Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
AD2 de Programação Orientada a Objetos
2° semestre de 2019

Nome:
Matrícula:
Pólo:

Considere que queiramos implementar um simulador de sistemas de arquivos, como o Explorer no Windows ou o Finder no MacOS. Um sistema de arquivos é composto de elementos, os quais podem ser pastas e/ou arquivos. Pastas, por sua vez, podem ter outras pastas e/ou arquivos internamente. Pastas e Arquivos possuem nomes. Arquivos também possuem um campo que informa o seu tamanho. Considere a sequência de chamadas abaixo, que caracteriza o uso deste simulador:

```
1: public class AD2_2019_2 {
2:     public static void main(String[] args) {
3:           Pasta p1 = new Pasta("dir1");
4:           p1.adiciona(new Arquivo("arquivo1.txt", 150));
5:           p1.adiciona(new Arquivo("arquivo2.txt", 200));
6:           Pasta p2 = new Pasta("dir2");
7:           p2.adiciona(new Arquivo("arquivo3.txt", 500));
8:           Pasta p3 = new Pasta("dir3");
9:           p3.adiciona(new Arquivo("arquivo4.txt", 350));
10:          p3.adiciona(p2);
11:          Pasta raiz = new Pasta("c:/");
12:          raiz.adiciona(p1);
13:          raiz.adiciona(p3);
14:          raiz.remove("dir1/arquivo1.txt");
15:          System.out.println(raiz.getTamanho());
16:          System.out.println(raiz);
17:    }
18:}
```

a) Implemente as classes Pasta e Arquivo, com seus atributos e construtores.
b) Implemente o método *adiciona()*. Na linha 4, por exemplo, está sendo adicionado ao diretório *dir1* um arquivo chamado *arquivo1.txt* com tamanho de 150 bytes.
c) Implemente o método *getTamanho()*, chamado na linha 14, sabendo que para um arquivo o método retorna o seu tamanho, enquanto que para uma pasta o método retorna a soma do seu conteúdo (soma dos tamanhos dos arquivos + soma dos conteúdos das subpastas). Para o exemplo acima, o valor impresso na linha 15 é 1050.
d) Implemente o método *remove()*, o qual remove um elemento (arquivo ou pasta) do sistema.

e) Implemente um método de impressão chamado à partir do método *toString()* (linha 16), o qual imprima a saída do programa dado como abaixo. Ou seja, indente a saída (imprimir com tabulação - "\t") de forma a respeitar a hierarquia no qual os elementos (arquivos e pastas) foram criados.

```
c:/
	dir1
		arquivo2.txt

	dir3
		arquivo4.txt
		dir2
			arquivo3.txt
```

Obs.: Ao longo de toda a resolução, use os conceitos de OO apresentados nas vídeo-aulas de forma a evitar redundância no código. Por exemplo, evitar a definição de atributos iguais, com a mesma funcionalidade, em classes diferentes.

RESPOSTA:

```
import java.util.*;

abstract class Elemento {
	private String nome;

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

	public abstract int getTamanho();

	public abstract String imprime(int nivel);

	public String getNome() { return this.nome; }

	public String indenta (int nivel) {
		return new String(new char[nivel]).replace("\0", "\t");
	}
}

class Arquivo extends Elemento {
	private int tamanho;

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

	public int getTamanho() {
		return tamanho;
	}

	public String toString() {
		return this.getNome();
```

```
        }

        public String imprime(int nivel) {
                return this.indenta(nivel) + this.toString();
        }
}

class Pasta extends Elemento {
        private List<Elemento> elementos = new ArrayList<Elemento>();

        public Pasta(String nome) {
                super(nome);
        }

        public void adiciona(Elemento elemento) {
                elementos.add(elemento);
        }

        public Elemento remove(String elemento) {
                Elemento alvo = null;
                String subelementos[] = elemento.split("/");

                for (Elemento e : this.elementos)
                        if (subelementos[0].equals(e.getNome()))
                                if (subelementos.length == 1) {
                                        alvo = e;
                                        this.elementos.remove(e);
                                        break;
                                }
                                else
                                        alvo = ((Pasta)
e).remove(elemento.substring(elemento.indexOf('/')+1));

                return alvo;
        }

        public int getTamanho() {
                int tamanho = 0;
                for (Elemento elemento : elementos)
                        tamanho += elemento.getTamanho();
                return tamanho;
        }

        public String imprime(int nivel) {
                String saida = indenta(nivel) + this.getNome() + "\n";
                for (Elemento e : this.elementos)
                        saida += e.imprime(nivel+1) + "\n";
                return saida;
        }

        public String toString() {
                return this.imprime(0);
        }
}

public class AD2_2019_2 {
        public static void main(String[] args) {
                Pasta p1 = new Pasta("dir1");
                p1.adiciona(new Arquivo("arquivo1.txt", 150));
                p1.adiciona(new Arquivo("arquivo2.txt", 200));
                Pasta p2 = new Pasta("dir2");
```

```
                p2.adiciona(new Arquivo("arquivo3.txt", 500));
                Pasta p3 = new Pasta("dir3");
                p3.adiciona(new Arquivo("arquivo4.txt", 350));
                p3.adiciona(p2);
                Pasta raiz = new Pasta("c:/");
                raiz.adiciona(p1);
                raiz.adiciona(p3);
                raiz.remove("dir1/arquivo1.txt");
                System.out.println(raiz.getTamanho());
                System.out.println(raiz);
        }
}
```