

UNIVERSIDADE FEDERAL DO PIAUÍ CENTRO DE CIÊNCIAS DA NATUREZA DEPARTAMENTO DE COMPUTAÇÃO GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Grupo 08**

**KAUÃ MACHADO DA SILVA**

**JEOVANDO DA SILVA REIS JUNIOR**

**WENCYO RAFAEL LIMA DE SOUSA**

**Relatório Técnico:   
Atividade de Participação 07**

**Java Collections API**

**Teresina-PI 2023**

**Relatório Técnico: Resolução das questões sobre a Java Collections API**

1. **Introdução**.

Segue o relatório detalhado da Atividade de Participação 07 dos alunos Kauã Machado Da Silva, Jeovando Da Silva Reis Junior e Wencyo Rafael Lima De Sousa, realizada na disciplina de Estrutura de Dados do Departamento de Computação - CCN da Universidade Federal do Piauí (UFPI), sob a orientação do professor Raimundo Moura

* 1. **Descrição do Problema**

O problema consiste em explicar as classes do Java Collectins API, bem como a implantação da tabela de símbolos, como discriminadas nas questões, assim como a consulta e exclusão e os dados estatísticos das funcionalidades.

**1.2. Java Collections API**

Java Collections API é uma parte da plataforma Java que fornece um conjunto de interfaces e classes para representar e manipular grupos de dados como uma única unidade, chamada coleção ou collection. As coleções podem ser usadas para armazenar, recuperar, modificar e organizar diferentes tipos de dados, como listas, conjuntos, mapas, pilhas, filas.

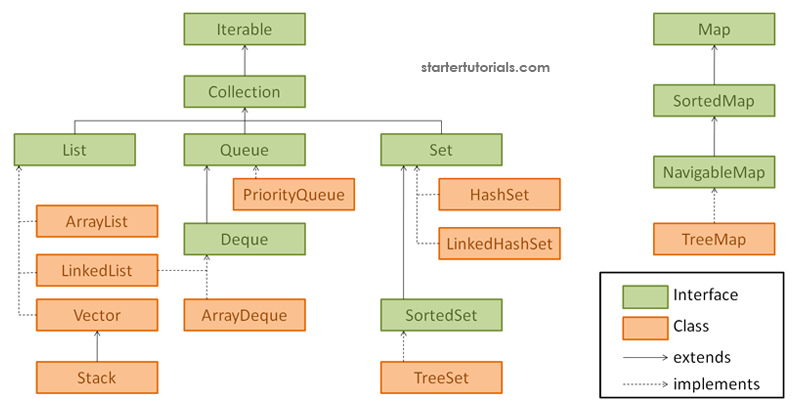
A Coleção é compostas pelos seguimentos elementos: interface, implementações, algoritmos. Sendo que são as classes concretas que fornecem a funcionalidade das interfaces e os algoritmos realizam operações úteis sobre as coleções.

Imagem 1: Java Collections Framework hierarchy. <https://www.startertutorials.com/corejava/introduction-java-collections-framework.html>

1. **Questões**
   1. **Classes Vector, LinkedList e ArrayList.**

As classes Vector, LinkedList e ArrayList são três implementações da interface List em Java, que representa uma coleção de elementos que podem ser acessados por um índice.

Vector e ArrayList são muito parecidos. O vector é anterior a inclusão da API Collection, eles utilizam um array interno para armazenar os elementos, e aumentam o tamanho desse array conforme a necessidade. A principal diferença entre eles é que o Vector é sincronizado, assim, considerada obsoleta e menos eficiente do que a classe ArrayList, que não é sincronizada por padrão.

A principal vantagem do vector consiste dele ser sincronizado, pode ser usado como uma pilha, tem métodos para obter a capacidade e o fator de incremento do array interno. Quanta a principal desvantagem é mais lento do que o ArrayList, tem métodos obsoletos que podem causar confusão, não é recomendado para uso em novas aplicações.

Tendo o ArrayList como principais vantagens: a eficiência para operações de acesso aleatório, pode ser facilmente acessado por qualquer método que implemente a interface List, pode ser facilmente convertido para um array. Por outro lado, as desvantagens, não é tão eficiente quanto LinkedList para operações de inserção e remoção, pode usar mais espaço do que LinkedList.

LinkedList é uma estrutura de dados que mantém uma lista de objetos. Os objetos são armazenados em uma lista ligada, que é uma lista de nós, onde cada nó contém um objeto e um ponteiro para o próximo nó e para o nó anterior.

As principais vantagens, é eficiente para operações de inserção e remoção, não usa tanto espaço quanto ArrayList ou Vector. Enquanto as desvantagem, não é sincronizado, o que significa que não é seguro para uso em threads concorrentes, não pode ser facilmente acessado por qualquer método que implemente a interface List,não pode ser facilmente convertido para um array.

* 1. **Classe HashSet, LinkedHashSet e TreeSet**

HashSet, LinkedHashSet e TreeSet são conjuntos em Java. Todos eles implementam a interface Set. A diferença entre eles é a forma como os elementos são armazenados e recuperados.

O HashSet usa uma tabela de hash para armazenar seus elementos. Isso significa que ele não mantém a ordem dos elementos e não permite elementos duplicados. É rápido para operações de inserção, remoção e busca; não guarda a ordem de inserção dos elementos e não permite elementos duplicados. Entretanto, pode usar mais memória do que outros tipos de conjuntos e pode ser difícil pesquisar elementos em um HashSet se você não souber o valor exato do elemento.

O LinkedHashSet é semelhante ao HashSet, mas mantém a ordem dos elementos em que foram adicionados. Ele usa uma lista vinculada para manter a ordem dos elementos. Porém, diferentemente dos outros conjuntos pode ser difícil pesquisar elementos em um LinkedHashSet se você não souber a ordem exata em que os elementos foram inseridos.

O TreeSet armazena seus elementos em uma árvore binária. Isso significa que ele mantém os elementos em ordem classificada e pode ser usado para classificar conjuntos. No entanto, ele pode ser mais lento do que outras implementações.

* 1. **Classe HashMap, LinkedHashMap e TreeMap**

Os três tipos de mapas em Java são HashMap, LinkedHashMap e TreeMap. Todos eles implementam a interface Map e têm métodos semelhantes.

HashMaps usam uma tabela hash para armazenar pares de chave-valor. A chave é usada para calcular o índice na tabela hash, onde o valor é armazenado. HashMaps são rápidos para operações de inserção, remoção e busca. Eles também permitem elementos duplicados. Ademais, HashMaps não mantêm a ordem de inserção dos elementos. Eles também podem usar mais memória do que outros tipos de mapas.

LinkedHashMaps são semelhantes a HashMaps, mas eles mantêm uma lista ligada das chaves. Isso permite que você itere sobre os elementos em ordem de inserção. Pode ser destacado que o LinkedHashMaps são rápidos para operações de inserção, remoção e busca. Eles também permitem elementos duplicados. Eles também mantêm a ordem de inserção dos elementos. Todavia, LinkedHashMaps podem usar mais memória do que HashMaps.

TreeMaps são implementados como árvores binárias balanceadas. Isso permite que você itere sobre os elementos em ordem crescente ou decrescente. O TreeMaps são rápidos para operações de busca também não permitem elementos duplicados e sempre mantêm os elementos em ordem. Em contraponto podem ser mais lentos do que outros tipos de mapas para operações de inserção e remoção de elementos em sequência. Eles também podem usar mais memória do que outros tipos de mapas.

1. **Conclusão**

Esta aplicação fornece uma interface para interagir com uma Árvore Binária de Busca, permitindo a inserção de nomes, exibição gráfica da árvore, visualização de informações e realização de travessias. É uma ótima maneira de visualizar e entender o funcionamento de uma árvore binária de busca por meio de uma interface gráfica simples.