

Universidade da Beira Interior

Programação Orientada a Objectos

Cursos: 1º ciclo: Eng.ª Informática, Matemática e Aplicações; 2º ciclo em EEC

Exame, 2021/02/05

Sem consulta e sem telemóvel
Duração: 90 minutos, 13.0 valores

1 – Suponha que para a entidade Pessoa se definiu uma classe, esquematizada abaixo, com os atributos: número de cartão de cidadão (cc), nome, uma lista de endereços de mail (contactos) e um array de telefones (telefones) em que no máximo cada pessoa tem **dois** telefones.

```
public class Pessoa {  
    private int cc;  
    private String nome;  
    private ArrayList<String> contactos;  
    private int[] telefones;
```

- a) Para a classe Pessoa, defina um construtor que receba como parâmetros o cartão de cidadão e o nome.
- b) Para a classe Pessoa, defina os getters e os setters. Deve construir o setContactos e o setTelefones de forma a que seja alterado o conjunto de contactos e de telefones e não apenas os seus endereços.
- c) Para a classe Pessoa, defina o método toString.
- d) Para a classe Pessoa, defina o método equals.
- e) Para a classe Pessoa, defina um construtor de cópia.
- f) Para a classe Pessoa, defina o método clone.
- g) Para a classe Pessoa, construa um método que verifica se um dado contacto faz parte da lista de contactos. O método deve devolver o valor true se o contacto dado como parâmetro estiver na lista e false caso contrário.
- h) Para a classe Pessoa, construa um método que verifica se um dado telefone faz parte do array de telefones. O método deve devolver o valor true se o telefone dado como parâmetro estiver no array e false caso contrário.

2 – Construa uma classe de teste, onde define a Pessoa **p1** com os seguintes atributos: cc = 12345; Nome = “Maria”; contactos = (amigo1@gmail.com, amigo2@gmail.com) e telefones = (1111 e 2222)

3 - Supondo que a seguir ao código do exercício 2 coloca as instruções:

```
System.out.println(p1);  
Pessoa p2 = new Pessoa (p1);  
Pessoa p3 = (Pessoa)p1.clone();  
Pessoa p4 = p1;  
p1.setNome("Novo Nome");  
System.out.println(p1);  
System.out.println(p4);  
System.out.println(p1 == p4 );  
System.out.println(p1.equals(p2));
```

a) Qual o output deste bloco de código?

b) Quais as variáveis que existem no programa anterior e qual o seu valor?

4 – Para a classe de teste construa um método público e static que receba como parâmetros um objeto do tipo ArrayList<Pessoa> e um objeto do tipo Pessoa. O método deve adicionar a pessoa à lista caso não exista nela uma pessoa com o mesmo nome. Se já existir na lista uma pessoa com o nome da pessoa que se pretende adicionar, deve ser gerada uma exceção do tipo Exception com a mensagem de erro “Nome repetido”. A exceção será tratada no programa que chamar o método.

5 – Um Aluno é uma Pessoa que tem como atributo adicional o nome da escola que frequenta.

a) Defina o cabeçalho da classe Aluno e os atributos.

b) Para a classe Aluno defina um construtor que receba como parâmetro um objeto do tipo Pessoa e a escola que o aluno frequenta.

c) Para a classe Aluno construa um método que permita adicionar um novo contacto ao aluno.

6 - Supondo que para a classe Aluno foram definidos os métodos get e set para o atributo que representa o nome da escola, construa, para a classe de teste, um método público e static que receba uma arrayList de Pessoas e conte quantos alunos da UBI têm mais de 2 contactos.

7 – Explique o que é e como funciona o mecanismo de ligação (binding) em Orientada a Objetos?

8 – O que é, e para que serve um Construtor em Java?

9 – Explique a diferença entre uma variável de classe e uma variável de instância.

10 – Suponha que tem o código:

```
try {  
    I1; I2; I3;  
}catch (Exception e)  
{ I4 }  
finally {  
    I5  
}  
I6  
...
```

a) Que instruções são executadas se ocorrer uma exceção na instrução I2?

b) Que instruções são executadas se não ocorrer nenhuma exceção?

11 – Suponha a seguinte interface: **public interface Exemplo{
 public int metodo1(int x, String y);
 public int metodo2(int x, String y);
}**

- Explique como faria para a sua classe Pessoa do exercício 1 implementar a interface Exemplo.

12 – O que é o Polimorfismo de inclusão? Dê um exemplo.

Classe java.util.ArrayList:

```
ArrayList() // construtor vazio, dimensão inicial zero.  
boolean add(Object element)  
// adiciona o elemento especificado ao final da lista  
void add( int index, Object obj)  
//insere o elemento especificado na posição index  
Object remove(int index )//remove o elemento da posiçã index  
boolean remove( Object o)  
//remove a primeira ocorrência do objecto dado como parâmetro  
Object set (int position, Object obj )  
// substitui o elemento da posição index pelo elemento dado  
Object get (int position)//devolve o elemento da posição index  
void clear() // remove todos os elementos da lista  
Object clone() // devolve uma cópia da lista  
boolean contains(Object element)  
// devolve true se a lista contém o elemento especificado  
boolean equals ( Object obj)  
// permite comparar duas listas  
int indexOf(Object element)  
// procura o índice da 1ª ocorrência de elemento  
boolean isEmpty() // verifica se a lista não tem componentes  
int size() // devolve a dimensão actual  
String toString ()
```
