

ESQUEMA AULA PRÁTICA 10

□ Ficheiros

1 – Implemente a classe Aluno listada abaixo.

```
import java.io.*;
public class Aluno implements Serializable {
// Para poder ser escrita num ficheiro a classe tem que implementar a interface
//Serializable

private static int ultimo = 0;
private int numero;
private String nome;

public Aluno (String n){
ultimo++;
numero = ultimo;
nome = n;
}
public static int getUltimo() { return ultimo; }
public static void setUltimo(int ultimo) { Aluno.ultimo = ultimo }
public void setNome (String n ){ nome = n; }
public String getNome (){ return nome; }
public int getNumero (){ return numero; }

public String toString (){
return "Aluno nº: " + numero + "\t Nome: " + nome ;
}
}
```

2 – Construa uma pequena aplicação que permita as seguintes opções:

- 1 - Criar aluno
- 2 – Consultar aluno, dado o seu número
- 3 – Consultar aluno, dado o seu nome
- 4 – Listar todos os alunos
- 5 – Apagar um aluno
- 6 – Corrigir o nome de um aluno
- 7 – Terminar

- Os alunos criados deverão ser armazenados num objeto alunos, do tipo `java.util.ArrayList<Aluno>`. Sempre que a lista de alunos for modificada, deverá ser escrita para um ficheiro (`fichAlunos`). Quando o programa é executado deverá começar por ler o conteúdo desse ficheiro.

3 – Analise a aplicação e tente perceber o que se passa com a variável estática quando escreve a lista de alunos para o ficheiro.

4 - No programa que escreve para o ficheiro, substitua a instrução,
`os.writeObject(alunos);`

por:

```
os.writeInt(Aluno.getUltimo());  
os.writeObject( alunos);
```

No programa que lê o ficheiro, substitua a instrução
`ArrayList lista = (ArrayList) is.readObject();`

por:

```
int ult= is.readInt();  
Aluno.setUltimo(ult);  
ArrayList lista = (ArrayList) is.readObject();
```

- O que mudou?

Exercícios de revisão da matéria:

5 - O programa abaixo usa a classe `JOptionPane` para pedir e mostrar dados ao utilizador.

a) Implemente o programa e estude a classe `JOptionPane`.

```
import javax.swing.*;  
public class TesteOption{  
    public static void main (String[]args){  
  
        JOptionPane.showMessageDialog(null,"Ora viva");  
        String valor = JOptionPane.showInputDialog(null, "Introduza um inteiro:");  
        int i = Integer.parseInt(valor);  
        JOptionPane.showMessageDialog(null, " O valor dado foi "+i);  
        System.exit(0);  
    }  
}
```

b) O que acontece se, quando a caixa de diálogo lhe pede um valor, você pressionar o botão “Cancel”? Altere o programa de forma a não terminar anormalmente.

c) Modifique o programa para ler um valor do tipo `double`.

6 - Simplificando, podemos afirmar que, uma pessoa é alguém de quem sabemos o nome, o sexo e a nacionalidade. Programe a classe `Pessoa` com os construtores (entre eles implemente o construtor não parametrizado – qual a utilidade? – e o *construtor por cópia*), selectores e modificadores que considerar necessários bem como os métodos públicos `toString`, `clone` e `equals` (ainda que os considere pouco naturais para uma pessoa terão interesse do ponto de vista da programação!).

7 - Programe a classe `Autor` de obra literária. Um autor é uma pessoa sobre a qual será interessante saber o ano de nascimento e falecimento (caso já tenha falecido), a sua nacionalidade bem como o prémio literário mais importante que conquistou, se detivermos esta informação.

8 - Programe a classe `Musico`. Um músico pode ser um indivíduo ou uma banda de que interessa manter uma descrição e o seu estilo musical dominante.

9 - Programe a classe `Amigo`. Um amigo é uma pessoa de quem sabemos a data de nascimento, o ano em que o conhecemos, um contacto, o nível de amizade que por ela nutrimos e ainda o “parceiro” com quem normalmente “anda”.

10 - Pretende-se organizar a colecção de livros e CDs áudio das nossas biblioteca e discoteca particulares¹.

Um livro é caracterizado pelo título, pelo autor, pela língua em que está escrito, pelo seu valor afectivo e se já o lemos pela indicação da data em que concluímos a sua leitura. Por vezes emprestamos livros a amigos e é necessário saber se o livro está emprestado e a quem.

A nossa colecção de CDs de música tem um tratamento semelhante à dos livros: podemos emprestá-los, têm determinado valor afectivo e poderemos querer saber quando os ouvimos pela última vez.

Existem pois analogias entre as duas classes e também diferenças (por exemplo, o CD de música não tem autor mas sim músico/banda e não será necessário indicar a língua mas sim o estilo de música).

Agrupe as semelhanças de comportamento e estrutura numa classe chamada `Biblionimo`². O objectivo desta classe é simplificar a subsequente programação das classes `Livro` e `CDAudio`.

11 - Programe uma classe (`FilmeDVD`) para manter e manipular informação sobre os filmes de uma filмотeca em suporte DVD.

¹ Exercício adaptado de "Programação com classes em C++", P. Guerreiro, FCA, 2000.

² Bibliónimo significa nome de qualquer livro de reputação universal (do grego *bíblion* (livro) + *ónyma* (nome)). Parece ser por isso um nome apropriado (até que sugiram um melhor) para uma classe que procura disponibilizar o comportamento comum dos nossos objectos culturais de consumo.