Disciplina: Programação Orientada a Objetos I
Professor: Maicon Rafael Zatelli

Coleções em Java: Set e Map

Leitura

Até o momento já vimos algumas coleções em Java, dentre elas vetores, matrizes, `List` (`ArrayList` e `LinkedList`) e `String` (que é uma coleção de caracteres). Assim, o objetivo dessa atividade é que vocês também explorem coleções como `Set` e `Map`.

Set

Para entender o funcionamento de um `Set`, leia os links abaixo. Em seguida, tente implementar cada um dos exemplos apresentados por eles.

- `https://www.javatpoint.com/java-hashset`
- `https://www.tutorialspoint.com/java/java_hashset_class.htm`

Map

Para entender o funcionamento de um `Map`, leia os links abaixo. Em seguida, tente implementar cada um dos exemplos apresentados por eles.

- `https://www.javatpoint.com/java-hashmap`
- `https://www.tutorialspoint.com/java/java_hashmap_class.htm`

Mais sobre coleções em Java

Para saber mais sobre coleções em Java acesse os links abaixo:

- `https://www.javatpoint.com/collections-in-java`
- `https://www.tutorialspoint.com/java/java_collections.htm`

Ordenação em Coleções

Outra funcionalidade importante em uma coleção é a ordenação, na qual os elementos membros da coleção podem ser comparados e ordenados de acordo com algum critério. Para entender o funcionamento da ordenação, leia o link abaixo e implemente o exemplo apresentado.

- `https://www.javatpoint.com/Sorting-in-collection-framework`

Atividade

1. Implemente um método `equals(Object o):boolean` na classe `Book`, disponibilizada no exemplo em `https://www.javatpoint.com/java-hashset`. Marque todos os atributos como privados. Este método receberá um objeto, que deve ser um livro e então deve retornar `true` se o identificador do livro for o mesmo daquele do objeto passado como parâmetro. No método `main`, tente criar um novo livro com o mesmo identificador de um já existente. Adicione ele no set e veja o que acontece ao imprimir todos os livros do set.

2. Implemente um método `equals(Object o):boolean` na classe `Book`, disponibilizada no exemplo em `https://www.javatpoint.com/java-hashmap`. Marque todos os atributos como privados. Este método receberá um objeto, que deve ser um livro e então deve retornar `true` se o identificador do livro for o mesmo daquele do objeto passado como parâmetro.

   A: Ao invés de usar a chave 1, 2, 3 no map, utilize os próprios identificadores dos livros. Para isso, crie um método `getId()` na classe `Book`.

   B: No método `main`, tente criar um novo livro com o mesmo identificador de um já existente. Adicione ele no map e veja o que acontece ao imprimir todos os livros do map.

3. Utilize um map para armazenar os votos de candidatos em uma eleição. Suponha que os números dos cadidatos sejam 1111, 2222, 3333, 4444, 5555. Leia um valor n do teclado e em seguida leia n valores inteiros, que correspodem aos votos efetuados. Ao final, imprima a quantidade de votos para cada candidato válido e o código do vencedor da eleição. Código de candidatos inválidos devem ser armazenados como código 0.

4. Faça o mesmo que a questão anterior, mas agora suponha que ao invés de informar um código numérico, é informado o nome do candidato, por exemplo "joao", "jose", "maria".

5. Utilize um set para armazenar números informados por um usuário. Leia um valor n do teclado e em seguida leia n valores inteiros (que podem ser números repetidos ou não). Ao final, imprima a quantidade de números diferentes informados pelo usuário.

6. Faça o mesmo que a questão anterior, mas agora suponha que ao invés de informar um número, é informado uma palavra, como "areia", "fruta", "ufsc".

Questões do URI

Para esta segunda parte da atividade, iremos resolver algumas questões de algoritmos do portal do URI. Caso você nunca utilizou o portal do URI, então registre-se primeiro no site:

`https://www.urionlinejudge.com.br/judge/en/register`

Depois, registre um time no link abaixo (este time pode ser composto por apenas você):

`https://www.urionlinejudge.com.br/judge/en/teams`

A seguir, acesso o link do torneio e registre-se nele para participar e resolver as questões:

`https://www.urionlinejudge.com.br/judge/en/tournaments/rank/683`

Para se registrar no torneio, clique no botão "register"e ao pedir a palavra chave, digite **ufsc2018**

Resolvendo Questões no URI

Algumas dicas para resolver as questões do URI.

- A entrada é informada como se fosse o usuário digitando os valores.
- A saída deve ser exatamente dada como especificado, ou seja, se o problema diz que a letra precisa ser maiúscula na saída, então ela deve ser maiúscula. Se diz que precisa ter espaço, então coloque espaço. Se diz para pular de linha, então pule linha. Se pede com arredondamento em 2 casas decimais, então faça isso.
- O código abaixo é uma ilustração de como resolver as questões no URI. Neste exemplo, soluciono o problema `https://www.urionlinejudge.com.br/judge/en/problems/view/1001`. Tente entender o código e submeter a solução abaixo para aprender como funciona.

```
public class Main {
    public static void main(String args[]) {
        java.util.Scanner scan = new java.util.Scanner(System.in); //Permitirá ler dados da tela

        int a, b;

        a = scan.nextInt(); //Leio o primeiro inteiro
        b = scan.nextInt(); //Leio o segundo inteiro

        int soma = a + b; //Somo os dois números

        System.out.println("X = " + soma); //Imprimo X = SOMA e pulo linha
                                           //(por isso o println e não o print)
    }
}
```

Uma estrutura de exemplo de solução é como abaixo:

```
//Declara seus imports aqui, se precisar.
public class Main {
    //Declare variáveis, atributos para usar em sua solução, caso necessário
    public static void main(String args[]) {
        java.util.Scanner scan = new java.util.Scanner(System.in); //Permitirá ler dados da tela

        //Escreva sua solução aqui
    }
}
```

Quando o enunciado do problema diz que a entrada termina com EOF, significa que o usuário não irá mais informar nada. Assim, você pode utilizar a seguinte estrutura para resolver o problema:

```
public class Main {
    public static void main(String[] args) throws IOException {
        java.util.Scanner scan = new java.util.Scanner(System.in);

        while (scan.hasNext()) { //Repete este loop enquanto tiver algo para ler

            //Escreva aqui sua lógica de solução

        }
    }
}
```

Caso precisar, pesquise mais sobre o método `hasNext()` da classe `Scanner`.

Quando o enunciado do problema diz que a entrada terminará com N = 0, significa que o "usuário" irá informar zero quando você ler algo da tela. Assim, isso é como fizemos nos demais exercícios, por exemplo, quando o código do aluno for 0, deve-se parar de ler os alunos da tela e sair do loop.