



Fundação CECIERJ – Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação Orientada a Objetos
APX2 1º semestre de 2021.

Nome –

Assinatura –

Observações:

1. **As respostas devem ser entregues apenas como CÓDIGO FONTE JAVA, NUM ARQUIVO ÚNICO PARA CADA QUESTÃO, COM EXTENSÃO .java;**
2. Entregas fora do formato especificado, por exemplo, em arquivos com extensão “.pdf”, “.doc” ou outros, **não serão corrigidas**. Serão aceitas apenas soluções escritas na linguagem Java. Programas com erro de compilação não serão corrigidos. Use o Java na versão 8 da linguagem.
3. Quando o enunciado de uma questão inclui especificação de formato de entrada e saída, tal especificação deve ser seguida à risca pelo programa entregue. Atender ao enunciado faz parte da avaliação e da composição da nota final.
4. Os exemplos fornecidos nos enunciados das questões correspondem a casos específicos apontados para fins de ilustração e não correspondem ao universo completo de entradas possíveis especificado no enunciado. Os programas entregues devem ser elaborados considerando qualquer caso que siga a especificação e não apenas os exemplos dados. Essa é a prática adotada tanto na elaboração das ADs desta disciplina quanto no mercado de trabalho.
5. As APs são um mecanismo de avaliação individual. Para a correção desta avaliação utilizaremos um software que verifica a similaridade entre programas. Respostas plagiadas não serão corrigidas e serão zeradas.
6. **Não seja o plágio do que ouve, vê, lê por aí. Estilo é plagiar a si mesmo** (Última frase é atribuída a Alfred Hitchcock).

Questão 1) (5.0 pontos)

Considere o código abaixo ilustrando o uso de parte de um sistema OO para controle de procedimentos médicos a serem realizados num hospital.

```
public class AP2_2021_1_Q1 {
```

```

public static void main(String[] args) {
    Solicitacao sangue = new Exame("Exame de Sangue", 20);
    Solicitacao biopsia = new Cirurgia("Biopsia", 30);
    Tratamento analiseAlergia = new Tratamento();
    analiseAlergia.adicionaSolicitacao(Arrays.asList(sangue, biopsia));
    System.out.println(analiseAlergia);
    Solicitacao anestesia = new Medicamento("Anestesia", 1000);
    Solicitacao septo = new Cirurgia("Correção de Septo", 120);
    Tratamento correcaoSepto = new Tratamento();
    correcaoSepto.adicionaSolicitacao(anestesia);
    correcaoSepto.adicionaSolicitacao(septo);
    System.out.println(correcaoSepto);
    Solicitacao cisto = new Cirurgia("Extração de Cisto", 60);
    Tratamento extracaoCisto = new Tratamento();
    extracaoCisto.adicionaSolicitacao(Arrays.asList(analiseAlergia, anestesia,
cisto));
    System.out.println(extracaoCisto);
}
}

```

Esta parte do sistema hospitalar permite o registro de tratamentos, os quais podem conter a realização de exames, cirurgias e administração de medicamentos, além de outros tratamentos menores (como um subtratamento). Por exemplo, um tratamento que fosse uma cirurgia poderia conter um subtratamento que seria o pré-operatório (com medicamentos e exames), a cirurgia em si e um subtratamento pós-operatório, com procedimentos similares.

No código acima, na criação de um exame é informado, além do nome do exame, o tempo médio em minutos de realização do exame. O nome e tempo médio também são informados na criação de uma cirurgia. Na criação de um medicamento são informados o princípio ativo do medicamento e o seu custo. Por falar em custo, para um exame, o custo é de R\$ 20,00 / minuto, enquanto que para uma cirurgia o custo é de R\$ 100,00 / minuto.

Na impressão dos tratamentos deve-se listar o nome de cada procedimento existente, juntamente com seu preço. Esta listagem deve ser ordenada do maior para o menor valor de procedimento. Para esta ordenação utilize o `sort()` da interface `List` (<https://docs.oracle.com/javase/8/docs/api/java/util/List.html#sort-java.util.Comparator->).

De forma a obrigar que todos os procedimentos tenham um custo associado, permitindo que estes sejam corretamente exibidos nas listagens, crie uma interface chamada `Solicitacao` que contenha uma método chamado `custo()`, o qual retorna um valor do tipo `double`.

Segue a saída a ser obtida para o código acima.

```

Tratamento 0 {
Biopsia, 3000.0
Exame de Sangue, 400.0
}
Tratamento 1 {
Correção de Septo, 12000.0
Anestesia, 1000.0
}

```

```
Tratamento 2 {  
  Extração de Cisto, 6000.0  
  Tratamento 0 {  
    Biopsia, 3000.0  
    Exame de Sangue, 400.0  
  }  
  Anestesia, 1000.0  
}
```

Os números ordinais após cada tratamento são gerados pelo sistema de forma única. A repetição do Tratamento 0 no fim da listagem ocorre porque este foi reutilizado, como pode ser observado no código do enunciado.

Utilize os conceitos de OO sempre que possível e adequado, evitando redundâncias como códigos duplicados. Leia atentamente o código acima para inferir todas as classes e métodos necessários.

Questão 2) (5.0 pontos)

Considere o código a seguir que contabiliza números que aparecem num arquivo texto:

```
class Conta{  
    int num;  
    int quantidade_ocorrencias;  
    List<Integer> linhas = new ArrayList<Integer>();  
  
    Conta(int num, int linha){  
        this.num = num;  
        quantidade_ocorrencias = 1;  
        linhas.add(new Integer(linha));  
    }  
  
    public String toString(){  
        return num + " ";  
    }  
}
```

Escreva um programa em Java que receba **N** arquivos de entrada, que possuem vários números (que podem ser repetidos), separados por espaço em branco, em uma linha, e responda:

- (a) Qual(is) é(são) o(s) número(s) que mais repetem nos arquivos individualmente?
- (b) Qual(is) é(são) o(s) número(s) que não se repetem nos arquivos individualmente?
- (c) Qual(is) é(são) o(s) número(s) que estão em todos os arquivos?

(d) Qual(is) é(são) o(s) número(s) que aparecem somente em um dos arquivos de entrada?

Para um melhor entendimento, dados os seguintes dois arquivos de entrada:

1 2 3
1 3 5 4

2 4
7

As respostas para as questões supracitadas são as que seguem:

(a)	(b)	(c)	(d)
1 3	2 5 4	2 4	1 3 5
2 4 7	2 4 7		7

Sua resposta **DEVE** usar a classe **Conta** para resolver o problema. Ademais, **SEU PROGRAMA DEVE SER IMPLEMENTADO USANDO OS CONCEITOS DE OO. ALÉM DISSO, SEU PROGRAMA SÓ PODE LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ, E ELE DEVE EXECUTAR COM QUAISQUER ARQUIVOS, SEGUINDO O FORMATO DE ENTRADA SUPRACITADO, COMO PARÂMETRO DE ENTRADA. SE O SEU PROGRAMA RESOLVER SOMENTE O EXEMPLO SUPRACITADO, SUA QUESTÃO SERÁ TOTALMENTE DESCONTADA.**