**AEDS II**

**Unidade 06**

**Estruturas de dados básicos flexíveis.**

Parte A: Introdução

Célula: Uma estrutura de dados que armazena um dado ou mais dados (Int, char, class, etc..) aponta para outro local da memoria podendo ser mais uma célula ou um local vazio (NULL).

Vantagens: Memoria flexível

Desvantagens: Consumo maior de memória (\*)

class Celula {

public int num; // Dado contido na célula.

public Celula prox; // Aponta a célula prox.

  //Construtor da classe vazia.

  public Celula() {

    this(0);

  }

   //Construtor da classe.

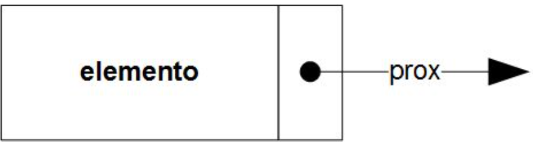
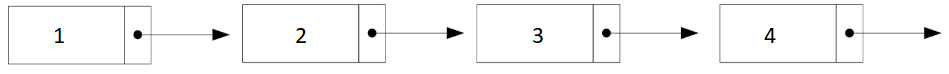
  public Celula(int num) {

   this.num = num;

this.prox = null;

  }

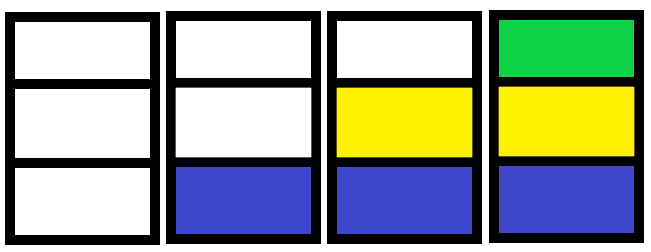
}

Parte B: Pilha (LI-FO)

Pilha: São estruturas de dados do tipo LIFO (last-in first-out), onde o último elemento a ser inserido, será o primeiro a ser retirado. Assim, uma pilha permite acesso a apenas um item de dados - o último inserido. Para processar o penúltimo item inserido, deve-se remover o último.

Inserir-(last-in)

****

**Código em JAVA**

  public void inserirFim(int num) {

    ultimo.prox = new Celula(num);

    ultimo = ultimo.prox;

  }

**Código em C**

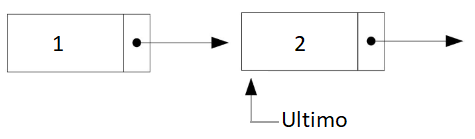
void inserirFim(int num)

{

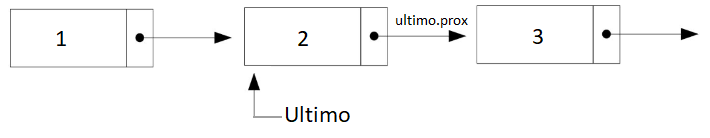
ultimo->prox = novaCelula(num);

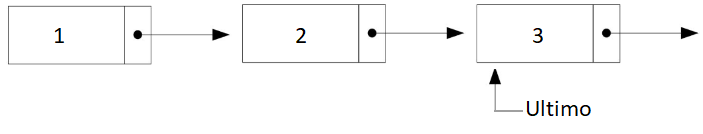
ultimo = ultimo->prox;

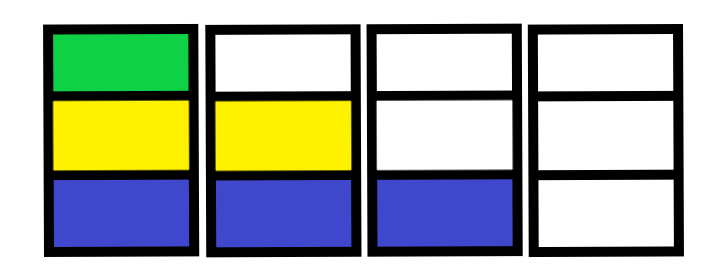
}

****

ultimo.prox = new Celula(int num);

**** ultimo = ultimo.prox;



Remover (first-out) 

**Código em JAVA**

  public Int remover() throws Exception {

    if (primeiro == ultimo) {

      throw new Exception("Erro ao remover (vazia)!");

    }

    // Caminhar ate a penultima celula:

    Celula i;

    for (i = primeiro; i.prox != ultimo; i = i.prox);

    int resp = ultimo.num;

    ultimo = i;

    i = ultimo.prox = null;

    return resp;

  }

**Código em C**

X

X

X

X

X

X

X

X

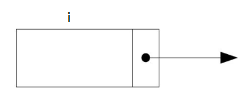
X

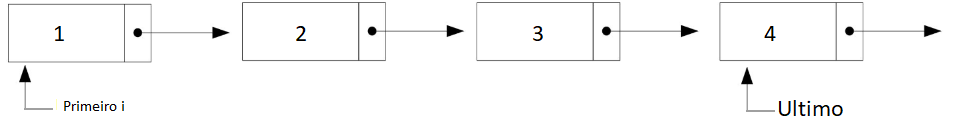
X

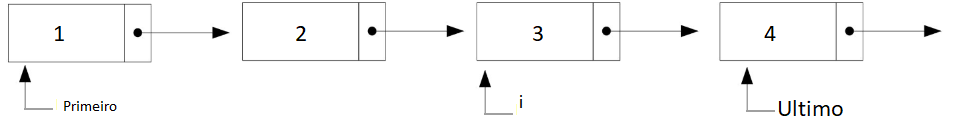
X

x

Celula i;



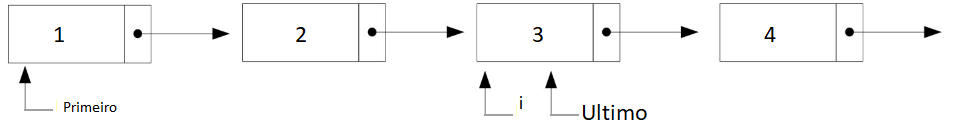
for (i = primeiro; i.prox != ultimo; i = i.prox); 



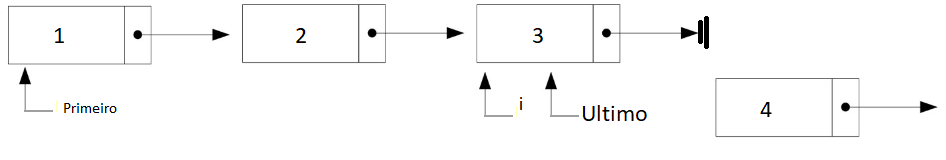
Int resp = ultimo.num;

|  |  |
| --- | --- |
| RESP = 4; |  |

ultimo = i;

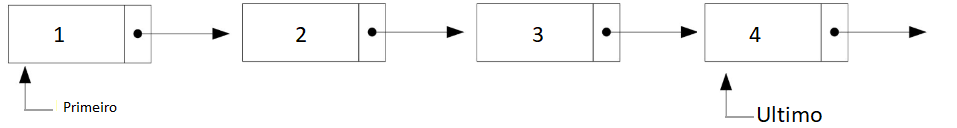


ultimo.prox = null;



Return: RESP = 4;

Mostrar



public void mostrar() {

    System.out.print("[ ");

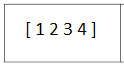
    for (Celula i = primeiro.prox; i != null; i = i.prox) {

      System.out.print(i.elemento + " ");

    }

    System.out.println("] ");

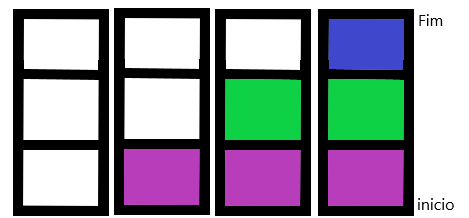
  }



Parte C: Fila (FI-LO)

Fila: São estruturas de dados do tipo LIFO (First-in last-out), onde o primeiro elemento a ser inserido, será o último a ser retirado. Assim, uma Fila permite acesso a apenas um item de dados.

Inserir-(First-in)



**Código em JAVA**

  public void inserirFim(int num) {

    ultimo.prox = new Celula(num);

    ultimo = ultimo.prox;

  }

**Código em C**

void inserirFim(int num)

{

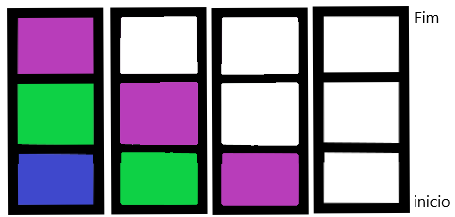
ultimo->prox = novaCelula(num);

ultimo = ultimo->prox;

}

**Consulte a Pag 3 para ver a inserçao sendo feita.**

Remover-(L-o)



**Código em JAVA**

public int remover() throws Exception {

    Celula tmp = primeiro;

    primeiro = primeiro.prox;

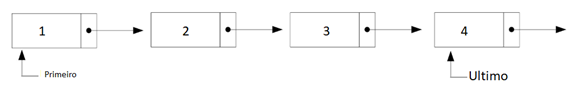
    int resp = primeiro.num;

    tmp.prox = null;

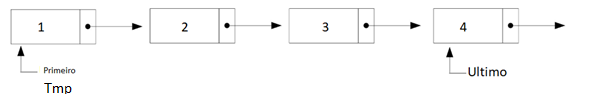
    tmp = null;

    return resp;

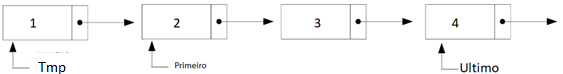
  }



Celula tmp = primeiro;



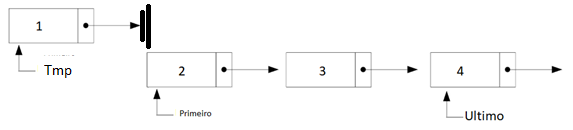
primeiro = primeiro.prox;



Int resp = primeiro.num;

Resp = 1;

tmp.prox = null;



tmp = null;

