



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
AD1 de Programação Orientada a Objetos
2º semestre de 2018

Nome:

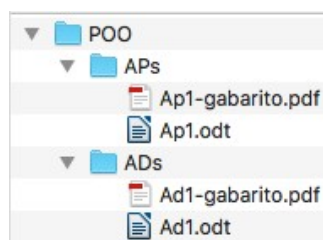
Matrícula:

Pólo:

Considere a classe abaixo para modelar um arquivo de um sistema de diretórios:

```
class Arquivo {  
    String nome;  
    String extensao;  
    double tamanho;  
}
```

- Crie um construtor para classe.
- Crie um método toString, redefinido da classe Object, de forma a retornar o nome do arquivo (concatenação do nome + . + extensão) e seu tamanho.
- Crie um método para retornar o tamanho do arquivo.
- Crie uma classe chamada Diretorio. Está deverá ter campos para armazenar o nome do diretório, um vetor de arquivos e outro vetor de diretórios (subpastas).
- Crie uma construtor para esta nova classe.
- Crie um método toString de forma a retornar o conteúdo do diretório: nome + lista de arquivos + lista de diretórios.
- Crie um método para retornar o tamanho do diretório (soma dos tamanhos dos arquivos do diretório e das subpastas – busca recursiva).
- Como um diretório pode conter arquivos e diretórios, simplifique esta implementação da seguinte maneira: implemente uma classe Item abstrata, a qual conterá um nome (arquivo ou diretório) e um método abstrato de obtenção do tamanho. Faça as classes Arquivo e Diretorio estenderem a classe Item, e façam as alterações necessárias para que o programa funcione.
- Altere os métodos toString para exibirem também o tamanho dos itens (para arquivos, o tamanho do respectivo; para diretórios, a soma dos tamanhos dos arquivos que contém e das subpastas).
- Crie um código de teste com valores representando a seguinte estrutura de diretórios:



GABARITO:

```
abstract class Item {
    String nome;
    abstract double getTamanho();
}

class Arquivo extends Item {
    String nome;
    String extensao;
    double tamanho;

    public Arquivo(String nome, String extensao, double tamanho) {
        this.nome = nome;
        this.extensao = extensao;
        this.tamanho = tamanho;
    }

    public String toString() {
        return this.nome + "." + this.extensao + ", " + this.tamanho + "\n";
    }

    double getTamanho() {
        return this.tamanho;
    }
}

class Diretorio extends Item {
    String nome;
    // Removido em função da inserção da classe Item
    // Arquivo conteudo[];
    // Diretorio subpastas[];
    Item conteudo[];

    public Diretorio(String nome, Item[] conteudo/*, Diretorio[]
subpastas*/) {
        this.nome = nome;
        this.conteudo = conteudo;
        // Removido em função da inserção da classe Item
        // this.subpastas = subpastas;
    }

    public String toString() {
        String saida = "Dir: " + this.nome + " s = " + this.getTamanho()
+ "\n";

        if (this.conteudo != null)
            for (Item i : conteudo) {
                saida += i;
            }
        // Removido em função da inserção da classe Item
        // if (this.subpastas != null)
        // for (Diretorio d : subpastas) {
```

```

//                saida += d;
//            }
            return saida;
        }

        double getTamanho() {
            double total = 0;
            for (Item i : conteudo) {
                total += i.getTamanho();
            }
            return total;
        }
    }

    public class AD1_2018_2 {
        public static void main(String[] args) {
            Arquivo ad1 = new Arquivo("Ad1", "odt", 20.7);
            Arquivo gab_ad1 = new Arquivo("Ad1-gabarito", "pdf", 55.3);
            Item ads[] = {ad1, gab_ad1};
            Diretorio dir_ads = new Diretorio("ADs", ads);

            Arquivo ap1 = new Arquivo("Ap1", "odt", 45.3);
            Arquivo gab_ap1 = new Arquivo("Ap1-gabarito", "pdf", 60.2);
            Item aps[] = {ap1, gab_ap1};
            Diretorio dir_aps = new Diretorio("APs", aps);

            Diretorio dir_poo[] = {dir_ads, dir_aps};
            Diretorio dir3 = new Diretorio("P00", dir_poo);

            System.out.println(dir3);
        }
    }

```