

CMP1054 - java

LISTAS LIGADAS ORDENADAS LINEARES

Listas ligadas Lineares

- Uma das formas mais simples de interligar os elementos de um conjunto.
- Estrutura em que as operações inserir, retirar e localizar são definidas.
- Podem crescer ou diminuir de tamanho durante a execução de um programa, de acordo com a demanda.
- Itens podem ser acessados, inseridos ou retirados de uma lista.

Listas ligadas Lineares

- Duas listas podem ser concatenadas para formar uma lista única.
- Ou uma pode ser partida em duas ou mais listas.
- Adequadas quando não é possível prever a demanda por memória.
 - Permitindo a manipulação de quantidades imprevisíveis de dados, de formato também imprevisível.
- São úteis em aplicações tais como:
 - manipulação simbólica
 - gerência de memória
 - Simulação
 - compiladores.

TAD Listas Ligadas

Ordenadas Lineares

- O conjunto de dados e de operações a ser definido depende de cada aplicação.
- Um conjunto de operações necessário a uma maioria de aplicações é definida a seguir.
- Os dados e as operações sobre eles são implementadas através objetos ou estruturas auto-referenciados.

Operações

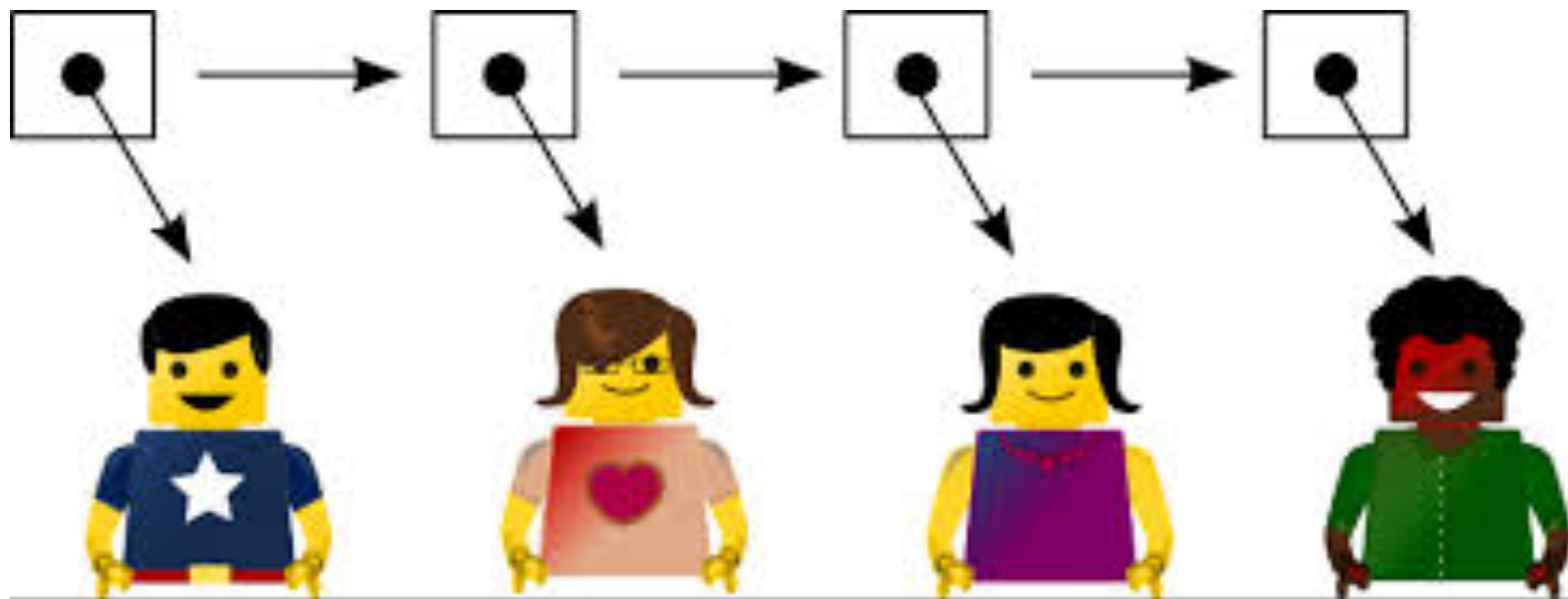
1. Criar uma lista vazia.
2. Adicionar um item na lista ordenada.
3. Remover um item da lista.
4. Pesquisar por um item na lista.
5. Modificar um item da lista.
6. Imprimir todos os itens da lista.

Listas Ligadas Lineares

- Cada elemento da lista contém a informação que é necessária para alcançar o próximo elemento.
- Permite utilizar posições não contíguas de memória.
- É possível inserir e retirar elementos sem necessidade de deslocar os itens seguintes da lista.

Listas Ligadas Lineares

- A lista é constituída de nós (elementos ou células).
- Cada nó contém um item da lista e uma referência para o nó seguinte.
- A classe Lista contém uma referência para o primeiro nó da lista.



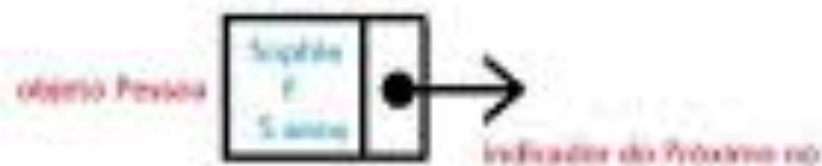
Classe Pessoa



Vetor
de Pessoa[]

Sophia	Samara	Marcos	Felipe	Eliab	Eloisa
F	F	M	M	F	F
5 anos	11 anos	10 anos	8 anos	7 anos	9 anos
0	1	2	3	4	5

Nó da Lista



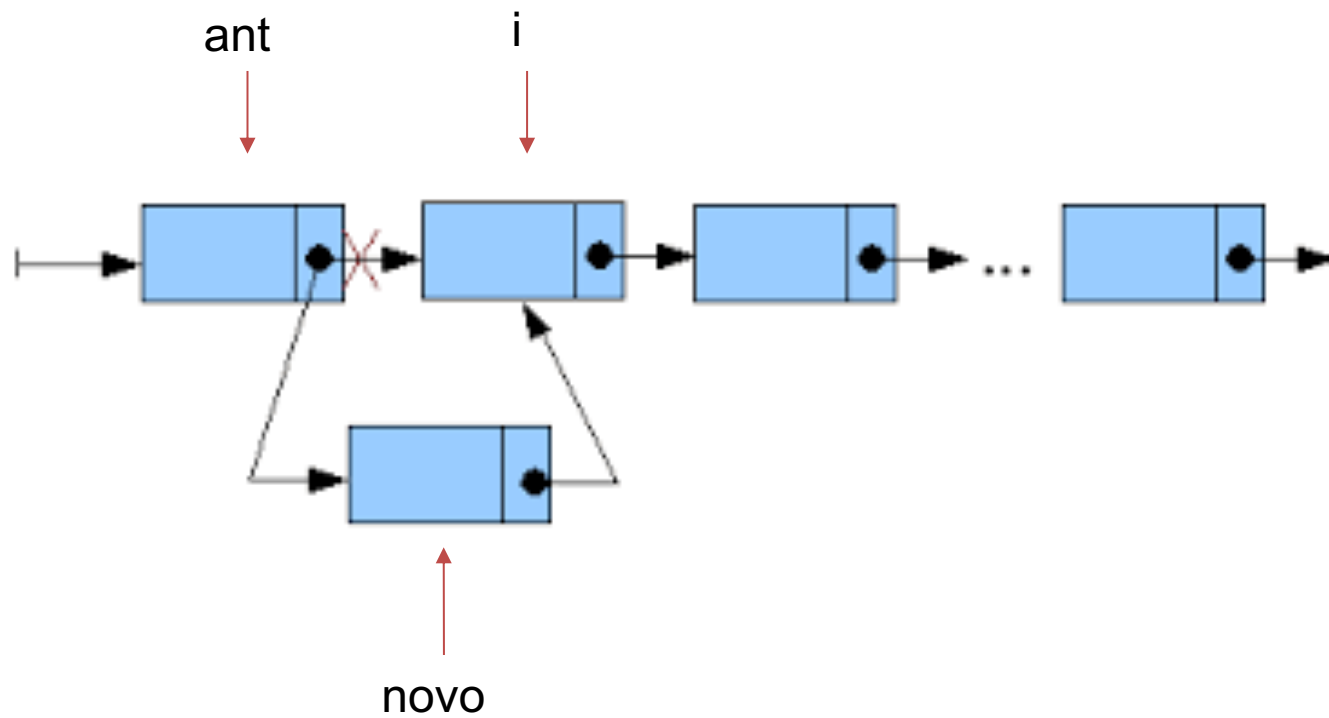
Lista Encadeada Simples



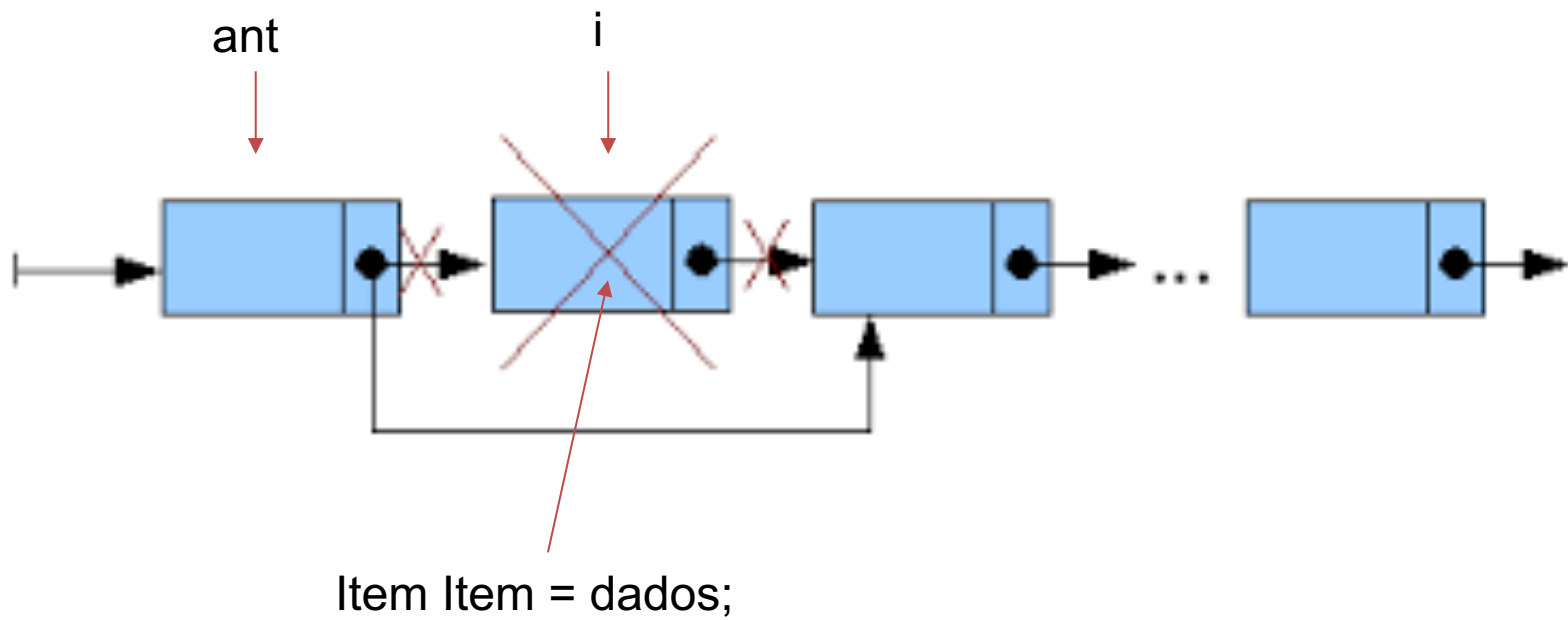
Implementação de uma lista ordenada

- A operação de inserção ou adição de um novo item será realizada em qualquer posição da lista que mantenha a ordem dos itens.
- A operações de pesquisa e de remoção.
 - Serão feitas por busca através de um atributo chave a ser escolhido previamente.

ADIÇÃO



REMOÇÃO



```
package listaligadainsercaoinicio;
```

```
public class Item {  
    private String descricao;  
    private int quantidade, prioridade;  
    public Item(String desc, int qt, int pri) {  
        descricao = desc;  
        quantidade = qt;  
        prioridade = pri;  
    }  
    public String getDescricao() { return descricao; }  
    public int getQuantidade() { return quantidade; }  
    public int getPrioridade() { return prioridade; }  
    @Override  
    public String toString(){  
        String aux = descricao + "\n";  
        aux = aux + Integer.toString(quantidade) + "\n";  
        aux = aux + Integer.toString(prioridade)+"\n";  
        return aux;  
    }  
}
```

```
package listaligadainsercaoinicio;
```

```
public class Lista {
```

```
    public class No{
```

```
        private Item dados;
```

```
        private No prox;
```

```
        public No(Item aux) {
```

```
            dados = aux;
```

```
            prox = null;
```

```
        }
```

```
    }
```

```
    private int tam;
```

```
    private No inicio;
```

```
    public Lista() {
```

```
        tam = 0;
```

```
        inicio = null;
```

```
    }
```

```
    public boolean vazia() { return (inicio == null); }
```

```
    public int getTam() { return tam; }
```

```

public Item busca(Item aux) {
    if(vazia()) return null;
    No i = null;
    for(i=inicio; i!=null && !aux.getDescricao().equals(i.dados.getDescricao()); i=i.prox);
    if(i==null) {
        return null;
    }
    Item novo = new Item(i.dados.getDescricao(), i.dados.getQuantidade(), i.dados.getPrioridade());
    return novo;
}

public boolean adicionaOrdenado(Item aux) {
    Item buscado = busca(aux);
    if(buscado != null) return false;
    No novo = new No(aux);
    if(vazia()) {
        inicio = novo;
        tam++;
        return true;
    }
    No i = inicio;
    No ant = inicio;
    for(; i!=null && aux.getDescricao().compareTo(i.dados.getDescricao())>0; ant = i, i=i.prox);
    if(i==ant) {
        novo.prox = inicio;
        inicio = novo;
    }
    else {
        ant.prox = novo;
        novo.prox = i;
    }
    tam++;
    return true;
}

```

```

public Item retira(Item aux) {

    if(vazia()) return null;

    No i = inicio;
    No ant = inicio;

    for(; i!=null && !aux.getDescricao().equals(i.dados.getDescricao()); ant = i, i=i.prox);

    if(i==null) { return null; }

    Item novo = i.dados;

    if(i==ant) { inicio=inicio.prox; }

    else { ant.prox = i.prox; }

    tam--;
    return novo;
}

public String imprima() {
    String aux = new String("");

    if(vazia()) { aux = "Lista vazia."; }

    else {
        for(No i = inicio; i!=null; i=i.prox) {
            aux = aux + i.dados.toString();
        }
    }

    return aux;
}
}

```



```
package listaligadainsercaoinicio;

import java.util.Scanner;

public class Main {

    static Scanner leia = new Scanner(System.in);

    public static Item obterItem() {
        String desc;
        int qt, pri;
        leia.skip("\\R");
        System.out.println("Descrição:");
        desc = leia.nextLine();
        System.out.println("Quantidade:");
        qt = Integer.parseInt(leia.next());
        System.out.println("Prioridade:");
        pri = Integer.parseInt(leia.next());
        Item aux = new Item(desc, qt, pri);
        return aux;
    }
}
```

```
public static int menu() {  
    int valor;  
    System.out.println("Digite:");  
    System.out.println("1 - para adicionar um item.");  
    System.out.println("2 - para remover um item.");  
    System.out.println("3 - para pesquisar um item.");  
    System.out.println("4 - para imprimir a lista.");  
    System.out.println("5 - para encerrar o programa.");  
    valor = Integer.parseInt(leia.next());  
    return valor;  
}
```

```
public static void main(String[] args) {  
  
    Lista lista = new Lista();  
  
    int n;  
  
    Item novo = null;
```

```
do {  
    n = menu();  
    switch(n) {  
  
        case 1:  
            novo = obterItem();  
            lista.adicionaFim(novo);  
            System.out.println("Lista com " + lista.getTam() + " itens.");  
            novo = null;  
            Break;  
  
        case 2:  
            novo = obterItem();  
            novo = lista.retira(novo);  
  
            if(novo == null) System.out.println("Erro!");  
  
            else System.out.println(novo.toString());  
  
            System.out.println("Lista " + lista.getTam() + " itens.");  
            novo = null;  
            break;  
    }  
}
```

case 3:

```
    novo = obterItem();  
    novo = lista.busca(novo);
```

```
    if(novo == null) System.out.println("Erro!");
```

```
    else System.out.println(novo.toString());
```

```
    System.out.println("Lista " + lista.getTam() + " itens.");  
    novo = null;  
    Break;
```

case 4:

```
    System.out.println("Lista " + lista.getTam() + " itens.");  
    System.out.println(lista.imprima());  
    Break;
```

case 5:

```
    System.out.println("Lista com " + lista.getTam() + " itens.");  
    System.out.println("Programa encerrando!");
```

```
} // fim switc() ... case
```

```
} while(n != 5); // fim do .. while
```

```
} // fim da função main
```

```
} // fim classe Main
```