

```
1 package model;
2
3 import java.math.BigDecimal;
4 import java.util.Random;
5
6 public class ContaBancaria {
7     private String nome;
8     private String cpf;
9     private String agencia;
10    private String numeroConta;
11    private BigDecimal saldo;
12
13    public ContaBancaria(String nome, String cpf) {
14        this.nome = nome;
15        this.cpf = cpf;
16        this.agencia = gerarAgencia();
17        this.numeroConta = gerarNumeroConta();
18        this.saldo = BigDecimal.ZERO;
19    }
20
21    private String gerarNumeroConta() {
22        Random random = new Random();
23        return String.valueOf(random.nextInt(99999)) + "-" + random.nextInt(9);
24    }
25
26    private String gerarAgencia() {
27        Random random = new Random();
28        return String.valueOf(random.nextInt(9999));
29    }
30
31    public void depositar(BigDecimal valor) {
32        this.saldo = this.saldo.add(valor);
33    }
34
35    public void sacar(BigDecimal valor) {
36        this.saldo = this.saldo.subtract(valor);
37    }
38
39    public String getNome() {
40        return this.nome;
41    }
42
43    public String toString() {
44        return "[Nome: " + this.nome + ", CPF: " + this.cpf + ", Agência: " + this.agencia +
45        ", Número da Conta: " + this.numeroConta + ", Saldo: R$" + this.saldo + "]\n";
46    }
47
48 package model;
49
50 import java.util.Arrays;
51
52 public class VetorObjeto {
53     public Object[] elementos;
54     public int tamanho;
55
56     //Lista de métodos
57     public VetorObjeto(int capacidade) {
58         this.elementos = new Object[capacidade];
59         this.tamanho = 0;
60     }
61
62     public void adiciona(Object elemento) throws Exception {
63         this.aumentaCapacidade();
```

```

64         this.elementos[this.tamanho] = elemento;
65         this.tamanho++;
66     }
67
68     public int tamanho() {
69         return this.tamanho;
70     }
71
72     @Override
73     public String toString() {
74         StringBuilder s = new StringBuilder();
75         s.append("[");
76
77         for (int i = 0; i < this.tamanho - 1; i++) {
78             s.append(this.elementos[i]);
79             if (i < this.tamanho - 2) {
80                 s.append("\n");
81             }
82         }
83
84         s.append("]");
85
86         return s.toString();
87     }
88
89     public Object busca(int posicao) throws Exception {
90         if (posicao >= 0 && posicao < this.tamanho) {
91             return this.elementos[posicao];
92         } else {
93             throw new Exception("Posição inválida.");
94         }
95     }
96
97     public int busca1 (Object elemento) {
98         for (int i = 0; i < this.tamanho; i++) {
99             if (this.elementos[i].equals(elemento)) {
100                 return i;
101             }
102         }
103
104         return -1;
105     }
106
107     //Sobrecarga de método para retornar os dados de objeto ContaBancaria pelo elemento nome
108     public String busca1(String elemento) {
109         if (elemento instanceof String) {
110             String nome = (String) elemento;
111             for (int i = 0; i < this.tamanho; i++) {
112                 if (this.elementos[i] instanceof ContaBancaria) {
113                     ContaBancaria conta = (ContaBancaria) this.elementos[i];
114                     if (conta.getNome().equals(nome)) {
115                         return conta.toString();
116                     }
117                 }
118             }
119         }
120         return "Elemento não encontrado.";
121     }
122
123
124     public boolean adicionaInicio(int posicao, Object elemento) throws Exception {
125         this.aumentaCapacidade();
126         if (posicao >= 0 && posicao < this.tamanho) {
127             for (int i = this.tamanho - 1; i >= posicao; i--) {
128                 this.elementos[i + 1] = this.elementos[i];
129             }

```

```

130         this.elementos[posicao] = elemento;
131         this.tamanho++;
132     }
133     return true;
134 }
135 }
136
137 private void aumentaCapacidade() {
138     if (this.tamanho == this.elementos.length) {
139         Object[] elementosNovos = new Object[this.elementos.length * 2];
140         for (int i = 0; i < this.elementos.length; i++) {
141             elementosNovos[i] = this.elementos[i];
142         }
143
144         this.elementos = elementosNovos;
145     }
146 }
147
148 public void remove(int posicao) throws Exception {
149     if (posicao >= 0 && posicao < this.tamanho) {
150         for (int i = posicao; i < this.tamanho - 1; i++) {
151             this.elementos[i] = this.elementos[i + 1];
152         }
153
154         this.tamanho--;
155     } else {
156         throw new Exception("Posição inválida.");
157     }
158 }
159 }
160
161 import model.ContaBancaria;
162 import model.VetorObjeto;
163
164 import java.math.BigDecimal;
165
166 public class MainTest {
167
168     public static void main(String[] args) {
169         ContaBancaria conta1 = new ContaBancaria("Fulano", "123.456.789-00");
170         ContaBancaria conta2 = new ContaBancaria("Ciclano", "987.654.321-00");
171         ContaBancaria conta3 = new ContaBancaria("Beltrano", "456.789.123-00");
172         ContaBancaria conta4 = new ContaBancaria("João", "789.123.456-00");
173         ContaBancaria conta5 = new ContaBancaria("Maria", "321.654.987-00");
174         ContaBancaria conta6 = new ContaBancaria("José", "654.987.321-00");
175         conta1.depositar(new BigDecimal("1000"));
176         conta2.depositar(new BigDecimal("2000"));
177         conta3.depositar(new BigDecimal("3000"));
178         conta4.depositar(new BigDecimal("1000000"));
179         conta5.depositar(new BigDecimal("100000000"));
180         conta6.depositar(new BigDecimal("250"));
181
182
183         VetorObjeto vetor = new VetorObjeto(5);
184
185         try {
186             vetor.adiciona(conta1);
187             vetor.adiciona(conta2);
188             vetor.adiciona(conta3);
189             vetor.adiciona(conta4);
190             vetor.adiciona(conta5);
191
192             System.out.println("Tamanho do vetor de Contas Bancárias: " + vetor.tamanho());
193             System.out.println("Conta na posição 5: " + vetor.busca(4));
194             System.out.println("Dados de Maria: " + vetor.busca1("Maria"));
195

```

```
196         vetor.adiciona(conta6);
197         System.out.println("Tamanho do vetor de após passar de 5 Contas: " +
vetor.tamanho());
198         System.out.println(vetor.toString());
199         vetor.remove(1);
200         System.out.println("Dados após exclusão da primeira conta: \n" +
vetor.toString());
201
202
203     } catch (Exception e) {
204         e.printStackTrace();
205     }
206 }
207 }
208
```
