

Acadêmico(a)			RA:
Curso	Tecnologia em Sistemas para Internet	Período:	
Disciplina	TSI32B - Estrutura, Pesquisa e Ordenação de Dados		
Professor	Eduardo Pena		
Exercícios sobre Listas, Filas e Pilhas			
Orientações gerais:			
1 - Resolva cada exercício em sua respectiva classe, cujo nome deve estar no formato EX"NumExercício"_"TítuloExercício".java.			
2 - A interpretação das questões é parte do processo de avaliação.			
3 - Enviar todos os arquivos (apenas .java) em um arquivo compactado .zip. Verificar prazos no Moodle Institucional.			

- Escreva um programa Java para criar um ArrayList de Strings, adicione algumas strings, percorra e imprima o ArrayList sequencialmente.
 - Refaça o exercício acima para um ArrayList de Objetos Aluno(ra,nome,nascimento,coeficiente).
- Escreva um programa Java para inserir um elemento em uma ArrayList de inteiros na: primeira posição do Arraylist; última posição do Arraylist; posição específica (dada por um índice qualquer) do Arraylist—verifique se o índice é válido.
- Dado um ArrayList de Objetos Aluno(ra,nome,nascimento,coeficiente), escreva um programa Java para atualizar o nome do aluno(a) em uma posição específica do Arraylist (dado por um índice).
- Escreva um programa Java para pesquisar um elemento em um ArrayList.
 - Reescreva o programa, mas sem usar o método definido no JDK, isto é, utilize um laço para percorrer e encontrar o elemento no ArrayList.
- Escreva um método removeDuplicados que recebe como parâmetro uma ArrayList de Strings e elimina quaisquer duplicatas do ArrayList. Por exemplo:


```
"eu", "tu", "tu", "ela", "ele", "ele", "eu"
```

 se torna


```
"eu", "tu", "ela", "ele"
```

 depois da chamada removeDuplicados(lista).
- Escreva um método filtraIntervalos que recebe um ArrayList de inteiros e dois valores inteiros min e max como parâmetros e remove todos os elementos cujos valores estão no intervalo min a max(inclusive) da lista.
- Escreva um método minimoNaFrente que recebe um ArrayList de inteiros como parâmetro e que move o(s) valor(es) mínimo(s) na lista para a frente—preservando a ordem dos demais elementos. Por exemplo:


```
{3, 8, 2, 4, 2, 14, 9}
```

 se torna


```
{2, 2, 3, 8, 4, 14, 9}
```

 depois de uma chamada minimoNaFrente(lista).

8. Escreva um método chamado `espelha` que receba um `ArrayList` de `Strings` como parâmetro e produza uma cópia espelhada da lista como saída. Dessa forma, os valores originais são seguidos por esses mesmos valores na ordem oposta. Por exemplo:

```
["carro", "de", "caminhao"]
```

se torna

```
["carro", "de", "caminhao", "caminhao", "de", "carro"]
```

depois de uma chamada `espelha(lista)`.

9. Escreva um programa Java para armazenar o conteúdo do arquivo de texto em estruturas `ArrayList`. Cada token (palavra, número, espaço) deve ser uma `String`. Cada linha do arquivo é um `ArrayList` de `Strings`. O arquivo é representado por um `ArrayList` com os `ArrayList` de `String` de cada linha.

- (a) Escreva um programa para ler um arquivo `.txt` e popular as estruturas `ArrayList`.
- (b) Implemente um método `imprime`, que imprime o conteúdo dos `ArrayLists` em tela, de modo que a apresentação fique como a contida no arquivo texto.
- (c) Escreva um método, `tornaMaiusculo`, que transforma todo o conteúdo das `ArrayLists` para palavras com letras maiúsculas.
- (d) Escreva um método `Busca`, que recebe uma `String` como parâmetro, e percorre as estruturas `ArrayLists` para retornar um conjunto com os índices que se encontram a palavra de entrada (não importando o case maiúsculo/minúsculo). Por exemplo:

Texto é um conjunto de palavras
Isto é um texto
Procure no Texto não

e `busca("texto")` retorna $\{1,1\}$ (para linha 1, token 1), $\{2,7\}$ (para linha 2, token 7), e $\{3,5\}$ (para linha 3, token 5).

10. Escreva um programa para gerar um `ArrayList` com N elementos gerados de forma aleatória.
- (a) `geraInteiro(n, min, max)` retorna um `ArrayList` com n inteiros (distribuídos uniformemente)—cada inteiro está entre `min` e `max`.
 - (b) `geraDouble(n, min, max)` retorna um `ArrayList` com n doubles (distribuídos uniformemente)—cada double está entre `min` e `max`.
 - (c) `geraString(n, t)` retorna um `ArrayList` com n `Strings` de comprimento t .
11. Escreva um programa que armazene em um `ArrayList` N inteiros (distribuídos uniformemente). Cada inteiro deve estar entre 0 e um valor armazenado em uma variável *limite*. Meça o tempo para geração do `ArrayList`.

Além disso, escreva um método para busca de números aleatórios dentro do `ArrayList` gerado. Execute o método k vezes. Meça o tempo mínimo, máximo e médio de cada busca; e o tempo total das k buscas. Apresente as leituras de performance em milissegundos. Finalmente, analise como os parâmetros N , *limite* e k afetam os tempos medidos.