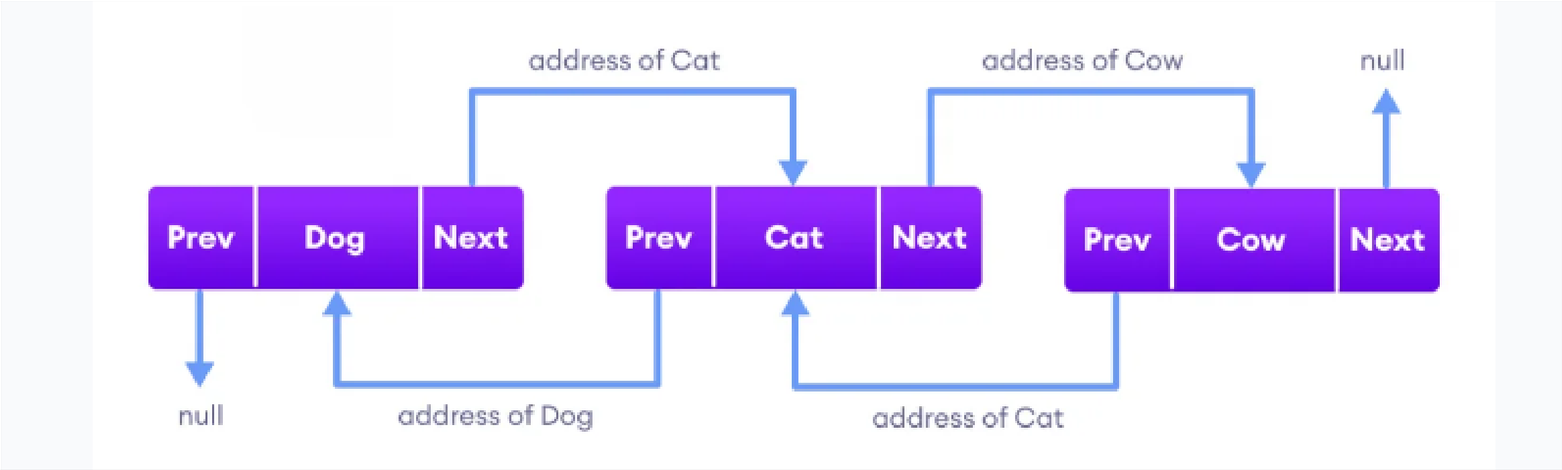
LinkedList

DESENVOLVIMENTO DE SOFTWARE

O QUE É ?



Listas encadeadas, cada elemento guarda a posição do próximo, ou seja, se eu quiser achar o quinto elemento, preciso do quarto; para achar o quarto, preciso do terceiro. E assim sucessivamente até percebermos que precisamos percorrer toda a lista se quisermos chegar num elemento qualquer.

Caracteristicas

1

Implementa a Interface List

Utiliza os métodos dísponiveis na interface List.

2

Redimencionamento

Não é necessário realocar memória para acomodar novos elementos.

3

Acesso Sequêncial

Acessa os índices de forma corrente. Percorre a lista do iníco ou do fim, não retorna índice específico.

4

Flexibilidade

Remover e adicionar índices rapidamente de outras estruturas, como pilhas e filas.

Métodos Próprios

addFirst();	**Adiciona item na 1° posição da lista;**
addLast();	**Adiciona item na última posição da lista;**
removeFirst();	**Remove o primeiro item da lista;**
removeLast();	**Remove o último item da lista;**
getFirst();	**Retorna o 1° item;**
getLest();	**Retorna o último item;**

```java
import java.util.LinkedList;
public class Exemplo {

    public static void main(String args[])
    {
        // Criar o objeto LinkedList
        LinkedList<String> ll = new LinkedList<String>();

        // Adicionando elementos na lista
        ll.add("A");
        ll.add("B");
        ll.addLast("C");
        ll.addFirst("D");
        ll.add(2, "E");

        //Realizando um for
        //size() para obter o tamanho de uma coleção

        for (int i = 0; i < ll.size(); i++) {
            System.out.println(ll.get(i));
        }

        System.out.println(ll);

        // Retornando um valor
        System.out.println(ll.get(3));

        // Removendo elementos na lista

        ll.remove("B");
        ll.remove(3);
        ll.removeFirst();
        ll.removeLast();

        System.out.println(ll);
    }
}
```

```
Output:
D
A
E
B
C

[D, A, E, B, C]

B

[A]
```

Lembre-se de que acessar elementos em uma LinkedList pelo índice pode ser ineficiente porque cada acesso implica uma travessia desde o começo da lista até o índice desejado.

Complexidade

Complexidade Constante

addFirst();
addLast();
removeFirst();
removeLast();
getFirst();
getLest();

Complexidade Linear

add(int index, E element);
get(int index);
remove(int index);

Alunos

1. **Felipe Ruzin**

2. **Ana Carolina Queiroz**

3. **Pedro Henrique Elias Calle**