

EGLOBAL 2019/2020 (ÉPOCA NORMAL)

As resoluções dos exercícios propostos deverão ter sempre um exemplo de aplicação para demonstrar a compreensão do raciocínio / resolução apresentada.

RESOLUÇÃO:

Grupo I

Nota: Para exemplificar nas propostas de solução das várias alíneas do grupo I, vamos supor que temos as listas $a = [9, 6, 3]$ e $b = [7, 6, 5]$, a nova lista será composta pelos elementos $[9, 0, 5]$.

1. Para este grupo pretende-se construir uma lista a partir de duas listas dadas em que, cada valor da nova lista será o maior entre os valores com o mesmo índice.

a) O'Caml:

```
(* exemplo de listas *)
let a = [9;6;3];;
let b = [7;6;5];;

(* Retorna o maior valor. Caso x e y sejam iguais retorna 0 *)
let maior x y =
  if x > y then x
  else if y > x then y
  else 0
;;

(* List.map2 aplica a função maior a cada índice dos elementos das listas l1 e l2.
Devolve a lista com o maior dos elementos correspondentes, ou 0 caso sejam iguais *)
let getListaMaior l1 l2 =
  List.map2 maior l1 l2;;

getListaMaior a b;;
```

b) Prolog:

```
/* Para o caso de as listas estares vazias, então o valor é vazio */
maior([], [], []).

/* Para o caso em que os elementos no índice i são iguais, então o valor é 0 */
maior([H1|T1], [H2|T2], [H3|T3]) :-
  H1 == H2,
  H3 is 0,
  maior(T1, T2, T3).

/* De seguida para os casos em que os elementos diferem é escolhido o maior dos dois valores
*/
maior([H1|T1], [H2|T2], [H3|T3]) :-
  H1 > H2,
  H3 is H1,
  maior(T1, T2, T3).

maior([H1|T1], [H2|T2], [H3|T3]) :-
  H1 < H2,
  H3 is H2,
  maior(T1, T2, T3).
```

```

/* Apenas para correr o programa com um exemplo */
run :-
A = [9,6,3],
B = [7,6,5],
maior(A, B, L),
write("Nova lista:"), nl,
write(L).

```

c) Java:

```

package novaLista;
import java.util.ArrayList;

public class novaLista {
    // Retorna o maior de dois valores ou 0 se forem iguais
    private int maior(int a, int b) {
        if (a > b) {
            return a;
        }
        else if (b > a) {
            return b;
        }
        else {
            return 0;
        }
    }
    // Devolve a lista com o maior dos elementos correspondentes,
    // ou 0 caso sejam iguais */
    private ArrayList<Integer> getListaMaior(ArrayList<Integer> l1, ArrayList<Integer> l2) {
        ArrayList<Integer> listaMaior = new ArrayList();
        for (int i = 0; i < l1.size(); i++) {
            listaMaior.add(maior(l1.get(i), l2.get(i)));
        }
        return listaMaior;
    }

    // Imprime uma lista de inteiros (para debug)
    private void printLista(ArrayList<Integer> l) {
        for (int i = 0; i < l.size(); i++) {
            System.out.print(l.get(i) + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        // criar objeto para poder chamar metodos
        novaLista q1 = new novaLista(); // usa construtor vazio (default)

        // criar duas listas
        ArrayList<Integer> l1 = new ArrayList();
        ArrayList<Integer> l2 = new ArrayList();

        // adicionar elementos
        l1.add(9); l1.add(6); l1.add(3);
        l2.add(7); l2.add(6); l2.add(5);
        q1.printLista(q1.getListaMaior(l1, l2)); // imprimir a lista
    }
}

```

Grupo II

1. Pretende-se implementar um programa em O'Cam1 que, dado uma árvore binária com inteiros, calcule a soma dos seus elementos.

(Definição do tipo b_tree (árvore binária) que permite armazenar em cada folha tipos genéricos (polimórfica) *)*

```
type 'a b_tree =  
  | Empty  
  | Node of 'a * 'a b_tree * 'a b_tree  
;;
```

(Objeto do tipo b_tree, conforme a árvore do diagrama mostrado *)*

```
let tree =  
  Node(5,  
    Node(7, Node(10, Empty, Empty), Node(15, Empty, Empty)),  
    Node(9, Node(2, Empty, Empty), Node(1, Empty, Empty))  
  )  
;;
```

(Recebe uma árvore binária de inteiros e retorna a soma dos nós *)*

```
let rec soma t =  
  match t with  
  | Empty -> 0  
  | Node(x, l, r) -> x + soma l + soma r  
;;
```

2. Construa um predicado que coloque numa lista L todos os corpos que orbitam em torno de X. E.g., orbita(lua, terra) -> a lua orbita em torno da terra.

Dando alguns factos como exemplo, o programa poderia ser implementado da seguinte forma:

%Base de dados de factos

```
orbita(terra, sol).  
orbita(lua, terra).  
orbita(marte, sol).  
orbita(mercurio, sol).  
orbita(fobos, marte).  
orbita(deimos, marte).
```

/ Predicado que retorna a lista pedida */*

```
corpos_orbitam(Y, X, L) :-  
  findall(Y, orbita(Y, X), L).
```

3.

```
package talao;  
import java.util.ArrayList;  
class Item {  
  String descricao;  
  double precoUnitario;  
  double quantidade;  
  double taxaIVA;  
  
  // Construtor do Item  
  public Item(String descricao, double precoUnitario, double quantidade,  
  double taxaIVA) {
```

```

        this.descricao = descricao;
        this.quantidade = quantidade;
        this.precoUnitario = precoUnitario;
        this.taxaIVA = taxaIVA;
    }

    // Devolve o total do iva do item
    public double getTotalIVA() {
        return quantidade * precoUnitario * taxaIVA / 100.0;
    }

    // Devolve o valor do item com iva
    public double getPrecoComIVA() {
        return quantidade * precoUnitario + getTotalIVA();
    }

    /* Getters e setters (porque atributos sao privados - apesar de neste problema apenas o
    getPrecoUnitario ser utilizados */
    public double getPrecoUnitario() {
        return precoUnitario;
    }

    public void setPrecoUnitario(double precoUnitario) {
        this.precoUnitario = precoUnitario;
    }

    // Mais getters e setters para os restantes atributos privados
    // Devolve uma string com a informação do item
    public String getItemString() {
        return descricao + "\t" + precoUnitario + "\t" + quantidade + "\t" +
            taxaIVA + "\t" + getTotalIVA() + "\t" + getPrecoComIVA();
    }
}

```

```

class Pedido {
    private String nomeCliente;
    private int numIdFiscal;
    private String morada;
    private int numTelemovel;
    private String data;
    private String hora;
    private String nomeEstafeta;
    private double totalIVA;
    private double totalPagar;

    private ArrayList<Item> refeicoes;
    private ArrayList<Item> gratificacoes;
    private ArrayList<Item> descontos;

    // Construtor com todos os dados essenciais
    public Pedido(String nomeCliente, int numIdFiscal, String morada,
        int numTelemovel, String data, String hora, String nomeEstafeta,
        ArrayList<Item> refeicoes, ArrayList<Item> gratificacoes,
        ArrayList<Item> descontos) {
        this.nomeCliente = nomeCliente;
        this.numIdFiscal = numIdFiscal;
        this.morada = morada;
        this.numTelemovel = numTelemovel;
        this.data = data;
        this.hora = hora;
        this.nomeEstafeta = nomeEstafeta;
    }
}

```

```

        this.refeicoes = refeicoes;
        this.gratificacoes = gratificacoes;
        this.descontos = descontos;
        totalIVA = 0.0;
        totalPagar = 0.0;

        calcularIVAtotalEttotalPagar();
    }
    // Pode ser private, e apenas uma funcao auxiliar dentro da classe
    private void calcularIVAtotalEttotalPagar() {
        // refeicoes
        for (int i = 0; i < refeicoes.size(); i++) {
            totalIVA += refeicoes.get(i).getTotalIVA();
            totalPagar += refeicoes.get(i).getPrecoComIVA();
        }
        // gratificacoes
        for (int i = 0; i < gratificacoes.size(); i++) {
            totalIVA += gratificacoes.get(i).getTotalIVA();
            totalPagar += gratificacoes.get(i).getPrecoComIVA();
        }
        // descontos
        for (int i = 0; i < descontos.size(); i++) {
            totalIVA -= descontos.get(i).getTotalIVA();
            totalPagar -= descontos.get(i).getPrecoComIVA();
        }
    }
    // NOTA: getters e setters, como os atributos sao privados, e caso seja necessário
    // aceder-lhes (o que nao e o caso para este problema)
    public String getNomeCliente() {
        return nomeCliente;
    }
    public void setNomeCliente(String nomeCliente) {
        this.nomeCliente = nomeCliente;
    }
    public int getNumIdFiscal() {
        return numIdFiscal;
    }
    public void setNumIdFiscal(int numIdFiscal) {
        this.numIdFiscal = numIdFiscal;
    }
    public String getMorada() {
        return morada;
    }
    public void setMorada(String morada) {
        this.morada = morada;
    }
    public int getNumTelemovel() {
        return numTelemovel;
    }
    public void setNumTelemovel(int numTelemovel) {
        this.numTelemovel = numTelemovel;
    }
    public String getData() {
        return data;
    }
    public void setData(String data) {
        this.data = data;
    }

```

```

}
public String getHora() {
    return hora;
}
public void setHora(String hora) {
    this.hora = hora;
}

public String getNomeEstafeta() {
    return nomeEstafeta;
}
public void setNomeEstafeta(String nomeEstafeta) {
    this.nomeEstafeta = nomeEstafeta;
}
public double getTotalIva() {
    return totalIVA;
}
public void setTotalIva(double totalIVA) {
    this.totalIVA = totalIVA;
}
public double getTotalPagar() {
    return totalPagar;
}
public void setTotalPagar(double totalPagar) {
    this.totalPagar = totalPagar;
}
public ArrayList<Item> getRefeicoes() {
    return refeicoes;
}
public void setRefeicoes(ArrayList<Item> refeicoes) {
    this.refeicoes = refeicoes;
}
public double getTotalIVA() {
    return totalIVA;
}
public void setTotalIVA(double totalIVA) {
    this.totalIVA = totalIVA;
}
public ArrayList<Item> getGratificacoes() {
    return gratificacoes;
}
public void setGratificacoes(ArrayList<Item> gratificacoes) {
    this.gratificacoes = gratificacoes;
}
public ArrayList<Item> getDescontos() {
    return descontos;
}
public void setDescontos(ArrayList<Item> descontos) {
    this.descontos = descontos;
}

// Funcao para adicionar uma linha de refeicao ao pedido
public void addRefeicao(Item refeicao) {
    refeicoes.add(refeicao);
}

// Funcao para adicionar uma linha de gratificacao ao pedido
public void addGratificacao(Item gratificacao) {

```

```

        gratificacoes.add(gratificacao);
    }

    // Funcao para adicionar uma linha de desconto
    public void addDesconto(Item desconto) {
        desconto.setPrecoUnitario(desconto.getPrecoUnitario());
        descontos.add(desconto);
    }

    // Funcao para imprimir o pedido
    void printPedido() {
        System.out.println("Ordem de servico realizada em: " + data + "@" + hora);
        System.out.println("Cliente: " + nomeCliente + "\t\tContribuinte: " + numIdFiscal);
        System.out.println("Morada: " + morada + "\tTelemovel: " + numTelemovel);
        System.out.println("Estafeta: " + nomeEstafeta);
        System.out.println("\nDescricao\tPreco U\tQTD\tIVA\tT IVA\tSub-total");
        System.out.println("REFEICOES:");
        Item it;
        for (int i = 0; i < refeicoes.size(); i++) {
            it = refeicoes.get(i);
            System.out.println(it.getItemString());
        }
        System.out.println("GRATIFICACOES:");
        for (int i = 0; i < gratificacoes.size(); i++) {
            it = gratificacoes.get(i);
            System.out.println(it.getItemString());
        }
        System.out.println("DESCONTOS:");
        for (int i = 0; i < descontos.size(); i++) {
            it = descontos.get(i);
            System.out.println(it.getItemString());
        }
        System.out.println("\t\t\t\t\tTot IVA\tTOTAL PAGAR");
        System.out.println("\t\t\t\t\t"+totalIVA+ "\t" + totalPagar);
    }
}

public class talao {
    public static void main(String[] args) {
        Pedido p;
        ArrayList<Item> refeicoes = new ArrayList();
        ArrayList<Item> gratificacoes = new ArrayList();
        ArrayList<Item> descontos = new ArrayList();
        //String descricao, double precoUnitario, double quantidade,
        double taxaIVA
        // Refeicoes
        Item ref1 = new Item("Pizza tipo 1", 10.0, 1.5, 20);
        Item ref2 = new Item("Massa assada", 5.0, 2.0, 10);
        // gratificacoes
        Item grat1 = new Item("Gorjeta emp 1", 10, 1, 0); // grats pagam IVA?
        // descontos
        Item desc1 = new Item("Cliente regular", 2, 1, 10);
        refeicoes.add(ref1);
        refeicoes.add(ref2);
        gratificacoes.add(grat1);
        descontos.add(desc1);
        p = new Pedido("Antonio Manel", 555444777, "Rua do alto, 8000",
        911191919, "2020/06/02", "16:00", "Ze de baixo", refeicoes, gratificacoes, descontos);
    }
}

```

```
    p.printPedido();  
  }  
}
```