

Número:

Nome:

1º teste de Introdução à Programação

12 de Novembro de 2016

2 horas

Proposta de Resolução

Pergunta 1 (4 Valores)

(a) Considere a seguinte classe:

```
public class Enigma{
    public static final int INICIAL = 5;
    public static final int FACTOR = 2;

    private int origem;
    private int valor;

    public Enigma (int valor){
        origem = valor;
        this.valor = INICIAL;
    }
    public int inc(){
        origem = origem + valor;
        valor = FACTOR * valor;
        return origem;
    }
    public void reIniciar(){
        valor = INICIAL;
    }
}
```

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(10);
```

Não escreve nada

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

15

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

25

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

45

```
x.inc();
```

Não escreve nada

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

165

```
x.reIniciar();
```

Não escreve nada

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

170

Número:

Nome:

(b) Considere a seguinte classe:

```
public class SegredoString{
    private String s;
    /**
     * @pre segredo!=null && !segredo.equals("")
     */
    public SegredoString(String segredo) {
        s = segredo;
    }
    public int daNumero(char c) {
        int i = 0;
        int cont = 0;
        while (i < s.length()){
            if (s.charAt(i) == c) /*s.charAt(i) devolve o caracter com*/
                cont++; /*índice i. O primeiro caracter tem índice 0*/
            i++;
        }
        return cont;
    }
    public int maiorSub(){
        int i = 1;
        int contM=0;
        int contC = 1;
        char c = s.charAt(0); /*s.charAt(0) devolve o primeiro caracter
                               da String s*/
        while (i < s.length()){ /*s.length() devolve o número de */
            if (s.charAt(i) == c) /*caracteres da String s*/
                contC++;
            else {
                if (contM < contC)
                    contM = contC;
                contC =1;
                c= s.charAt(i);
            }
            i++;
        }
        if (contM < contC)
            contM = contC;
        return contM;
    }
}
```

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".

```
SegredoString x = new SegredoString("ola pessoal!");
```

Não escreve nada

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

2

```
System.out.println(x.daNumero('a'));
```

2

```
System.out.println(x.daNumero('A'));
```

0

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

1

Número:

Nome:

Pergunta 2 (8 valores)

Implemente em Java a seguinte classe para gerir a venda de bilhetes de comboio. Leia todos os comentários em baixo para compreender o funcionamento da classe.

```
public class Comboio {  
    /* Constantes */
```

```
private static final int CAPACIDADE = 1000;
```

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

```
private float precoBilhete, conta;  
private int capacidade, maiorFamilia, nPessoas;
```

```
/** Cria comboio com capacidade para 1000 pessoas com preço de bilhete dado  
 * @param precoBilhete - preço do bilhete dum adulto  
 * @pre: precoBilhete >= 0  
 */
```

```
    public Comboio(float precoBilhete) {
```

```
        capacidade = CAPACIDADE;  
        this.precoBilhete = precoBilhete;  
        conta = 0f;
```

```
    }
```

```
/** Cria comboio com a capacidade dada e o preço de bilhete dado  
 * @param precoBilhete - preço do bilhete dum adulto  
 * @param capacidade - lotação máxima  
 * @pre: precoBilhete >=0  
 */
```

```
    public Comboio(float precoBilhete, int capacidade) {
```

```
        this.capacidade = capacidade;  
        this.precoBilhete = precoBilhete;  
        conta = 0f;
```

```
    }
```

```
/** Indica a lotação máxima do comboio, ou seja a sua capacidade  
 * @return lotação máxima  
 */
```

```
    public int daLotacaoMaxima() {
```

```
        return capacidade;
```

```
    }
```

Número:

Nome:

```
/* Indica os lugares ainda vagos no comboio, assume-se que
 * tanto um adulto como uma criança ocupa um lugar.
 * @return lugares vagos
 */
```

```
    public int daLugaresVagos() {
```

```
        return capacidade - nPessoas;
```

```
    }
```

```
/* Venda de bilhetes para uma familia de nAdultos adultos e nCrianças
 * crianças. O preço do bilhete de criança é 1/3 do de adulto.
 * Note que é uma familia de nAdultos+nCrianças pessoas.
 * @pre: daLugaresVagos() >= nAdultos + nCrianças
 *      && nAdultos >= 0 && nCrianças >= 0 && nAdultos + nCrianças > 0
 * @return Valor a pagar
 */
```

```
    public float vendaBilhetes(int nAdultos, int nCrianças) {
```

```
        if (nAdultos + nCrianças > nPessoas)
```

```
            maiorFamilia = nAdultos + nCrianças;
```

```
        float valor = nAdultos * precoBilhete + nCrianças * precoBilhete * 1/3;
```

```
        conta += valor;
```

```
        return valor;
```

```
    }
```

```
/* Retorna o valor em caixa.
```

```
 * @return valor em caixa
```

```
 */
```

```
    public float caixa() {
```

```
        return conta;
```

```
    }
```

```
/* Retorna o número de pessoas (adultos e crianças) que compõem a família
```

```
 * maior, ou seja a venda de bilhetes para um maior número de pessoas.
```

```
 * Cuidado, que não é a venda de maior valor.
```

```
 * @pre: cash() > 0
```

```
 * @return número de pessoas da maior família
```

```
 */
```

```
    public int maiorFamilia() {
```

```
        return maiorFamilia;
```

```
    }
```

```
}
```

Número:

Nome:

Pergunta 3 (4 valores)

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

```
public static void main (String [] args){
    int numMax;           // Número máximo de passageiros
    int nA;               // Número de adultos
    int nC;               // Número de crianças
    String fim = "N";     // String para terminar o ciclo de leitura de dados
    Comboio comboioFCT;   // O comboio a usar neste programa (ver pergunta 2)
    Scanner in = new Scanner(System.in);

    System.out.println("Lotacao do comboio:");
    // ler valor inteiro que representa a lotação e criar o respectivo comboio
    // com um preço de bilhetes para adulto de 9 euros
    numMax = in.nextInt(); comboioFCT = new Comboio(9, numMax);
    in.nextLine();
    // Enquanto houver lugares vagos no comboio e fim diferente de "S"
    while
    {
        (comboioFCT.daLugaresVagos() > 0 && !fim.equals("S"))

        System.out.println("Escreve numero adultos e numero crianças:");
        nA = in.nextInt();
        nC = in.nextInt();
        in.nextLine();
        if (    nA >= 0 && nC >= 0 && (nA + nC > 0)
            && comboioFCT.daLugaresVagos() >= (nA+nC))
            // Vende os bilhetes pedidos e escreve na consola o valor a pagar
            // por esta família
            System.out.println(comboioFCT.vendaBilhetes(nA, nC));
        else
            System.out.println("Compra nao efectuada!");

        System.out.println("Deseja acabar? (S para sair): ");
        fim = in.next().toUpperCase();
        in.nextLine();
    }
    if (comboioFCT.caixa() > 0)
        // Escreve na consola o número de elementos da maior família
        System.out.println(comboioFCT.maiorFamilia());

    in.close();
}
```

Número:

Nome:

Pergunta 4 (4 valores)

Para o seguinte enunciado:

Objectivo: Simular uma calculadora.

Descrição e Funcionalidades

Na calculadora é possível realizar operações binárias com valores reais de grande precisão. As operações binárias são: *soma*, *subtração*, *multiplicação* e *divisão*. Para além disso é sempre possível consultar qual o valor da última conta/operação realizada, a qual se denota por “\$”. Mais ainda, é sempre possível usar esse valor como operando numa operação a calcular, ou seja calcular “5+\$” ou “\$ - 28.78”, etc.

Os cálculos realizados na calculadora envolvem a seguinte expressão: Um operador binário seguido de dois valores reais, podendo um deles ser o último valor. No caso da divisão assuma que o segundo valor nunca é zero. Por exemplo, para calcular “5+\$” a expressão seria “+ 5 \$”.

No início assume-se que o último valor calculado é zero.

(a) Que classe seria necessário definir e implementar?

Uma classe que realize as operacoes soma, subtração, multiplicação e divisão.

(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.

```
/* construtor - cria a calculadora */
public Calculadora()
/* somador - soma o valor dado ao valor da última conta realizada*/
public void Somador(float numero)
/* subtrator - subtrai o valor dado ao valor da última conta realizada */
public void Subtrator(float numero)
/* multiplicador - multiplica o valor da última conta realizada pelo valor dado */
public void Multiplicador(float numero)
/* divisor - divide o valor da última conta realizada pelo valor dado */
public void Divisor(float numero)
/* mostrador - mostra o valor da última conta realizada/devolve o valor de resultado */
public float Mostrador()
```

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

private float resultado; //o resultado da ultima conta realizada