

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
ESCOLA AGRÍCOLA DE JUNDIAÍ
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
COMPONENTE: TAD0009 – PROG. ORIENTADA A OBJETOS
PROF. JOSENALDE OLIVEIRA

Exercícios Propostos – Fixação ArrayList 2

1. Crie uma classe `TestArrayListNumber` com o método `main()` e escreva os códigos que atendam aos requisitos abaixo:
 - a. Crie um `ArrayList` de `Integer`, com a notação `List<Integer> list = new ArrayList<Integer>();`
 - b. Adicionar a este `ArrayList` 10 números inteiros aleatórios no intervalo `[10, 20]`. Verifique como fazer isto com o método `random()` da classe `Math` (`Math.random()`)
 - c. Verifique se há números repetidos na lista. Se houver informe-os e quantas vezes aparecem na lista. Exemplo: `list = {10,15,15,5,10,10,10,3,8,12}`. Saída: `{10: 4, 15:2}`
 - i. Dica: explore o método `contains` e o método `indexOf`
 - d. Crie uma cópia desta lista com o método `clone()`
 - e. Ordene em ordem decrescente esta lista clonada
 - f. Remova o maior número da lista original
 - g. Remova o maior número da lista clonada
 - h. Una a duas listas, colocando a lista clonada no fim da lista original
 - i. Verifique o tamanho da lista resultante
 - j. Retorne a soma dos elementos da lista resultante. Pesquise na web se há outras formas de fazer esta soma para além do método clássico
 - k. Substitua o menor (ou menores) números da lista resultante pelo valor médio da lista
 - l. Crie um vetor clássico Java preenchido com 5 valores inteiros e converta para lista. Pesquisa o comando `asList()`...
 - m. Pesquise sobre a collection `SET` em Java. Ela permite ordenação e valores duplicados? Inclua um exemplo didático com `Set` para mostrar sua resposta.
2. Crie uma classe `GeraEmailsInstitucionais` com o método `main()`
 - a. Leia (ou obtenha) o nome completo de 20 docentes da UFRN (aleatoriamente). Dica: usar o dataset (fonte de dados) abertos de docentes UFRN disponível em arquivo .CSV em: <https://dados.ufrn.br/dataset/docentes>, no recurso <https://dados.ufrn.br/dataset/docentes/resource/6a8e5461-e748-45c6-aac6-432188d88dde> (docentes.csv)
 - b. Gerar, para cada docente, seu e-mail institucional, totalmente em minúsculo, da seguinte forma: primeironome.ultimonome@ufrn.br. Exemplo: Josenalde Barbosa de Oliveira, o e-mail gerado será josenalde.oliveira@ufrn.br. Sugestão usar o método `split()` das `Strings`. Caso haja e-mails gerados duplicados (pessoas que diferem nos nomes do meio), tratar estes casos, gerando o e-mail completo, exemplo: João Paulo Silva e João Pedro Silva: `joao.paulo.silva` e `joao.pedro.silva`
3. Crie uma classe que receba um nome completo na forma de uma *String* a partir do console e mostre a abreviatura deste nome. Não se devem abreviar as palavras com 2 ou menos letras. A abreviatura deve vir separada por pontos. Ex: Paulo Jose de Almeida Prado. Abreviatura: P. J. de A. P. Lembre do parâmetro `String args[]` do método `main()`. Você deve testar com o código executado no terminal. Exemplo: se seu código é `Abrevia.java`, será: `java Abrevia "Paulo da Silva do Amaral"`

4. Seja a distância entre cidades dada pela tabela abaixo:

	1	2	3	4	5
1	00	15	30	05	12
2	15	00	10	17	28
3	30	10	00	03	11
4	05	17	03	00	80
5	12	28	11	80	00

Implemente um programa que:

- leia a tabela acima em um *array* bidimensional. O programa não deve perguntar distâncias já informadas (por exemplo, se o usuário já forneceu a distância entre 1 e 3 não é necessário informar a distância entre 3 e 1, que é a mesma) e também não deve perguntar a distância de uma cidade para ela mesma, que é 0.
- leia um percurso fornecido pelo usuário em um *array* unidimensional. Calcule e mostre a distância percorrida. Por exemplo: dado o percurso 1, 2, 3, 2, 5, 1, 4, para a tabela mostrada como exemplo teremos: $15 + 10 + 10 + 28 + 12 + 5 = 80$ km.