

Número:

Nome:

1º teste de Introdução à Programação

18 de Novembro de 2017

2 horas

Pergunta 1 (4 Valores)

(a) Considere a seguinte classe:

```
public class Enigma {
    public static final int NEUTRO = 1;

    private int origem;
    private int valor;

    public Enigma (int valor) {
        origem = valor;
        this.valor = NEUTRO;
    }
    public int proximo() {
        valor = valor * origem;
        return valor;
    }
    public void reiniciar() {
        valor = NEUTRO;
    }
}
```

Complete nos espaços o que seria escrito na consola após a execução de cada instrução na sequência de instruções dadas. Caso não seja escrito nada, escreva "Não escreve nada".

Enigma x = new Enigma(3);

System.out.println(x.proximo());

System.out.println(x.proximo());

System.out.println(x.proximo());

x.proximo();

System.out.println(x.proximo());

x.reiniciar();

System.out.println(x.proximo());

(b) Considere a seguinte classe:

Número:

Nome:

```
public class SegredoInt {
    private int n;
    /**
     * @pre: numero > 0
     */
    public SegredoInt(int numero) {
        n = numero;
    }

    /**
     * @pre: digito >= 0 && digito <= 9
     */
    public int daNumero(int digito) {
        int i = n;
        int cont = 0;
        while (i != 0) {
            if (i % 10 == digito)
                cont++;
            i = i / 10;
        }
        return cont;
    }
    public String converte() {
        int i = n;
        String result = "";
        while (i != 0) {
            result = (i % 2) + result;
            i = i / 2;
        }
        return result;
    }
}
```

Complete nos espaços o que seria escrito na consola após a execução de cada instrução na sequência de instruções dadas. Caso não seja escrito nada, escreva "Não escreve nada".

```
SegredoInt x = new SegredoInt(232);
```

```
SegredoInt y = new SegredoInt(56990);
```

```
System.out.println(x.converte());
```

```
System.out.println(x.daNumero(0));
```

```
System.out.println(y.daNumero(6));
```

```
System.out.println(x.daNumero(2));
```

Pergunta 2 (8 valores)

Implemente em Java a seguinte classe para deslocar uma entidade numa zona. A zona é uma superfície rectangular com um dado comprimento (comp) e uma dada

Número:

Nome:

altura (alt). A entidade está sempre localizada num ponto da zona definido por uma abcissa (valor inteiro no intervalo [1..comp]) e uma ordenada (valor inteiro no intervalo [1..alt]). Note que quer o comprimento quer a altura da zona têm que ser valores inteiros positivos maiores ou iguais a 10.

Antes de começar a programar, leia todos os comentários em baixo para compreender o funcionamento da classe, e a poder implementar.

```
public class Zona {
```

```
    /** Constantes */
```

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

```
    /** Cria zona com o comprimento e a altura dada.
```

```
    * Coloca a entidade na posição (comp/2,alt/2)
```

```
    * @param comp - comprimento da zona
```

```
    * @param alt - altura da zona
```

```
    * @pre:
```

```
    */
```

```
    public Zona(int comp, int alt) {
```

```
}
```

```
    /** Indica se as coordenadas (x,y) são coordenadas na zona. Ou seja, se estão
```

```
    * dentro da superfície rectangular que define a zona.
```

```
    * @param x - coordenada horizontal (abcissa)
```

```
    * @param y - coordenada vertical (ordenada)
```

```
    * @return true, caso estejam dentro da zona; false, caso contrário
```

```
    */
```

```
    public boolean estaNaZona(int x, int y) {
```

```
}
```

```
    /** Indica a abcissa da posição corrente da entidade
```

```
    * @return abcissa da posição da entidade
```

```
    */
```

```
    public int daAbcissaEntidade() {
```

Número:

Nome:

```
}  
/** Indica a ordenada da posição corrente da entidade  
* @return ordenada da posição da entidade  
*/  
public int daOrdenadaEntidade() {
```

```
}  
/** Coloca a entidade na coordenada (x,y) dada, e retorna a distância percorrida.  
* Esta distância obtém-se somando a distância vertical com a distância  
* horizontal entre duas posições: a corrente e a nova posição dada. Por  
* exemplo, se a entidade estiver na posição (2, 3) e a posição dada for (8, 1),  
* a distância é a soma da distância vertical (|2 - 8|, ou seja, 6) com a  
* distância horizontal (|3 - 1|, ou seja, 2). A distância total, neste caso,  
* seria 6+2, ou seja, 8.  
* @param x - abcissa da nova posição  
* @param y - ordenada da nova posição
```

```
* @pre:  
* @return distância percorrida no movimento  
*/  
public int moverEntidade(int x, int y) {
```

```
}  
/** Indica a distância total percorrida pela entidade.  
* @return distância total  
*/  
public int distanciaTotalEntidade() {
```

```
}  
/** Indica o número de movimentos feitos pela entidade.  
* @return número de movimentos  
*/  
public int numeroMovimentosEntidade() {
```

```
}  
/** Indica a maior distância percorrida pela entidade num dado movimento.
```

```
* @pre:  
* @return maior distância percorrida num movimento  
*/  
public int maximaDistanciaEntidade() {
```

Número:

Nome:

}

/** Cálculo do valor a cobrar à entidade pelo uso da zona. Dependendo da
* distância média realizada pela entidade nos seus movimentos, a entidade paga
* um dado valor em euros (valor real) por cada unidade de distância.

* Distância média por movimento (no intervalo)	Valor a pagar por unidade de distância (em Euros)
* [0..250]	0.75
*]250..345]	0.6
*]345..500]	0.5
* >500	0.45

* @return Valor a pagar

*/

public float custoEntidade() {

}

/** Métodos privados */

}

Pergunta 3 (4 valores)

Este programa principal usa a classe Zona referida na pergunta 2 (note que pode resolver esta questão mesmo que não tenha respondido à questão 2). Complete o código de acordo com os pedidos nos comentários colocados no código.

public class Main {

/** Leitura de movimentos da entidade na zona dada.

* A leitura acaba quando o utilizador indica a posição (0,0).

*/

Número:

Nome:

```
private static void lerMovimentos (Zona z){
    Scanner in = new Scanner(System.in);
    int nX=1;        // Abcissa
    int nY=1;        // Ordenada
    // Enquanto nX e nY diferentes de zero
    while
    {
        System.out.println("Indique as coordenadas para onde a entidade se vai mover:");
        nX = in.nextInt();
        nY = in.nextInt();
        in.nextLine();
        if
        {
            // Move a entidade para as coordenadas indicadas na zona z e escreve a
            // distância percorrida
        }
    } // fim while
    in.close();
}
/* Escreve alguns dados estatísticos referentes à entidade na zona dada */
private static void processarEstatisticas (Zona z){
    // Escreve na consola a distância total da entidade na zona dada
    // Escreve na consola a maior distância
    // Escreve na consola o valor a cobrar à entidade
    // Escreve posição final da entidade na consola: abcissa e ordenada da posição
}
/** Programa principal - leitura de movimentos e escrita de algumas estatísticas */
public static void main (String [] args){
    Zona zonaFCT = new Zona(30,40); // A zona a usar neste programa (ver pergunta 2)
    lerMovimentos(zonaFCT);
    if (zonaFCT.numeroMovimentosEntidade() > 0)
        processarEstatisticas(zonaFCT);
    else
        System.out.println("Sem movimentos.");
}
```

Pergunta 4 (4 valores)

Para o seguinte enunciado:

Objectivo: Simular um comilão.

Descrição e Funcionalidades

Um comilão é uma entidade que devora "sopa", "prato" e "sobremesa".

Número:

Nome:

De cada vez que come, ingere calorias de uma destas três alternativas e acumula essas calorias. O número de calorias ingeridas de cada vez depende do tipo de alimento ingerido.

É sempre possível saber quantas vezes o comilão comeu de cada tipo de alimento. Para além disso, é sempre possível saber o total de calorias acumuladas pelo comilão.

No momento de criação do comilão podem-se indicar as calorias associadas a cada tipo de alimento. Caso não seja indicado nada, assume-se que uma sopa tem 50 calorias, um prato tem 300 calorias e uma sobremesa tem 100 calorias. Por exemplo, um comilão que tivesse comido duas sopas, um prato e uma sobremesa teria ingerido 500 calorias.

(a) Que classe seria necessário definir e implementar?

(b) Indique os métodos da interface dessa classe. Não deve implementar, só indicar o seu nome o tipo de dados que devolve e os seus parâmetros (assinatura/cabeçalho do método). Para além disso indique, caso existam, as pré-condições.

(c) Que variáveis de instância teria na classe? Para cada uma delas indique o seu tipo e o que representa.