

Universidade da Beira Interior

Programação Orientada a Objectos

Cursos: 1º ciclo: Eng.ª Informática, Tecnologias e Sistemas de Informação,

SEM CONSULTA

Duração: 2 horas

Antes de responder a cada exercício deve-o ler na totalidade.

1 – Um espetáculo pode ser caracterizado por um código que o identifica, um nome, uma duração (em minutos) e uma classificação etária, que significa a idade mínima exigida aos espectadores. Estude a classe `Espetaculo` esquematizada abaixo:

```
public class Espetaculo {
    private static int ultimo = 0 ;
    private int codigo;
    private String nome;
    private int duracao;
    private int idade;

    public Espetaculo ( String nome){
        ultimo ++;
        codigo = ultimo;
        this.nome = nome;
    }
    // construtor de cópia – cria uma objeto com todos os atributos iguais ao do objeto parâmetro
    public Espetaculo (Espetaculo e){ ... } .
    public static int getUltimo(){ ... }
    public int getCodigo(){ ... }
    public String getNome(){ ... }
    public int getDuracao(){ ... }
    public int getIdade(){ ... }
    public static void setUltimo(int ult){ ... }
    public void setCodigo(int codigo){ ... }
    public void setNome (String nome){ ... }
    public void setDuracao (int duracao){ ... }
    public void setIdade (int idade){ ... }
    public Object clone(){
        Espetaculo e = new Espetaculo(this);
        return e;
    }
    // Método equals, devolve true se todos os atributos do objeto recetor forem iguais aos do objeto parâmetro.
    public boolean equals (Object o){ ... }
```

a) Construa o constructor de cópia

a) Construa o método `toString`.

b) Construa o método `equals`.

2 – Construa uma classe de exceção com o nome `DuplicadoException`.

3 – Considere o programa da classe `Main` listada abaixo:

```
public class Main {
    public static void main(String[] args) {
        Espetaculo e1 = new Espetaculo("Mandela: Longo Caminho para a Liberdade");
        e1.setDuracao(146);
        e1.setIdade(12);
        System.out.println(e1);
        Espetaculo e2 = new Espetaculo(e1);
        System.out.println(e1 == e2);
        System.out.println(e1.equals(e2));
    }
}
```

a) Represente esquematicamente as variáveis que existem no programa anterior, indicando para cada uma qual o seu valor final.

b) Qual o output do programa anterior?

c) Para a classe `Main` acima, construa um método público e `static` chamado **contaMaior18**, que receba como parâmetro um objeto do tipo `java.util.ArrayList<Espetaculo>` e devolva como resultado o número de espetáculos para maiores de 18 anos que a lista contém.

d) Para a classe `Main` acima, construa um método público e `static` chamado **duracaoMedia** que receba como parâmetros um objeto do tipo `java.util.ArrayList<Espetaculo>` e um inteiro que represente uma idade. O método deve devolver o valor médio da duração dos espetáculos que a lista contém, para maiores que a idade recebida como parâmetro.

e) Para a classe `Main` acima, construa um método público e `static` chamado **novoEspetaculo** que receba como parâmetros um objeto do tipo `java.util.ArrayList<Espetaculo>` e um objeto do tipo `String` com o nome de um espetáculo. O método deve verificar se esse espetáculo faz parte da lista. Em caso afirmativo deve ser gerada a exceção `DuplicadoException` com a mensagem de erro: “Erro: o espetáculo `EEE` já está inserido”, onde `EEE` deve conter o nome do espetáculo recebido como parâmetro. Se o espetáculo não faz parte da lista, deve pedir ao utilizador os restantes dados do espetáculo (duração e idade) e inserir o objeto do tipo `Espetaculo` no final da lista. A exceção gerada neste método deve ser tratada no método que o invocar, isto é no `main`.

Nota: Suponha que tem acesso à classe `Ler` usada nas aulas práticas.

f) Construa um bloco de código para o método `main` anterior que permita testar o método **novoEspetaculo**.

4 – O Cinema é um espetáculo com um realizador, uma sinopse e um conjunto de atores.

a) Defina a classe Cinema, como subclasse de Espetaculo, com os atributos: String realizador; String sinopse; ArrayList <String> atores; e com um construtor que receba como parâmetros um objeto do tipo Espetaculo, o realizador e a sinopse.

b) Para a classe cinema defina os seguintes métodos:

i) Método de consulta (get) do atributo atores

ii) Método de modificação (set) do atributo atores.

iii) Método que permita adicionar um novo ator ao filme caso ainda não exista na lista de atores.

iv) Método que permita modificar a duração do filme//espetáculo.

5 - Numa frase, com tamanho máximo de duas linhas, defina cada um dos seguintes conceitos:

i) Classe

ii) Instância

iii) Assinatura de um método

iv) Sobreposição (overriding) de um método

v) Interface

vi) Classe abstrata

Class 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 ()
```