + "Saldo: R$" + this.saldo + "\n");
20     }
21
22     public void realizarSaque(Double valor) {
23
24         if(valor <= this.saldo) {
25
26             this.saldo -= valor;
27             System.out.println("Saque de R$" + valor + " realizado com sucesso.\n"
28                                 + "Conta: " + this.numeroConta + "\n");
29
30         } else {
31
32             System.out.println("Saldo insuficiente.\n"
33                                 + "Conta: " + this.numeroConta + "\n");
34         }
35     }
36
37     public void realizarDeposito(Double valor) {
```

```
3 public class ContaTeste {
4
5     public static void main(String[] args) {
6
7         Conta c1 = new Conta(123, "José da Silva");
8         Conta c2 = new Conta(987, "Maria de Oliveira");
9
10        c1.verificarSaldo();
11        c1.realizarDeposito(1000.00);
12        c1.realizarSaque(700.00);
13        c1.verificarSaldo();
14
15        c2.verificarSaldo();
16        c2.realizarSaque(100.00);
17    }
18 }
```

Problems @ Javadoc Declaration Console

<terminated> ContaTeste (1) [Java Application] C:\Users\vinic\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre

Conta: 123.
Saldo: R\$0.0

Conta: 123
Depósito de R\$1000.0 realizado com sucesso.

Saque de R\$700.0 realizado com sucesso.
Conta: 123

Conta: 123.
Saldo: R\$300.0

Conta: 987.
Saldo: R\$0.0

Saldo insuficiente.
Conta: 987

Problema

Problema

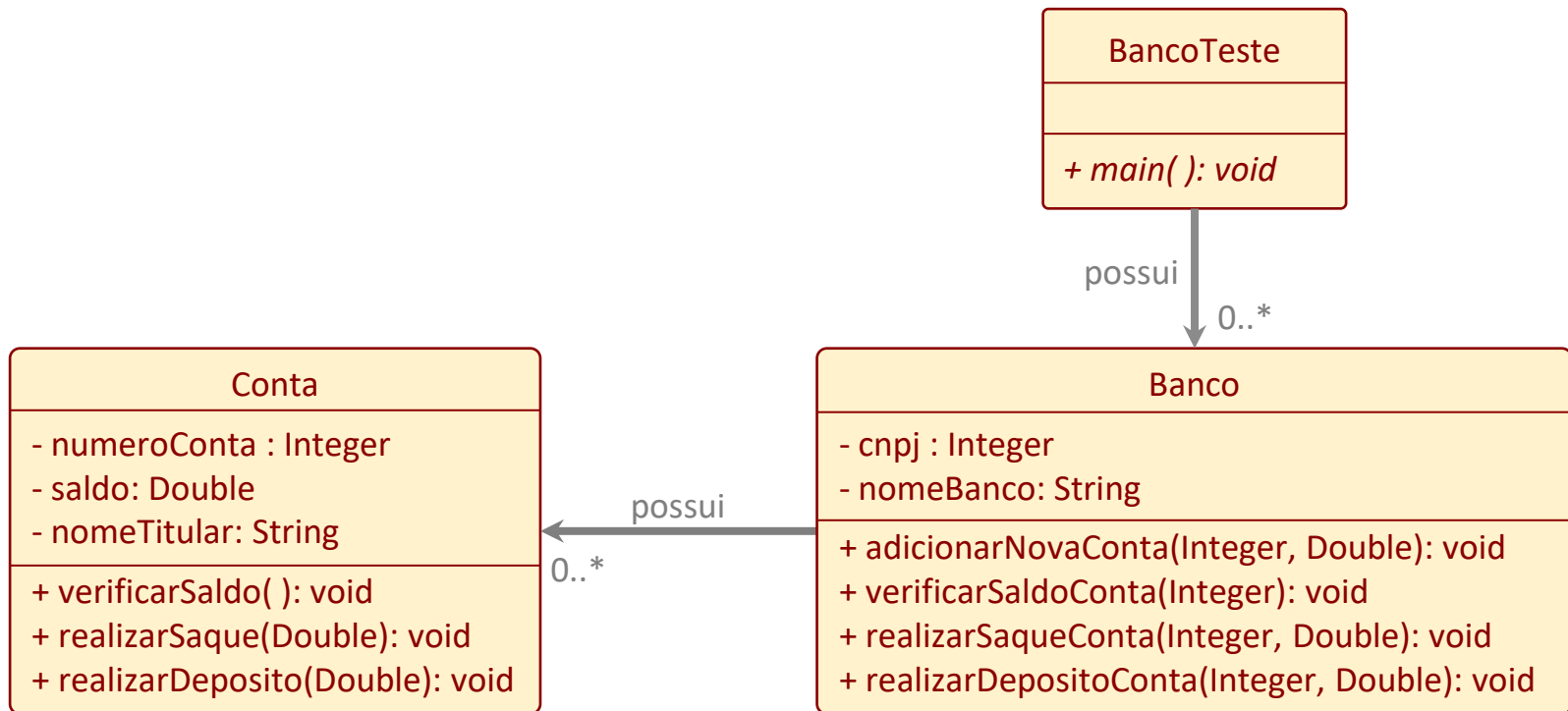
*A classe de teste contém **todas as instâncias de contas**, porém ela é apenas um classe de teste e não faz parte da regra de negócio da aplicação.*

Problema

*Então, qual é a entidade que **controla/manipula** todas as **N** contas?*

Solução

Contas Bancárias



Contas Bancárias

```
2
3 public class BancoTeste {
4
5     public static void main(String[] args) {
6
7         Banco b1 = new Banco(159357, "NewBank S.A.");
8
9         b1.adicionarNovaConta(123, "José da Silva");
10        b1.adicionarNovaConta(987, "Maria de Oliveira");
11
12        b1.verificarSaldoConta(123);
13        b1.realizarDepositoConta(123, 1000.00);
14        b1.realizarSaqueConta(123, 700.00);
15        b1.verificarSaldoConta(123);
16
17        b1.verificarSaldoConta(987);
18        b1.realizarSaqueConta(987, 100.00);
19    }
20 }
21
```



```

6 public class Banco {
7
8     private Integer cnpj;
9     private String nomeBanco;
10    private List<Conta> contas;
11
12    public Banco(Integer cnpj, String nomeBanco) {
13
14        this.cnpj = cnpj;
15        this.nomeBanco = nomeBanco;
16        this.contas = new ArrayList<Conta>();
17    }
18
19    public void adicionarNovaConta(int numeroConta, String nomeTitular) {
20
21        Conta conta = new Conta(numeroConta, nomeTitular);
22        this.contas.add(conta);
23    }
24
25    public void verificarSaldoConta(Integer numeroConta) {
26
27        for (Conta conta : contas) {
28
29            if (conta.getNumeroConta().equals(numeroConta)) {
30
31                conta.verificarSaldo();
32                break;
33            }
34        }
35    }
36
37    public void realizarSaqueConta(Integer numeroConta, Double valor) {
38
39        for (Conta conta : contas) {
40
41            if (conta.getNumeroConta().equals(numeroConta)) {
42
43                conta.realizarSaque(valor);
44                break;
45            }
46        }
47    }
48
49    public void realizarDepositoConta(Integer numeroConta, Double valor) {
50
51        for (Conta conta : contas) {
52
53            if (conta.getNumeroConta().equals(numeroConta)) {
54
55                conta.realizarDeposito(valor);
56                break;
57            }
58        }
59    }

```

```

3 public class Conta {
4
5     private Integer numeroConta;
6     private Double saldo;
7     private String nomeTitular;
8
9     public Conta(Integer numeroConta, String nomeTitular) {
10
11         this.numeroConta = numeroConta;
12         this.saldo = 0.00;
13         this.nomeTitular = nomeTitular;
14     }
15
16     public void verificarSaldo() {
17
18         System.out.println("Conta: " + this.numeroConta + ".\n"
19             + "Saldo: R$" + this.saldo + "\n");
20     }
21
22     public void realizarSaque(Double valor) {
23
24         if(valor <= this.saldo) {
25
26             this.saldo -= valor;
27             System.out.println("Conta: " + this.numeroConta + "\n"
28                 + "Saque de R$" + valor + " realizado com sucesso.\n");
29
30         } else {
31
32             System.out.println("Conta: " + this.numeroConta + "\n"
33                 + "Saldo insuficiente.\n");
34         }
35     }
36
37     public void realizarDeposito(Double valor) {
38
39         this.saldo += valor;
40         System.out.println("Conta: " + this.numeroConta + "\n"
41             + "Depósito de R$" + valor + " realizado com sucesso.\n");
42     }
43
44     public Integer getNumeroConta() {
45         return numeroConta;
46     }
47
48     public Double getSaldo() {
49         return saldo;
50     }
51
52     public String getNomeTitular() {
53         return nomeTitular;
54     }
55 }

```

Exercícios

Exercício 1

*Desenvolva um sistema que o usuário informe **cinco valores inteiros**. Armazene cada valor em um **ArrayList**.*

*Ao final, faça uma busca no ArrayList exibindo a **soma de todos os valores** e a **média aritmética** deles.*

Exercício 2

Desenvolva um código que insere **500 mil números** em um **ArrayList** e, só após todos os valores inseridos, imprima-os em console. Utilize um método da classe **System** para cronometrar o tempo gasto. Utilize o modelo:

```
long inicio = System.currentTimeMillis();  
  
// inserir 500 mil números no ArrayList  
  
// percorrer todo o ArrayList e imprimir os valores de cada índice  
  
long fim = System.currentTimeMillis();  
long tempo = fim - inicio;  
  
System.out.println("Tempo gasto: " + tempo + "ms");
```