Universidade da Beira Interior

Programação Orientada a Objectos

Cursos: Matemática / Informática, Ensino da Informática, Engenharia Informática

Frequência - 2006/01/9

Duração: 2 horas

SEM CONSULTA

(8 valores)
I

1 - Suponha que uma pessoa é uma entidade de quem sabemos o nome, o sexo e a nacionalidade.
- Construa a classe Pessoa, definindo os atributos da classe com os adequados modificadores de acesso e definindo os seguintes métodos:

a) Construtor sem parâmetros.

b) Construtor de cópia.

c) Método toString.

d) Método equals.

e) Defina ainda as assinaturas dos métodos selectores e modificadores (getters e setters) para cada atributo da classe.

2 – Suponha a classe Disciplina como definida abaixo:

```
public class Disciplina{
        private int codigo;
        private String nome;
        public Disciplina (int cod, String n){
                código = cod;
                nome = n;
        }
        public int getCodigo() {...}
        public String getNome() {...}
        public void setCodigo( int cod){...}
        public void setNome (String nome) {...}
        public String toString() {...}
        public Object clone(){...}
        public boolean equals (Object o){...}
}
```

a) Suponha agora que um Aluno é uma Pessoa que frequenta um curso, identificado por um nome, tem um número de aluno, e está inscrito a um conjunto de disciplinas (objecto do tipo Vector). Apenas as disciplinas a que o aluno está inscrito fazem parte da instância do objecto Aluno. O número de aluno é atribuído sequencialmente, de forma automática, a cada novo objecto do tipo Aluno que é criado.

Defina para a classe Aluno:

i) – O cabeçalho e os atributos.

ii) – Um construtor de cópia

iii) – Um construtor que recebe como parâmetro um objecto do tipo Pessoa e o nome do curso.

iv) – O método toString.

v) – Um método que dado o nome de uma disciplina, verifique se o aluno está inscrito a essa disciplina.

vi) – Um método para atribuir ao aluno uma nova disciplina.

vii) - Um método para ordenar alfabeticamente o vector de disciplinas pelo seu nome.

Para a alínea vii) Pode usar o seguinte método auxiliar:

```
private  int procuraMenorAlfab (Vector v, int inicio){
 // Dado um Vector, v, devolve a posição do objecto que contém o nome alfabeticamente
//menor existente na secção limitada por inicio e v.size()-1.
   int iMenor = inicio;
    Disciplina c1, c2;
    for(int i=inicio+1; i < v.size(); ++i){
        c1 = (Disciplina) v.elementAt(i);
        c2 = (Disciplina) v.elementAt(iMenor);
        if (c1.getNome().compareTo(c2.getNome())<0)
            iMenor = i;
    }
    return iMenor;
}
```

API da classe java.util.Vector:

```
Vector()// construtor vector vazio, dimensão inicial zero.
Vector(int capacidInicial)   // construtor vector vazio, com dimensão inicial.
void addElement(Object elemento) // adiciona o elemento especificado ao final do vector.
void insertElementAt(Object obj, int indice)// insere o elemento especificado na posição indice.
void removeElementAt(int indice) // remove o elemento na posição indice.
void setElementAt(Object obj, int indice) // substitui o elemento da posição indice pelo objecto dado.
Object elementAt(int indice)// devolve o componente presente no indice.
void clear() // remove todos os elementos do vector.
Object clone()  // devolve uma cópia do vector.
boolean contains(Object elemento)   // verifica se o objecto especificado é um componente deste vector
Object firstElement() // devolve o primeiro componente (indice 0) do vector.
Object lastElement() // devolve o último componente do vector.
int indexOf(Object elemento) // procura o índice da 1ª ocorrência de elemento (utiliza o método equals)
int indexOf(Object elemento, int indice) // inicia a procura anterior na posição indice.
boolean isEmpty() // verifica se o vector não tem componentes
int size()            // devolve a dimensão actual.
...
```