

# Primeira Prova de MAB 240 — Computação II

Fabio Mascarenhas

14 de Outubro de 2016

A prova é individual e sem consulta. Responda as questões na folha de respostas, a lápis ou a caneta. Se tiver qualquer dúvida consulte o professor.

Nome: \_\_\_\_\_

DRE: \_\_\_\_\_

Questão:	1	2	3	4	5	Total
Pontos:	1	2	2	2	3	10
Nota:						

Um sistema de arquivos possui três tipos de arquivos: *pastas*, que são conjuntos de arquivos, *arquivos de dados*, que são simples vetores de bytes, e *links*, que são arquivos especiais que apontam para algum outro arquivo do sistema de arquivos. Podemos definir classes para cada um dos tipos, todas implementando a interface `Arquivo` (datas são strings como “2016-10-14”, e a função `Data.hoje()` retorna a data de hoje):

```
class Pasta implements Arquivo {
    String nome;
    String data;
    HashSet<Arquivo> conteudo;
    public Pasta(String nome, Arquivo... arqs) {
        // questão 2
    }
}

class ArqDados implements Arquivo {
    String nome;
    byte[] dados;
    String data;
    public ArqDados(String nome, byte[] dados) {
        this.nome = nome; this.dados = dados;
        this.data = Data.hoje();
    }
}

class Link implements Arquivo {
    String nome;
    Arquivo destino;
    String data;
    public Link(String nome, Arquivo destino) {
        this.nome = nome; this.destino = destino;
        this.data = Data.hoje();
    }
}
```

1. (1 ponto) Mostre como criar instâncias de cada uma dessas três classes.
2. (2 pontos) Implemente o construtor da classe `Pasta`.

3. (2 pontos) Suponha que a interface `Arquivo` possua um método `int tamanho()`, que retorna o tamanho do arquivo; o tamanho de um arquivo de dados é o número de bytes que ele guarda, o tamanho de uma pasta é a soma do tamanho de todos os arquivos que ela contém, e o tamanho de um link é sempre 0. Implemente o método `tamanho` das três classes.

A interface `Predicado` dada a seguir representa *predicados* (no sentido matemático, de função booleana) sobre arquivos:

```
interface Predicado {
    boolean testa(Pasta p);
    boolean testa(Link l);
    boolean testa(ArqDados ad);
}
```

4. (2 pontos) Implemente a classe `MaisRecente` que implementa `Predicado` e recebe uma string representando uma data em seu construtor. Os métodos `testa` desse predicado retornam `true` sempre que a data do arquivo correspondente for mais recente que essa data. Use o método `compareTo(String outra)` da classe `String` para fazer a comparação, esse método retorna -1 se a string for menor que a outra, 0 se forem iguais e 1 se a string for maior.
5. (3 pontos) Suponha que a interface `Arquivo` possua um método `List<Arquivo> busca(Predicado pred)`, que retorna uma lista com todos os arquivos para o qual o predicado é verdadeiro (ou uma lista vazia caso não existam arquivos com o predicado verdadeiro). Implemente esse método nas classes `Pasta`, `ArqDados` e `Link`.

BOA SORTE!