

Exercícios – Filas Dinâmicas

1) Considere o vetor a seguir:

0	5	7	-4	3	5	-2	-1	10	4	3	-6	2	-9	1	-5
---	---	---	----	---	---	----	----	----	---	---	----	---	----	---	----

Faça o teste de mesa do algoritmo:

```
Fila f = new Fila();
```

```
Para (int valor : vetor) {
```

```
    Se (valor >= 0) {
```

```
        f.insert(valor + 100);
```

```
    } senao {
```

```
        f.insert(valor * 2);
```

```
    }
```

```
}
```

```
Enquanto (!f.isEmpty()) {
```

```
    int num = f.remove();
```

```
    Se (valor >= 0) {
```

```
        escreva(num * 3);
```

```
    } senao {
```

```
        escreva(num);
```

```
    }
```

```
    escreva(f.size());
```

```
}
```

2) Criar um projeto Java que implementa com a biblioteca FilaStrings para simular um identificador de chamadas telefônicas enquanto o aparelho está fora da rede ou desligado. A aplicação deve ter uma classe no package controller, TelefoneController que tem os seguintes métodos:

- `insereLigacao(Fila f, String numeroTelefone): void`, que insere números de telefone em uma fila iniciada
- `consultaLigacoes(Fila f):void`, que desenfilera cada ligação realizada e exibe no console. Exibir uma exceção caso não haja ligação

Deve ter também uma classe Principal no package view com operações usando JOptionPane que permita ao usuário inserir números e consultar as chamadas perdidas. A aplicação deve rodar até uma opção de saída ser selecionada pelo usuário

3) Considerando a biblioteca FilaObject, já criada, faça:

Criar um projeto Java (EscalonamentoProcessos) e importe a biblioteca FilaObject. Esse novo projeto irá receber Objectos com dados de processos com o formato do objecto abaixo e simular o escalonamento de processos RoundRobin.

A classe EscalonadorController no package controller deve ter 1 método:

- O método escalonador que recebe uma Fila de processos, como parâmetros, verifica se o número que vezes é maior que 1, decrementa esse valor no objeto e, se for maior que 1, insere o objeto novamente na fila;

A classe Principal, no package view, no seu método Main, deve inicializar uma fila Com processos, como no exemplo abaixo

Objeto:

Processo
+Nome : String
+QtdRetornos : Int

Exemplo de entrada:

```
{"notepad.exe;14","write.exe;35","chrome.exe;27","acrobat.exe;52","firefox.exe;18","word.exe;25"}
```

4) Criar um projeto Java que implementa com a biblioteca FilaStrings para simular um ambiente corporativo, comum, como agências bancárias que tem diversos computadores e 1 impressora central. Nesses casos, a impressora tem uma fila de impressões para que cada documento enviado comece e termine a impressão sem que documentos se misturem.

A aplicação deve ter uma classe no package controller, ImpressoraController que tem os seguintes métodos:

- `insereDocumento(Fila f, String documento)`: void, o documento no formato ID_PC;Nome_Arquivo (Já validado antes do envio) deverá enfileirar os documentos enviados
- `imprime(Fila f)`:void, que desenfilera um documento da fila, por vez, exibe no console [#PC: ID_PC – Arquivo: Nome_Arquivo]. Cada impressão dura de 1 a 2 segundos usar `Math.random()` ou a classe `Random` e um `Thread.sleep*(tempo)` para simular o tempo de impressão. Exibir uma exceção caso não haja documento na fila de impressão.

Deve ter também uma classe Principal no package view com operações usando `JOptionPane` que permita ao usuário inserir e validar os documentos de entrada e iniciar um procedimento de impressão. A aplicação deve rodar até uma opção de saída ser selecionada pelo usuário

*** Como a classe de ImpressoraController não é uma Thread, para forçar um sleep, deve-se usar uma chamada estática da classe Thread (`Thread.sleep(tempo)`)**