

(Exercício 1) Pesquise qual a principal diferença entre as implementações da interface Set: HashSet, LinkedHashSet e TreeSet.

HashSet só se aplica a objetos que possuem função hash.

LinkedHashSet é quase igual ao Hash set mas, os nós possuem uma ligação a mais.

TreeSet se aplica somente a elementos ordenados.

(Exercício 2) Pesquise qual a principal diferença entre as implementações da interface List: ArrayList, LinkedList, Vector e Stack.

ArrayList, é implementado como um array de alocação dinâmica, sempre que preenchido, seu tamanho aumenta metade do tamanho atual, ou seja, se possui 10 espaços cheios, passa a ter 15 espaços, no qual 5 estão vazios, também permite que os elementos sejam acessados por métodos get e set e adicionados por add e remove.

LinkedList, quase idêntico ao ArrayList, suas principais diferenças é que os métodos add e remove são mais eficientes do que o ArrayList, porém os métodos get e set são piores.

Vector, é semelhante a um ArrayList mas é sincronizado e possui alocação dinâmica diferente, em vez de somar metade do espaço, ele aumenta o dobro, ou seja, se possuem 20 espaços cheios, ela aumentara para 40, com 20 vazios.

Stack é mais limitado que o vetor, no qual só se consegue adicionar um elemento acima do último ou remover o último elemento.

(Exercício 3) Pesquise qual a principal diferença entre as implementações da interface Map: HashMap, TreeMap e Hashtable.

HashMap, Pode ter chave nula ou vários valores nulos e não possui ordem.

TreeMap, Não pode ter chave nula mas pode ter vários valores nulos e possui ordem crescente.

HashTable, Pode não ter nenhuma chave ou valor nulo, é sincronizado.

(Exercício 4) A classe Properties se encontra na hierarquia da interface Map. Usando a leitura complementar disponível nos slides (especificamente o website do Caelum), explique qual o seu uso e demonstre sua utilização em uma classe Main.

É utilizado para configurar aplicações, mapear String, possui métodos de gravação e leitura do mapeamento que se baseia em um arquivo de texto, por conta de facilitar sua persistência.

```

public class ex05 {
    public static void main(String[] args) {

        Properties prop = new Properties();
        Set props;
        String str;

        //Definir o valor das propriedades
        prop.setProperty("DataBase ", "db1");
        prop.setProperty("User ", "nome");
        prop.setProperty("Password ", "senha");

        //Definir chave
        props = prop.keySet();

        //Mostrar propriedades
        Iterator iterator = props.iterator();
        while(iterator.hasNext()) {
            str = (String) iterator.next();
            System.out.println("Propriedade: " + str + "\nValor da propriedade: " + prop.getProperty(str));
        }
    }
}

```

#### :: Saída - prop (run)

```

run:
Propriedade: User
Valor da propriedade: nome
Propriedade: Password
Valor da propriedade: senha
Propriedade: DataBase
Valor da propriedade: db1
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

```

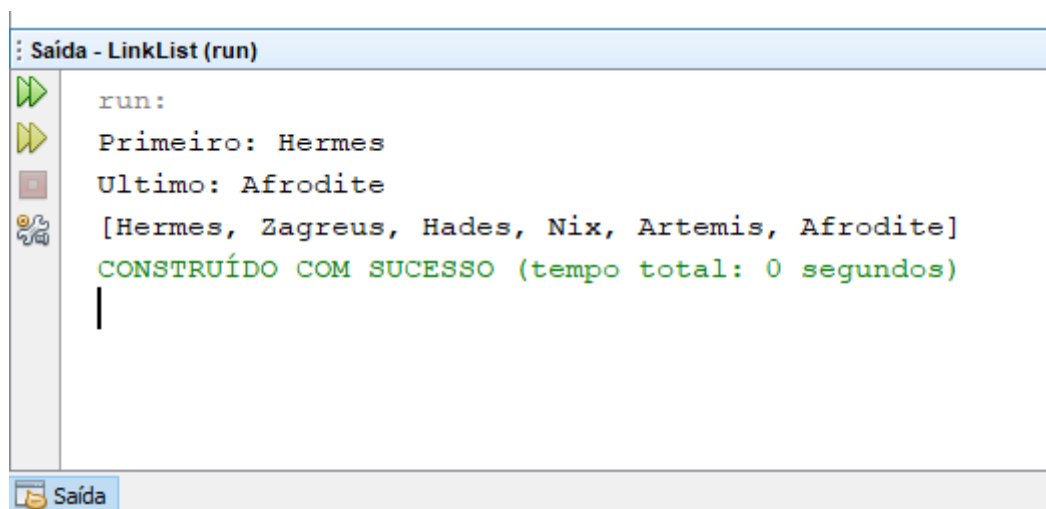
(Exercício 5) Qual a diferença entre a classe ArrayList e LinkedList?

LinkedList é mais rápido para inserir e retirar, e ArrayList é mais rápido em percorrer.

(Exercício 6) Para LinkedList, escreva uma classe que teste os métodos addFirst(), addLast(), getFirst(), getLast(), removeFirst() e removeLast().

```
public class ex06 {  
    public static void main(String[] args) {  
        LinkedList<String> hades = new LinkedList<String>();  
        hades.add("Zagreus");  
        hades.add("Hades");  
        hades.add("Nix");  
        hades.add("Artemis");  
  
        hades.addFirst("Hermes");  
        hades.addLast("Afrodite");  
  
        System.out.println("Primeiro: " + hades.getFirst());  
        System.out.println("Ultimo: " + hades.getLast());  
  
        //hades.removeFirst();  
        //hades.removeLast();  
  
        System.out.println(hades);  
    }  
}
```

Metodos addFirst() addLast() getFirst() getLast()



```
run:  
Primeiro: Hermes  
Ultimo: Afrodite  
[Hermes, Zagreus, Hades, Nix, Artemis, Afrodite]  
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)  
|
```

```

//
public class ex06 {
    public static void main(String[] args) {
        LinkedList<String> hades = new LinkedList<String>();
        hades.add("Zagreus");
        hades.add("Hades");
        hades.add("Nix");
        hades.add("Artemis");

        hades.addFirst("Hermes");
        hades.addLast("Afrodite");

        //System.out.println("Primeiro: " + hades.getFirst());
        //System.out.println("Ultimo: " + hades.getLast());

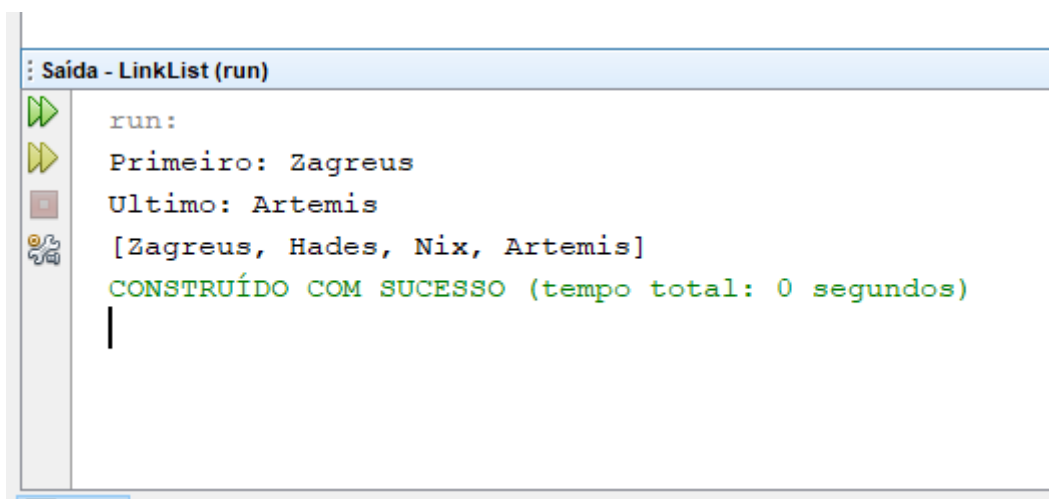
        hades.removeFirst();
        hades.removeLast();

        System.out.println("Primeiro: " + hades.getFirst());
        System.out.println("Ultimo: " + hades.getLast());

        System.out.println(hades);
    }
}

```

removeFirst(); removeLast();



```

Saída - LinkList (run)
run:
Primeiro: Zagreus
Ultimo: Artemis
[Zagreus, Hades, Nix, Artemis]
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

```

