

## Exercício-Programa 2 – 2012

O exercício-programa solicitado apresenta 4 classes:

- `public class InfoAluno`
- `public class BancoDeDadosInfoAluno`
- `public class Resources`
- `public class EP2`

### 1. `public class InfoAluno`

Esta classe é a célula básica do exercício-programa. Ela contém as informações referentes a um aluno.

a. Atributos: Nome e Número USP do aluno, respectivamente, `nomedoaluno` e `nusp`, do tipo `String`. Notas da P1, P2, EP1, EP2, EP3, EP4, média de provas, média de EPs e média final, respectivamente, `P1`, `P2`, `EP1`, `EP2`, `EP3`, `EP4`, `T`, `L` e `MF`, do tipo `double`. Todas são do tipo de acesso `private`.

b. Construtor:  
`public InfoAluno():` Atribue valor `null` para os atributos do tipo `String` do objeto e `0` para os do tipo `double`.

c. Métodos: A classe apresenta dois tipos de métodos: os de inserção de dados e os de coleta. Para simplificar a explicação, será adotado `InsertX` para o método de inserção do atributo `X`, `GetX` para o método responsável pela coleta deste mesmo atributo `X`, e `TIPO` para o tipo do atributo, `String` ou `double`, convenientemente.

i. `public void InsertX(TIPO x):` Atribue o valor `x` ao atributo `X` do objeto;

ii. `public TIPO GetX():` Retorna o valor presente no atributo `X` do objeto;

Deve-se dar enfoque especial aos métodos `GetPMean()`, `GetEPMean()` e `GetFinalMean()`. Estes retornam as médias do aluno, respectivas às provas, EPs e média final, segundo as fórmulas:

$$T = \frac{P1+2.P2}{3} \quad L = \frac{EP1+EP2+EP3+EP4}{4} \quad MF = \frac{5.T+L}{6}$$

d. Override de método:

`public int compareTo(InfoAluno p):` Ao implementar a interface `Comparable`, será necessário sobrescrever o método `compareTo` que retorna um `int`. O método `compareTo` possibilita que o `List` de alunos, utilizado para ordenar o Banco de Dados, possa ser ordenado pelo nome.

### 2. `public class BancoDeDadosInfoAluno`

Esta classe contém os atributos e métodos que caracterizam um Banco de Dados.

a. Atributos:

i. `private final int NMax:` Número máximo de alunos que o Banco de Dados comporta. Valor fixo;

ii. `private InfoAluno[] BancoDeDados:` Array contendo as informações dos alunos;

iii. `NumeroDeAlunos:` Número de objetos do tipo `InfoAluno` que o array `BancoDeDados[]` contém em um determinado momento do programa.

b. Construtor:

`public BancoDeDadosInfoAluno():` reserva memória para o array `BancoDeDados` do tipo `InfoAluno[]` e atribue `0` para `NumeroDeAlunos`.

c. Métodos: A classe apresenta métodos de manipulação de um Banco de Dados.

i. `public boolean InsereAluno(...):` Os parâmetros de entrada deste método são os atributos de um objeto do tipo `InfoAluno()` (Nome, NUSP, notas). Estes foram omitidos para manter esta documentação pouco poluída. O aluno a ser inserido é colocado no final do Banco de Dados atual e este novo Banco de Dados é ordenado alfabeticamente por um método que será visto a seguir;

ii. `public boolean DeletaAluno(String DelName):` Retorna `true` caso encontrar um aluno com nome `DelName` no array

`BancoDeDados` e delete-o, ou `false` caso contrário. O algoritmo de busca empregado foi a Busca Binária;

iii. `public void SearchAluno(String SearchName):` Realiza uma busca no array `BancoDeDados` por um aluno com nome `SearchName` e, caso encontre-o, imprime na tela os dados deste. Caso contrário, emite o aviso de que o aluno buscado não existe no banco de dados em questão. O algoritmo de busca empregado foi a busca Binária;

iv. `public boolean FileImport(String FileName):` Retorna `true` caso importar TODOS os alunos e seus respectivos dados de um arquivo `.txt` externo para o array `BancoDeDados[]`, ou `false` caso contrário. Cada linha de leitura do arquivo deve apresentar a seguinte estrutura: NOME DO ALUNO NUSP P1 P2 EP1 EP2 EP3 EP4. As possibilidades para o retorno `false` são: erro na abertura do arquivo; erro na leitura do arquivo; ou o atributo `NumeroDeAlunos` atingiu o valor máximo, igual à `NMax`. Caso haja “vagas” para alunos no banco de dados depois da abertura do arquivo, mas não para inserir TODOS que nele existem, os primeiros serão anexados ao Banco de Dados até que se atinja o valor máximo, igual à `NMax`. Depois disso, o programa acusará que o limite foi atingido e retornará `false`. Ao término da importação, o novo Banco de Dados é ordenado alfabeticamente por um método que será visto a seguir;

v. `public boolean FileExport(String FileName):` Retorna `true` caso exportar o banco de dados para um arquivo `.txt` externo, ou `false` caso contrário. A estrutura de cada linha do arquivo exportado é a mesma esperada pelo método `FileImport()`, de forma que este método funcione como um backup. As possibilidades para o retorno `false` são: erro na abertura do arquivo; ou erro na escrita do arquivo;

vi. `public double[] ClassMean():` Retorna um array contendo as médias das 9 notas que um aluno possui, atributos do objeto `InfoAluno()`. A média de cada nota foi calculada da seguinte forma:

$$\bar{X} = \frac{\sum_{i=0}^{NumeroDeAlunos-1} X_i}{NumeroDeAlunos}$$

vii. `public double[] ClassStdDev(double[] meanc):` Recebe as médias calculadas no método anterior e retorna um array contendo os desvios-padrão das 9 notas que um aluno possui, atributos do objeto `InfoAluno()`. O desvio-padrão de cada nota foi calculado da seguinte forma:

$$\sigma_x = \sqrt{\frac{1}{NumeroDeAlunos} \sum_{i=0}^{NumeroDeAlunos-1} (X_i - \bar{X})^2}$$

viii. `public void PrintAluno(int index):` Imprime na tela todos os atributos (Nome, NUSP, notas) do aluno presente no índice `index`;

### 3. `public class Resources`

Esta classe é uma tentativa de implementar uma interface que contém os recursos de ordenação alfabética de um Banco de Dados e Busca Binária de um aluno. Entretanto, como não visto em aula, limitou-se apenas a ser uma superclasse da classe `BancoDeDadosInfoAluno`, tendo o mesmo resultado final.

- Atributos: Não possui
- Construtor: Não possui
- Métodos:

i. `public InfoAluno[] alphabetic(InfoAluno[] Data, int k):` Ordena o array `Data` do tipo `InfoAluno[]` em ordem alfabética fazendo uso de `List` e do comando `Collections.sort`. O outro parâmetro `k` corresponde ao último componente válido do array `Data` (no nosso caso, `NumeroDeAlunos-1`);

ii. `public int BinSearch(InfoAluno[] Data, int left, int right, String SearchName):` Implementa o algoritmo de busca binária para buscar o aluno com nome `SearchName` no array `Data` do tipo `InfoAluno[]`. Este método é auto-recursivo, o que explica os parâmetros `right` e `left`, correspondendo aos extremos esquerdo e direito do array buscado.

### 4. `public class EP2`

Esta classe contém o `main` do programa. Ele é composto basicamente pela estruturação visual do programa e as opções de interação do usuário com o programa. Estas são:

- 0 – Inserir aluno manualmente
- 1 – Remover aluno
- 2 – Buscar Aluno
- 3 – Leitura da base de dados de um arquivo externo
- 4 – Escrita da base de dados em um arquivo externo
- 5 – Impressão do relatório final
- 6 – Finalizar execução