

Número:

Nome:

## 2º teste de Introdução à Programação

21 de Dezembro de 2016

2:30 horas

**Notas prévias:** O teste é individual, sem consulta. Pode responder a lápis.

### Enunciado do problema

Na época natalícia quase todas as casas portuguesas têm uma árvore de Natal, com montes de prendas de diferentes cores, tamanhos e para diferentes pessoas.

Na realidade, uma dada pessoa pode ter várias prendas (em geral, o número previsto de prendas por pessoa não costuma exceder 10, embora isso possa acontecer). Mas há famílias que são tão organizadinhas que colocam todas as prendas de uma dada pessoa numa caixa, denominada de “caixa surpresa”.

Cada caixa surpresa é de uma pessoa e contém um conjunto não vazio de prendas. Numa árvore de natal existem zero ou mais caixas surpresa. Neste Natal decidiu-se limitar o número de caixas surpresa a 20, ou seja, numa árvore só podem existir no máximo 20 caixas surpresa. Para além disso, de forma a facilitar a organização das caixas surpresa, elas estão ordenadas por ordem crescente de idade. Em caso de empate na idade, por estão ordenadas por ordem da sua criação, ou seja, as mais antigas ficam antes das mais recentes.

Cada prenda é para uma dada pessoa, e é caracterizada por uma descrição. A pessoa é identificada pelo seu nome e caracterizada pela sua idade e faixa etária (criança, jovem, adulto e idoso).

Cada vez que se coloca uma prenda para uma pessoa na árvore de natal, deve-se colocar essa prenda na respectiva caixa surpresa. Caso ainda não exista a caixa, cria-se uma caixa surpresa para essa pessoa e coloca-se a prenda na caixa. A organização das prendas na caixa é pela ordem de chegada/inserção. Às vezes é necessário “aldrabar” a ordem de chegada e inserir numa dada posição. Por exemplo, se tivermos 3 prendas (p1, p2 e p3) e quisermos inserir uma nova prenda (p4) na segunda posição, a ordem das prendas na caixa ficaria (p1, p4, p2 e p3).

Por vezes é necessário retirar uma caixa surpresa da árvore de natal. Por exemplo se um dos familiares se portou mal o Pai Natal pode não lhe querer dar prendas. Neste caso as prendas da caixa a eliminar serão distribuídas pelas caixas surpresas existentes da mesma faixa etária.

A distribuição consta em transferir de forma equilibrada as prendas para as caixas surpresa da mesma faixa etária. Caso não existam mais caixas dessa faixa etária, não se faz nenhuma distribuição de prendas. Caso existam caixas, mas não seja possível distribuir o mesmo número por todas essas caixas, deve-se garantir que a diferença entre o número máximo de prendas atribuídas a caixas da mesma faixa etária e o número mínimo de prendas atribuídas a caixas dessa faixa não excede 1. Por exemplo se a caixa a eliminar tiver 7 prendas e for de uma pessoa idosa, e na árvore só existirem mais 4 caixas surpresas dessa faixa etária com as idades de 80, 82, 87 e 90, respectivamente, então as pessoas de 80, 82 e 87 anos ficam com mais 2 prendas cada uma, e a de 90 anos com mais uma prenda.

Os alunos do MIEI já definiram as classes do problema: `ArvoreNatal`, `CaixaSurpresa`, `Prenda`, `Pessoa` e `Main`.

Assuma que as classes `Prenda` e `Pessoa` estão implementadas:

Número:

Nome:

```
/**
 * A classe Prenda representa o conceito de prenda.
 */
public class Prenda {
    private String descricao; // Descricao da prenda
    /**
     * Cria uma nova prenda com descricao desc
     * @param desc a descricao da nova prenda
     */
    public Prenda(String desc){...}
    /**
     * Devolve a descricao da prenda
     * @return a descricao da prenda
     */
    public String daDescricao(){...}
}

/**
 * A classe Pessoa representa uma pessoa da festa de Natal.
 */
public class Pessoa {
    // Constantes para definir as faixas etárias.
    public static final int CRIANCA = 0;
    public static final int JOVEM = 1;
    public static final int ADULTO = 2;
    public static final int IDOSO = 3;

    // Variaveis de instancia
    private String nome; // nome da pessoa
    private int idade; // idade da pessoa

    /**
     * Cria uma pessoa com determinado nome e idade
     * @param nome o nome da pessoa a criar
     * @param idade a idade da pessoa a criar
     * @pre nome != null && idade >= 0
     */
    public Pessoa(String nome, int idade){...}

    /**
     * Devolve o nome da pessoa
     * @return o nome da pessoa
     */
    public String daNome(){...}

    /**
     * Devolve a idade da pessoa
     * @return a idade da pessoa
     */
    public int daIdade(){...}

    /**
     * Devolve a faixa etaria da pessoa (uma das constantes definidas nesta
     * classe). A faixa etaria e calculada com base na idade da pessoa.
     * @return a faixa etaria da pessoa
     */
    public int daEtaria(){...}
    //...
}
}
```

Número:

Nome:

**Grupo 1 (7 Valores):**

Complete a implementação da classe CaixaSurpresa preenchendo as áreas para si reservadas (rectângulos em branco).

```
// Esta classe representa objectos do tipo caixa surpresa
public class CaixaSurpresa {
    public static final int PRENDAS_PREVISTAS=10;
    private Pessoa dono; // O destinatario da caixa surpresa
    private Prenda[] prendas; // O conjunto de prendas na caixa surpresa
    private int contaPrendas; // O numero de prendas na caixa
    private int actual; // A posicao actual do iterador de prendas

    // Cria uma nova caixa surpresa, dado o nome do destinatario da caixa
    // surpresa, a sua idade e uma descricao da primeira prenda a colocar
    // na caixa. Pre: nome!=NULL && idade >= 0 && desc != NULL
    public CaixaSurpresa (String nome, int idade, String desc){
        prendas= new Prenda[PRENDAS_PREVISTAS];
        prendas[0]= new Prenda(desc);
        contaPrendas = 1;
        dono = new Pessoa(nome,idade);
        actual = -1;
    }

    //Métodos privados, caso necessite
```

```
// Devolve o numero de prendas na caixa
public int numeroPrendas(){
    return contaPrendas;
}
```

```
// Acrescenta uma prenda a caixa, dada a descricao dessa prenda. Se
// necessario, aumenta a sua capacidade da caixa. Pre: desc!=NULL
public void adicionaPrenda(String desc){
```

```
}
```

Número:

Nome:

```
// Adiciona a prenda descrita no parametro desc na posicao pos da caixa de
// prendas. Pre: desc!= NULL && pos>0 && pos <= numeroPrendas()
public void adicionaPrenda(String desc, int pos){
```

```
}
```

```
// Devolve o nome do dono da caixa surpresa.
public String daNomeDono(){
```

```
}
```

```
// Devolve a idade do dono da caixa surpresa.
public int daIdadeDono(){
```

```
}
```

```
// Devolve a faixa etaria do dono da caixa surpresa.
public int daEtariaDono(){
```

```
}
```

```
// Inicializa o iterador de prendas da caixa surpresa.
public void iniciaIterador(){
```

```
    actual =0;
```

```
}
```

```
// Testa se existem mais prendas a visitar com o iterador de prendas da
// caixa surpresa, retornando true caso existam, ou false, caso contrario.
```

```
public boolean temSeguinte(){
    return actual >=0 && actual<contaPrendas;
```

```
}
```

```
// Devolve a proxima Prenda a iterar na caixa surpresa.
```

```
public Prenda seguinte(){
    return prendas[actual++];
```

```
}
```

```
}
```

Número:

Nome:

**Grupo 2** (10 valores)

Implemente em Java a classe ArvoreNatal:

// A classe ArvoreNatal implementa a arvore de Natal, que deve ter uma coleccao  
// de ate 20 caixas surpresa

```
public class ArvoreNatal {  
    // Constantes
```

```
    // Variaveis de instancia
```

```
private int actual; // usada no iterador ja implementado
```

```
    // Cria uma arvore de natal, que começa sem caixas surpresa.
```

```
public ArvoreNatal(){
```

```
        actual = -1; // usada no iterador ja implementado
```

```
}
```

```
    // Metodos privados, caso necessite
```

Número:

Nome:

```
// Testa se existe alguma caixa surpresa para a pessoa com o nome passado
// por argumento. Pre: nome!=NULL
public boolean temCaixaSurpresa(String nome){
```

```
}
```

```
// Devolve o numero de prendas na caixa surpresa da pessoa cujo nome e
// passado como argumento. Pre: nome!=NULL && temCaixaSurpresa(nome)
public int numeroPrendasCaixaSurpresa(String nome){
```

```
}
```

```
// Adiciona uma nova prenda na caixa da pessoa cujo nome e passado como
// argumento. Se essa pessoa ainda nao tiver uma caixa, sera criada uma no
// momento, usando, alem do nome, a idade da pessoa e a descricao da
// prenda. Note que a colecao deve estar organizada por ordem crescente
// de idade. Em caso de empate de idades, as caixas mais "antigas"
// aparecem primeiro. Caso tenha de se criar uma caixa surpresa mas o
// limite (20 caixas) ja tenha sido atingido, o metodo não faz nada.
// Pre: nome!=NULL && idade > 0 && desc!=NULL
public void adicionaPrenda(String nome, int idade, String desc){
```

```
}
```

Número:

Nome:

```
// Semelhante ao anterior, mas adiciona a prenda numa posicao pos.  
// Pre: nome!=NULL && temCaixaSurpresa(nome) && pos>0 &&  
// pos <= numeroPrendasCaixaSurpresa(nome)  
public void adicionaPrenda(String nome, String desc, int pos){
```

```
}
```

```
// Retira a caixa da pessoa com o nome passado como argumento. As prendas  
// da caixa a eliminar sao distribuidas de forma equilibrada pelas caixas  
// existentes da mesma faixa etaria. Caso nao existam mais caixas dessa  
// faixa etaria, nao se faz nenhuma distribuicao de prendas. Caso  
// nao seja possivel distribuir o mesmo numero de prendas por todas as  
// caixas, deve-se garantir que a diferenca entre o numero maximo de  
// prendas atribuidas a caixas da mesma faixa etaria e o numero minimo de  
// prendas atribuidas a caixas dessa faixa não excede 1.  
// Pre: nome != NULL && temCaixaSurpresa(nome)  
public void retiraCaixaSurpresa(String nome){
```

```
}
```

Número:

Nome:

```
// Iterador de caixas surpresa - assuma que esta feito e use se e onde
// necessitar neste teste. O iterador itera os elementos organizados por
// idade crescente e, em caso de empate, por antiguidade das caixas.
public void iniciaIterador(){...} // inicializa o iterador

public boolean temSeguinte(){...} // testa se existem elementos a visitor

public CaixaSurpresa seguinte(){...} // visita elemento seguinte
}
```

**(pode usar o resto desta página como folha de rascunho, se necessitar...)**

Número:

Nome:

**Pergunta 3** (3 valores)

No programa principal é necessário escrever num dado ficheiro o nome e idade das crianças que têm caixas surpresa numa dada árvore de Natal. Em cada linha do ficheiro é escrito o nome e a idade de uma dada criança, separada por “;”. Caso não existam crianças não se escreve nada. Implemente o método seguinte, existente na classe `Main`.

```
// Escreve no ficheiro de nome nomeFicheiro o nome de todas as crianças com
// caixa surpresa na árvore de natal minhaArv.
private static void escreveCrianças(ArvoreNatal minhaArv, String nomeFicheiro)
    throws FileNotFoundException {
```

```
}
```