

## Lista de Exercícios

### SET

- 1) (0,75) Crie uma classe `Aluno` com os atributos `matricula` (`String`) e `nome` (`String`). Sobrescreva os métodos `equals()` e `hashCode()` para considerar dois alunos iguais se possuírem a mesma matrícula.

Em seguida, crie um programa em Java (`Main`) que contenha um `HashSet<Aluno>` e adicione vários alunos, incluindo alguns com a mesma matrícula. Imprima a Lista de Alunos, não deve haver alunos com matrículas iguais.

Pare a entrada de dados quando a matrícula for igual a 0

(Obrigatório o uso de `HashSet`)

Entrada	Saída
10 Bruno 20 Jose 10 Pedro 0	Alunos: Bruno José

Código sobrescrito para o `equals` e `hashCode`:

```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    return matricula.equals(((Aluno) o).getMatricula());
}

@Override
public int hashCode() {
    return Objects.hash(matricula);
}
```

- 2) (0,75) Escreva um programa(método `main`) que crie dois `HashSet<Integer>` com valores aleatórios. O programa deverá criar um novo `HashSet` que contenha a interseção entre os dois conjuntos (valores presentes em ambos), sem usar o método `retainAll()`. Compare os elementos manualmente, retorne um novo `HashSet` com a interseção e imprima os números presentes no novo `HashSet`. Na entrada dos números para um SET, pare quando digitar um número negativo

Entrada	Saída
1 2 3 4 5 6 -1 4 5 6 7 8 9 -1	4 5 6
1 1 1 1 2 2 2 3 3 3 -1 2 2 2 4 4 4 -1	2

Lembrete: para comparar dois textos utilize o método equals ou equalsIgnoreCase da classe String.

- 3) (0,75) Crie um programa em Java que preencha uma List<Integer> com elementos duplicados, converta-a para um HashSet e depois de volta para uma List, removendo as duplicatas. Observe que não será mantido a ordem de entrada, pois o HashSet não mantém ordem. Na entrada dos números pare quando digitar um número negativo

Entrada	Saída
4 1 2 3 4 -1	1 2 3 4

- 4) (0,75) Crie um programa em Java que Verifique a existência de uma determinada string dentro de um conjunto:
- Crie um HashSet para armazenar um conjunto de strings. Entre diversas palavras até digitar "fim".
  - Verifique se uma determinada string existe no conjunto.
  - Utilize o método `contains()` para essa verificação.

Entrada	Saída
José José Pedro Manoel Pedro fim Ana	Não existe

- 5) (1,5) Crie um programa em Java que gerencia a lista de convidados de um evento. O sistema deve permitir adicionar convidados, remover convidados e verificar se uma pessoa está na lista de convidados. Cada convidado terá apenas seu nome (String) armazenado. Utilize HashSet para implementar a lista de convidados. Pode

trabalhar com a classe Contato ou utilizar apenas String. O sistema deve ter as seguintes funcionalidades disponibilizadas em um Menu:

- Adicionar um convidado.
- Remover um convidado.
- Verifique se um convidado está na lista.
- Exibir todos os convidados.
- Imprimir a quantidade de convidados cadastrados.
- Apagar todos os convidados.

6) (1,00) Escreva um programa em Java que recebe uma lista de números inteiros do usuário e então armazena esses números e a quantidade de ocorrências em um HashMAP<Integer, Integer> para armazenar o número como chave e a contagem de repetições como valor.

- Crie um HashMAP para armazenar a dupla<Número, Ocorrências>.
- Entre com diversos números **até digitar um valor negativo**.
- Verifique se o número existe como chave e aumente o valor correspondente a chave(número). Caso não exista, crie um novo elemento no Mapa.
- Ao fim imprima o conteúdo do mapa conforme exemplo abaixo.

Entrada	Saída
1 2 3 2 1 3 4 5 4 4 -1	1: 2 2: 2 3: 2 4: 3 5: 1

7) (1,0) Faça um programa em Java que preenche um HashMap<String, Integer> e então inverte as chaves e os valores. Vamos assumir que o usuário não irá digitar valores repetidos(podem validar, mas não é obrigatório). Use um novo HashMap<Integer, String> para armazenar o resultado.

- Crie um HashMap para armazenar a dupla<String, Integer>.
- Entre com diversas duplas de valores String e Integer e adicionem no HashMap. Até digitar a palavra Fim
- Crie um novo HashMap para armazenar a dupla<Integer, String>
- Copie os elementos do primeiro HashMap invertendo os valores.
- Ao fim imprima o conteúdo do mapa conforme exemplo abaixo.

Entrada	Saída
A 1 B 2 C 3 Fim	{1=A, 2=B, 3=C}

- 8) (1,5) Crie uma classe Produto com os atributos id (String) e nome (String). Crie um programa(Main) que utilize um HashMap<String, Produto> para armazenar produtos, onde a chave é o id. Na main crie um Menu que disponibilize as seguinte opções:
- a) Adicionar um produto.
  - b) Buscar um produto pelo id.
  - c) Remover um produto pelo id.
  - d) Listar todos os produtos.
- 9) (1,0) Escreva um programa em Java que conte a frequência de cada letra em uma string, ignorando espaços e diferenciando letras maiúsculas e minúsculas. Use um HashMap<Character, Integer> para armazenar as contagens. (Similar ao exemplo de contar ocorrência de números).

Entrada	Saída
Hello World	{H=1, e=1, l=3, o=2, W=1, r=1, d=1}

- 10) (1,5) Crie uma classe Aluno com os atributos matricula (String), nome (String) e cpf (String). Crie um programa em Java (classe Main) que utilize um HashMap<String, Aluno> para armazenar alunos, onde a chave é a matrícula. Em seguida, crie um HashMap<String, Double> para armazenar as notas dos alunos, onde a chave é a matrícula. Na main crie um Menu que disponibilize as seguinte opções:
- a) Adicionar um aluno.
  - b) Adicionar uma nota a um aluno.
  - c) Calcular a média das notas de todos os alunos.
  - d) Imprimir os dados dos alunso