



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação Orientada a Objetos**

**AP3 1º semestre de 2019.**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
- 

**Questão 1) (5.0 pontos)**

Considere a classe abaixo a ser usada num gerenciador para manipulação de tarefas:

```
class Tarefa {
    String descricao;
    boolean feita;

    public Tarefa(String descricao) {
        this.descricao = descricao;
        this.feita = false;
    }
}
```

- a) (1.0) Implemente os métodos realizar(), que conclui esta tarefa, e toString(), que retorna uma string representando a tarefa.
- b) (2.0) Defina uma classe chamada Recorrente, a qual é um tipo de tarefa que pode ocorrer com alguma frequência (diária, semanal ou mensal, nesta questão). A realização de uma tarefa recorrente representa apenas postergá-la para o próximo ciclo, dependendo de sua frequência. Utilize a classe LocalDate, a qual define métodos para adição versáteis, como plusDays(), plusWeeks(), plusMonths() e plusYears().
- c) (2.0) Implemente uma classe chamada Tarefas, onde cada objeto deve conter um conjunto de objetos do tipo Tarefa. Além disso, devem ser implementados métodos para adição de uma tarefa, realização de todas as tarefas e impressão destas tarefas na tela.

RESPOSTA:

```

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

class Tarefa {
    String descricao;
    boolean feita;

    public Tarefa(String descricao) {
        this.descricao = descricao;
        this.feita = false;
    }

    public void realizar() {
        this.feita = true;
    }

    public String toString() {
        return this.descricao + " : " + (this.feita ? "feita" : "não feita");
    }
}

class Recorrente extends Tarefa {
    LocalDate dia;
    tFrequencia frequencia;
    static enum tFrequencia {DIARIA, SEMANAL, MENSAL};

    public Recorrente(String descricao, LocalDate dia, tFrequencia frequencia) {
        super(descricao);
        this.dia = dia;
        this.frequencia = frequencia;
    }

    public void realizar() {
        switch (this.frequencia) {
            case DIARIA:
                this.dia = this.dia.plusDays(1);
                break;
            case SEMANAL:
                this.dia = this.dia.plusWeeks(1);
                break;
            case MENSAL:
                this.dia = this.dia.plusMonths(1);
                break;
            default:
                break;
        }
    }
}

class Tarefas {
    List<Tarefa> tarefas;

    public Tarefas() {
        tarefas = new ArrayList<Tarefa>();
    }

    public void adicionaTarefa (Tarefa t) {
        tarefas.add(t);
    }

    public void realizaTarefas () {
        for (Tarefa t : tarefas)
            t.realizar();
    }
}

```

```

    public void imprimirTarefas () {
        for (Tarefa t : tarefas)
            System.out.println(t);

        // Versão funcional (Não apresentada no curso!)
        //tarefas.forEach(System.out::println);
    }
}

```

### Questão 2) (5.0 pontos)

O gerenciador de arquivos de um computador apresenta uma árvore composta por elementos, que podem ser pastas (diretórios) ou arquivos. Uma pasta tem um nome e pode ter dentro dela outras pastas ou arquivos. Um arquivo não tem sub-elementos, representando uma folha na árvore. Cada arquivo tem um nome e um tamanho em bytes. O tamanho da pasta é calculado recursivamente pela soma do tamanho dos elementos que ela contém. Faça um programa em Java para representar tal estrutura e permitir o cálculo do tamanho de qualquer elemento usando adequadamente o paradigma orientado a objetos. O exemplo a seguir do uso desse programa deveria exibir 1200 na tela.

```

Pasta p1 = new Pasta("dir1");
p1.add(new Arquivo("arquivo1.txt", 150));
p1.add(new Arquivo("arquivo2.txt", 200));
Pasta p2 = new Pasta("dir2");
p2.add(new Arquivo("arquivo3.txt", 500));
Pasta p3 = new Pasta("dir3");
p3.add(new Arquivo("arquivo4.txt", 350));
p3.add(p2);
Pasta raiz = new Pasta("c:/");
raiz.add(p1);
raiz.add(p3);
System.out.println(raiz.getTamanho());

```

RESPOSTA:

```

import java.util.*;

abstract class Elemento{
    private String nome;

    public Elemento(String nome){ this.nome = nome; }

    public abstract int getTamanho();
}

class Arquivo extends Elemento{
    private int tamanho;

    public Arquivo(String nome, int tamanho){
        super(nome);
        this.tamanho = tamanho;
    }

    public int getTamanho(){
        return tamanho;
    }
}

```

```
}
```

```
class Pasta extends Elemento{  
    private List<Elemento> elementos = new ArrayList<Elemento>();  
  
    public Pasta(String nome){ super(nome); }  
  
    public void add(Elemento elemento){ elementos.add(elemento); }  
  
    public int getTamanho(){  
        int tamanho = 0;  
        for(Elemento elemento:elementos) tamanho += elemento.getTamanho();  
        return tamanho;  
    }  
}
```