

Programação Orientada a Objeto

Collections - ArrayList



Programação de Computadores II

Professor: Edwar Saliba Júnior

Exercício:

1. No exercício anterior você desenvolveu um *software* com as seguintes características:
 - Sistema de Gestão Hospitalar, contendo as classes Médico e Paciente herdeiras de Pessoa.
 - A classe Pessoa possui os seguintes atributos: CPF, identidade, nome, endereço e telefone. E os seguintes métodos: cadastro, alteração e consulta de dados.
 - A classe Médico possui: CTPS, CRM e Especialidade.
 - A classe Paciente possui: Nome do Convênio e o Número.

Neste exercício foi solicitado que fossem implementadas estas classes com o intuito de cadastrar, alterar, excluir ou consultar um único objeto da classe Paciente e Médico.

Usando ArrayList, altere o programa desenvolvido na última aula de modo que agora possamos cadastrar quantos Médicos e Pacientes quisermos. E também de modo que possamos fazer quaisquer uma das operações básicas de um *software* de cadastro (inclusão, alteração, exclusão, consulta e relatório), tanto com os Médicos cadastrados como também com os Pacientes.

A seguir, encontra-se um exemplo de utilização de ArrayList (Figura 1, 2 e 3):

```
1 package arraylist;
2
3 public class Pessoa {
4     private String nome;
5     private float altura;
6     private int idade;
7
8     public Pessoa(String nome, float altura, int idade) {
9         this.nome = nome;
10        this.altura = altura;
11        this.idade = idade;
12    }
13
14    public void alterar(String nome, float altura, int idade) {
15        this.nome = nome;
16        this.altura = altura;
17        this.idade = idade;
18    }
19
20    public void consultar(){
21        System.out.println("Nome : " + nome);
22        System.out.println("Altura : " + altura);
23        System.out.println("Idade : " + idade);
24    }
25 }
```

Figura 1: Classe Pessoa

```

1  package arraylist;
2
3  import java.util.ArrayList;
4  import java.util.Iterator;
5  import java.util.Scanner;
6
7  public class UsandoArrayList {
8
9  public static void main(String[] args) {
10     // Criando um ArrayList de objetos do tipo Pessoa.
11     ArrayList<Pessoa> pessoas;
12     pessoas = new ArrayList<Pessoa>();
13
14     // Declarando um objeto Iterator para manipular o ArrayList.
15     Iterator i;
16     // Atribuindo o iterator do ArrayList "pessoas" para "i".
17     i = pessoas.iterator();
18
19     Scanner e = new Scanner(System.in);
20
21     // Declaração de variáveis.
22     String nome;
23     int idade, resp, posicao;
24     float altura;
25     Pessoa p;
26
27     do{
28         System.out.println("\n\n-=[Controle de Pessoas]=-");
29         System.out.println("1 - Incluir");
30         System.out.println("2 - Alterar");
31         System.out.println("3 - Excluir");
32         System.out.println("4 - Consultar");
33         System.out.println("5 - Relatório");
34         System.out.println("6 - Sair");
35         System.out.println("Opção: ");
36         resp = e.nextInt();
37         e.skip("\n");
38
39         switch(resp){
40             case 1:
41                 System.out.println("\nDigite os dados da Pessoa:");
42                 System.out.println("Nome   :");
43                 nome = e.nextLine();
44                 System.out.println("Altura :");
45                 altura = e.nextFloat();
46                 System.out.println("Idade  :");
47                 idade = e.nextInt();
48
49                 // Cria um objeto "p" do tipo Pessoa.
50                 p = new Pessoa(nome,altura,idade);
51
52                 // Adiciona "p" ao ArrayList "pessoas".
53                 pessoas.add(p);
54
55                 break;
56             case 2:
57                 System.out.println("Qual posição deseja alterar?");
58                 posicao = e.nextInt();
59                 e.skip("\n");
60
61                 // "p" recebe uma referência para um objeto em "pessoas".
62                 p = (Pessoa) pessoas.get(posicao);

```

Figura 2: Classe Principal - Parte 1/2

```

62         p = (Pessoa) pessoas.get(posicao);
63         System.out.println("\n\n--[Dados atuais da Pessoa]=-");
64         p.consultar();
65
66         System.out.println("\nDigite os novos dados da Pessoa:");
67         System.out.println("Nome   :");
68         nome = e.nextLine();
69         System.out.println("Altura :");
70         altura = e.nextFloat();
71         System.out.println("Idade  :");
72         idade = e.nextInt();
73
74         // Adiciona "p" ao ArrayList "pessoas".
75         p.alterar(nome,altura,idade);
76
77         break;
78     case 3:
79         System.out.println("Qual posição deseja excluir?");
80         posicao = e.nextInt();
81
82         System.out.println("\n\n--[Dados da Pessoa]=-");
83         pessoas.get(posicao).consultar();
84
85         int conf;
86         do{
87             System.out.println("\nConfirma exclusão? (1-Sim / 2-Não):");
88             System.out.println("Resposta: ");
89             conf = e.nextInt();
90
91             if(conf == 1)
92                 // Apaga o objeto contido em uma posição do ArrayList.
93                 pessoas.remove(posicao);
94             if(conf == 2)
95                 System.out.println("A pessoa não foi excluída.");
96         }while((conf != 1) && (conf != 2));
97
98         break;
99     case 4:
100        System.out.println("Qual posição deseja consultar?");
101        posicao = e.nextInt();
102
103        // "p" recebe uma referência para um objeto em "pessoas".
104        p = (Pessoa) pessoas.get(posicao);
105        System.out.println("\n\n--[Dados da Pessoa]=-");
106        p.consultar();
107
108        break;
109     case 5:
110        System.out.println("\n\n--[Relatório de Pessoas]=-");
111        // "i" recebe uma referência para o início do ArrayList.
112        i = pessoas.iterator();
113        // Testa se existe um objeto.
114        while(i.hasNext()){
115            // Atribui uma referência de um objeto Pessoa para "p".
116            p = (Pessoa) i.next();
117            p.consultar();
118            System.out.println("\n");
119        }
120    }
121    }while(resp != 6);
122 }
123 }

```

Figura 3: Classe Principal - Parte 2/2