

AULA TEÓRICA 4

Tema 5. Colecções.

- Listas (noção geral)
- Array de objectos
 - Criação.
 - Ordenação de um array de objectos

Array de objectos

Nos exemplos anteriores array armazenava dados de tipos primitivos. **Array de objectos** é um array cujos elementos são objectos ou, mais correctamente, referências para objectos. Característica importante de array de objectos: criação de array e criação de objectos, que serão armazenados no array, são duas coisas separadas.

Criação

Exemplo:

Elabore um programa em Java composto, no mínimo, por 3 classes que permita gerir a informação sobre venda de revistas. A informação sobre: nome de revista, quantidade e preço unitário encontra-se num ficheiro do tipo texto. Ler e visualizar para ecrã informação gravada.

Pretende-se:

- ler informação do ficheiro e armazenar num array de objectos;
- calcular total global das vendas realizadas;
- visualizar os dados armazenados no array de objectos com totais calculados para cada revista. Aplicar a formatação em **Mt** onde é necessário;
- gravar informação sobre os nomes das revistas e respectivos totais num outro ficheiro do tipo texto;

```

import java.text.DecimalFormat; import java.io.*;

public class Revista {
    private String nome;
    private int quant;
    private float preco;
    private float total;
    private DecimalFormat moeda;

    public Revista(String no, int qu, float pr) {
        nome = no;
        quant = qu;
        preco = pr;
        total = calcTotal();
        moeda = new DecimalFormat ("###,###.00 Mt");
    }

    private float calcTotal() {
        return quant*preco; }
    public float getTotal() {
        return total; }
    public String getNome() {
        return nome; }
    public String toString() {
        return nome+"\t"+quant+" de preco "+ moeda.format(preco)+
            "\ttotal="+ moeda.format(total);
    }
}

```

```

import java.util.StringTokenizer;  import java.io.*;
public class ArrayRevistas {
    private Revista[] lista;private int cont;

    public ArrayRevistas() {
        lista = new Revista[100];cont=0;
    }

    public void lerFichCriarArray() {
        StringTokenizer umaCadeia;
        String umaLinha="", nom, nomeFich = "revistas.txt";int qde; float pr;
        try { FileReader fr = new FileReader(nomeFich);
            BufferedReader fichIn = new BufferedReader(fr);
            umaLinha = fichIn.readLine();
            while (umaLinha != null) {
                umaCadeia = new StringTokenizer(umaLinha,";");
                nom = umaCadeia.nextToken();
                qde = Integer.parseInt(umaCadeia.nextToken());
                pr = Float.parseFloat(umaCadeia.nextToken());
                lista[cont] = new Revista(nom,qde,pr);
                cont++;
                umaLinha = fichIn.readLine();
            }
            fichIn.close();
        }catch (FileNotFoundException fn){
            System.out.println(" Ficheiro nao encontrado!");
        }catch (NumberFormatException nn){
            System.out.println(nn.getMessage());
        }catch (IOException ex){ System.out.println(ex.getMessage()); }
    }
}

```

```

public float calcTotGlobal() {
    float som = 0;
    for (int j = 0; j < cont; j++)
        som += lista[j].getTotal();
    return som;
}

public String toString() {
    String x = "";
    for (int a = 0; a < cont; a++)
        x += lista[a] + "\n";    //.toString() opcional
    return x;
}

public void gravarFichTxt(String nf) {
    try { FileWriter fw = new FileWriter(nf);
        BufferedWriter fichOut = new BufferedWriter(fw);
        for (int k = 0; k < cont; k++) {
            fichOut.write(lista[k].getNome()+";"+lista[k].getTotal());
            fichOut.newLine();
        }
        fichOut.close();
    } catch (IOException xx) { System.out.println(xx.getMessage());
    }
}

```

```

import java.text.DecimalFormat;
import java.io.*;
public class GerirRevistas {
    public static void main (String args[]) {
        ArrayRevistas arev= new ArrayRevistas();
        DecimalFormat m = new DecimalFormat ("###,###.00 Mt");

        arev.lerFichCriarArray();
        System.out.println("Dados do fich txt e armazenados no array: ");
        System.out.println(arev);
        System.out.println("Total =" + m.format(arev.calcTotGlobal()));
        arev.gravarFichTxt("Out.txt");
    }
}

```

Conteúdo do ficheiro do tipo texto "revistas.txt":

```

Golo;20;200
Tempo;30;150
. . . .

```

Ordenação de um array de objectos

Exemplo: desenvolver um programa que leia os dados de um conjunto de estudantes (nome, e um conjunto de notas), calcule a sua média e ordene os estudantes por ordem decrescente das médias.

```
public class Estudante {  
    //Atributos  
    private Validacoes vv;  
    private String nome;  
    private byte notas[];  
    private byte media;  
  
    //Construtor da classe, promove a inicialização dos atributos  
    public Estudante() {  
        vv = new Validacoes();  
        nome = vv.validarString((byte)5, (byte)25, " nome: ");  
        byte numNotas = vv.validarByte((byte)0, (byte)5, "Quantas notas? ");  
        notas = introdNotas(numNotas);  
        media = calcMedia();  
    }  
}
```

```

//Cria array Notas e preenche com notas do aluno
private byte[] introdNotas(byte numN) {
    notas = new byte[numN];
    for (int i = 0; i < numN; i++) {
        notas[i] = vv.validarByte((byte)0, (byte)20, (i+1)+"-a nota: ");
    }
    return notas;
}

//devolve uma String composta pelas notas
private String devolveNotas() {
    String visual=""; //de um estudante
    for (byte k = 0; k < notas.length; k++)
        visual += notas[k]+" ";
    return visual;
}

//Método para cálculo da média de um estudante
private byte calcMedia() {
    byte soma = 0;
    for (int z=0; z<notas.length; z++)
        soma += notas[z];
    return (byte) Math.round(soma / notas.length);
}

```



```

public byte getMedia() {
    return media; }

public String toString() {
    return nome+", notas: "+devolveNotas()+", Meida="+media+"\n"; }
}

```

=====

```

import java.io.*;
public class Validacoes {
    . . . . .
}

```

=====

```

import java.io.*;
public class Turma {
    Validacoes val;
    private Estudante[] lista;
    byte numEst;

    public Turma() {
        val = new Validacoes();
        numEst = val.validarByte((byte)3, (byte)60, " qde de estud.:");
        lista = new Estudante[numEst];
    }
}

```

```

public void criarArrayEst() {
    for (byte i = 0; i < numEst; i++) {
        System.out.println("Dados do " + (i+1) + "-o estudante:");
        lista[i] = new Estudante(); //criação de objecto e armazen.
    }                                //no array de objectos
}

public String toString() {
    String v="";
    for (byte k=0; k < lista.length; k++)
        v += lista[k] + "\n";      //ou lista[k].toString()
    return v;
}

public void ordenaTurma() {
    Estudante aux;
    byte i_maior;
    for (byte i=0; i < lista.length-1; i++) {
        i_maior = localizaMaior(i);
        aux = lista[i];
        lista[i] = lista[i_maior];
        lista[i_maior] = aux;
    }
}

```

```

private byte localizaMaior(byte inicio) {
    byte ind_maior = inicio;
    for (byte k = (byte)(inicio+1); k < lista.length ; k++)
        if (lista[k].getMedia() > lista[ind_maior].getMedia())
            ind_maior = k;
    return ind_maior;
}

```

```

}

```

```

=====

```

```

import java.io.*;
public class GereTurma {
    public static void main(String[] args) {
        Turma t = new Turma();    //cria um objecto da classe Turma
        t.criarArrayEst();
        System.out.println("Lista de estudantes: ");
        System.out.println(t);    //equivale a (t.toString());
        t.ordenaTurma();
        System.out.println("\nLista ordenada por media: ");
        System.out.println(t);
    }
}

```

Referência bibliográfica:

António José Mendes; Maria José Marcelino.

“Fundamentos de programação em Java 2”. FCA. 2002.

Elliot Koffman; Ursula Wolz.

“Problem Solving with Java”. 1999.

F. Mário Martins;

“Programação Orientada aos objectos em Java 2”, FCA, 2000,

John Lewis, William Loftus;

“Java Software Solutions: foundation of program design”, 2nd edition, Addison-Wesley

John R. Hubbard.

“Theory and problems of programming with Java”. Schaum’s Outline series. McGraw-Hill.

H. Deitel; P. Deitel.

“Java, como programar”. 4 edição. 2003. Bookman.

Rui Rossi dos Santos.

“Programando em Java 2– Teoria e aplicações”. Axcel Books. 2004