

1º Teste de “Introdução à Programação” (2013/14 – 1º Semestre)

Duração 2H

Instruções importantes:

- Responda a cada pergunta no espaço atribuído para o efeito.
- Verifique que **todas** as folhas estão identificadas com o seu nome e número.
- Antes de começar a resolver, leia o enunciado do **princípio até ao fim**.
- **Pode** usar caneta ou lápis.
- Não é permitido consultar quaisquer elementos para além deste enunciado.
- Qualquer tentativa de fraude comprovada acarretará a reprovação na disciplina.
- Não é permitido sair da sala antes que o teste termine.

Número do Aluno: _____ Nome do Aluno: _____

Grupo I

Considere o seguinte enunciado do problema:

Objectivo:
Gerir as economias mensais dum estudante.

Descrição:
Um estudante para além da sua mensalidade (valor inteiro em Euros), pode realizar actividades remuneratórias que aumentam o seu poder económico. No entanto, durante o mês, tem diferentes gastos como refeições, renda, etc. Será sempre possível realizar um gasto, caso após a realização do gasto, o estudante ficar com pelo menos 5 Euros. Todo o dinheiro sobranete de um dado mês poderá ser utilizado nos meses seguintes.
Todos os montantes em Euros (receitas e gastos) são valores inteiros maiores do que zero.

Funcionalidades:

- Registrar a realização de uma actividade remuneratória, indicando o valor em Euros usufruído;
- Verificar se é possível realizar um gasto de uma dada quantia em Euros;
- Registrar a realização dum gasto, indicando o valor em Euros, caso seja possível a sua realização;
- Consultar o valor em Euros da mensalidade do mês corrente;
- Consultar o valor actual em Euros das economias do estudante;
- Registrar o início do mês, indicando o valor da mensalidade nesse mês;
- Para criar um objecto da classe é necessário indicar a mensalidade do mês corrente.

A. A classe `StudentCash` representa objectos que permitem gerir as economias mensais de um estudante. Indique a interface da classe `StudentCash`.

Para cada método indique o tipo do seu resultado e o tipo dos seus parâmetros (se existirem). Explique ainda a sua finalidade (para que serve) de forma clara e intuitiva, assim como as pré-condições. Indique ainda que métodos são modificadores e que métodos são de consulta. **(3 valores)**

B. Indique as variáveis de instância e constantes que achar necessárias para representar o estado de cada objecto da classe `StudentCash`. Para cada variável indique o seu tipo e explique a sua finalidade. **(2 valores)**

Número do Aluno: _____ Nome do Aluno: _____

Grupo II

Este exercício visa a construção de uma classe `OurRestaurant` que permite criar objectos que controlam as vendas num restaurante, que só tem a opção de 2 menus: carne ou peixe. Cada venda é paga no momento do pedido dos menus. Este pedido indica o número de menus de carne e de peixe desejados. Cada menu é vendido a 10 Euros, no entanto se forem pedidos mais de 2 menus de cada categoria, será realizado um desconto de 20% no preço de cada menu.

Assume-se que existem sempre menus para satisfazer as vendas, os quais são elaborados no momento do pedido.

Nestes restaurantes, as despesas dizem respeito: (1) aos valores em Euros da elaboração dos menus, específico de cada restaurante, e (2) ao preço a pagar ao Estado (taxa de IVA), comum a todos os restaurantes. As despesas associadas aos menus de peixe e de carne são indicadas no momento da criação do restaurante. O valor a pagar ao Estado é 23% do total de vendas realizadas.

A. Complete a seguinte implementação da classe `OurRestaurant` (7 valores):

```
public class OurRestaurant{
    /** Constantes **/
```

```
/** Variáveis de instância **/
```

```
/**
 * Constructor
 * @param menuFish preço do menu de peixe para o restaurante
 * @param menuMeat preço do menu de carne para o restaurante
 * @pre menuFish > 0 && menuMeat > 0
 */
```

```
public OurRestaurant(float menuFish, float menuMeat){
```

```
}
```

Número do Aluno: _____ Nome do Aluno: _____

```
/**
 * Fazer um pedido de menus, calcular preço total a cobrar,
 * tendo em conta os descontos caso existam (número de menus de peixe
 * superior a 2 e número de menus de carne superior a 2).
 *
 * @param numberOfFish número de menus de peixes pedidos
 * @param numberOfMeat número de menus de peixes pedidos
 * @pre numberOfFish >= 0 && numberOfMeat >= 0
 * @return valor a pagar em Euros
 */
public float request(int numberOfFish, int numberOfMeat){
```

```
}

/**
 * Consultar o número total de menus vendidos a preço normal
 *
 * @return número de menus vendidos a preço normal
 */
public int normalMenus(){
```

```
}

/**
 * Consultar o número total de menus vendidos a preço reduzido
 *
 * @return número de menus vendidos a preço reduzido
 */
public int reducedMenus(){
```

```
}

/**
 * Calcular o total de Euros em caixa resultante das vendas realizadas
 *
 * @return valor em Euros que está em caixa
 */
public float cash(){
```

```
}
```

Número do Aluno: _____ Nome do Aluno: _____

```
/**
 * Calcular o total de Euros a pagar ao estado,
 * com taxa de IVA a 23% sobre as vendas realizadas
 *
 * @return valor em Euros
 */
public float iva(){
```

```
}

/**
 * Calcular o lucro do restaurante (vendas - despesas), tendo como base
 * que as despesas são:
 * - valor em Euros gasto na elaboração dos menus de peixe e de carne
 *   (valores dados no constructor)
 * - valor em Euros pago ao Estado com a taxa de IVA a 23%
 *
 * @return valor em Euros
 */
public float profit(){
```

```
}

/**
 * Indicar se o restaurante teve lucro
 *
 * @return True, caso houve lucro
 */
public boolean hasProfit(){
```

```
}

/** Métodos Auxiliares, caso necessite **/
```

Número do Aluno: _____ Nome do Aluno: _____

}

B. Complete o seguinte programa principal, que simula o serviço num restaurante (4 valores).

```
import java.util.Scanner;
```

```
public class Main{
```

```
    public static void main(String [] args){
```

```
        OurRestaurant fctBar;
```

```
        Scanner s= new Scanner (System.in);
```

```
        float fish,meat;
```

```
        /* Pedir ao utilizador o preço do menu de peixe e  
        inicializar a variável respectiva */
```

```
        _____
```

```
        s.nextLine();
```

```
        System.out.println("Preco menu carne:");
```

```
        meat = s.nextFloat();
```

```
        s.nextLine();
```

```
        if ((fish > 0) && (meat > 0)){
```

```
            /* Criar o restaurante fctBar com os preços dados pelo utilizador */
```

```
            _____
```

```
            processRequest(s, fctBar);
```

```
        }
```

```
        s.close();
```

```
    }
```

```
private static void processRequest(Scanner s, OurRestaurant r){
```

```
    int nFish,nMeat;
```

```
    String answer;
```

```
    System.out.println("Deseja efectuar pedido (sim ou nao):");
```

```
    answer= s.next().toUpperCase();
```

```
    s.nextLine();
```

```
    while (answer.equals("SIM")){
```

```
        System.out.println("Numero de menus peixe:");
```

```
        nFish = s.nextInt();
```

```
        s.nextLine();
```

```
        System.out.println("Numero de menus carne:");
```

```
        nMeat = s.nextInt();
```

```
        s.nextLine();
```

```
        if ((nFish >= 0) && (nMeat >= 0))
```

```
            /* Processa o pedido e escreve o valor a pagar na consola */
```

```
            _____
```

```
            System.out.println("Deseja efectuar pedido (sim ou nao):");
```

```
            answer= s.next().toUpperCase();
```

```
            s.nextLine();
```

```
        }
```

```
    }
```

```
}
```

Número do Aluno: _____ Nome do Aluno: _____

Grupo III

Considere a classe WhatIsIt definida da seguinte forma:

```
public class WhatIsIt{
    public static final String MSG1 = "mensagem 1";
    public static final String MSG2 = "mensagem 2";
    public static final String MSG3 = "mensagem 3";
    public static final String MSG4 = "mensagem 4";
    public static final String MSGA = "mensagem a";
    public static final String MSGB = "mensagem b";
    public static final String MSG = "mensagem";
    public static final int CONSA = 10;
    public static final int CONSB = 10;

    private int a, b, c, d;

    public WhatIsIt(int x, int y){
        a = x;
        b = y;
        c = CONSA;
        d = CONSB;
    }
    public boolean method_1(){
        return (a == c) && (b == d);
    }
    public String method_2(int x){
        a = a + x;
        return method_4();
    }
    public String method_3(int y){
        b = b + y;
        return method_4();
    }
    private String method_4(){
        String msg = "";
        if (method_1())
            msg = MSG;
        else {
            msg = msg + method_4a();
            msg = msg + ";" + method_4b();
        }
        return msg;
    }
    private String method_4a(){
        String msg;
        if (a > c)
            msg = MSG1;
        else if (a < c)
            msg = MSG2;
        else
            msg = MSGA;
        return msg;
    }
    private String method_4b(){
        String msg;
        if (b > d)
            msg = MSG3;
        else if (b < d)
            msg = MSG4;
        else
            msg = MSGB;
        return msg;
    }
}
```

Número do Aluno: _____ Nome do Aluno: _____

A. Complete o seguinte traço de execução realizado no BlueJ (3 valores)

```
WhatIsIt j = new WhatIsIt(6,7);
j.method_1()
[ ] (boolean)
j.method_2(4)
[ ] (String)
j.method_3(9)
[ ] (String)
j.method_2(1)
[ ] (String)
j.method_2(-2)
[ ] (String)
j.method_3(-2)
[ ] (String)
j.method_2(1)
[ ] (String)
j.method_1()
[ ] (boolean)
j.method_3(-1)
[ ] (String)
j.method_3(-3)
[ ] (String)
```

B. Qual poderá ser o objectivo da classe WhatIsIt? (1 valor)