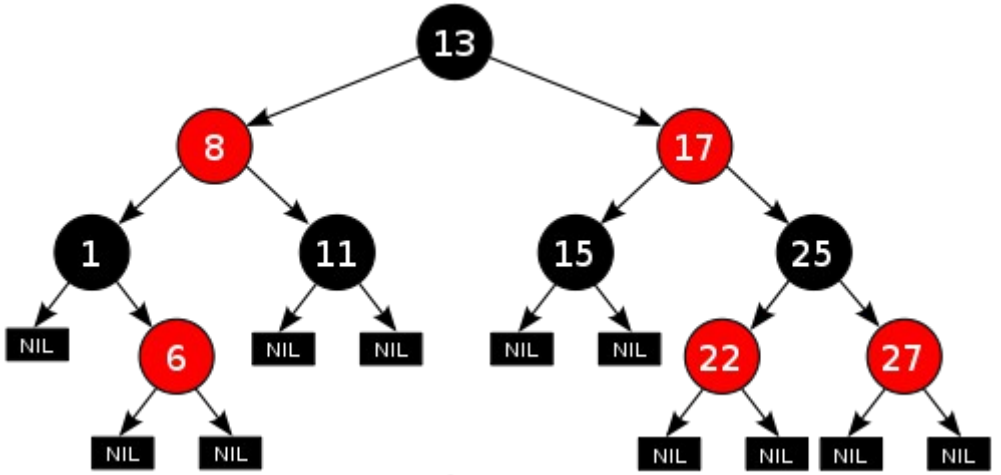
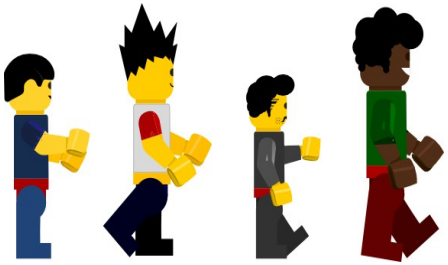
Listas Encadeadas





Linguagem de Programação 2
(Estrutura de Dados)

Agenda

1 – O que são e porque vamos usar?

2 – XXX

3 – XXX

4 – XXX

5 – XXXX

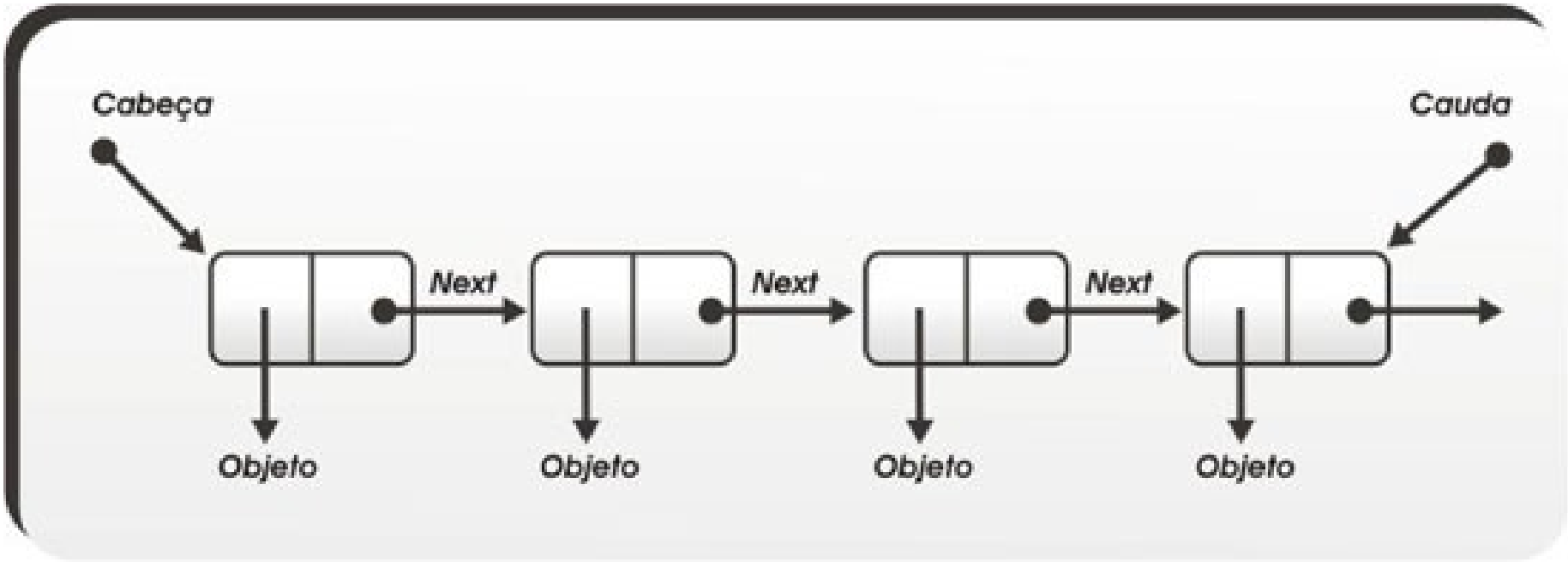
1 – O que são as Listas Encadeadas e porque usar?

Listas Encadeadas

- Vetores são estruturas interessantes e simples, porém vimos que apresentam dificuldades!
  - Exemplo:
    - Se quisermos uma busca rápida, utilizamos um vetor ordenado, porém teremos uma inserção lenta...ou seja, não tem muito “como fugir”.
    - Vetores ao serem instanciados, já tem seu tamanho definido por padrão!
- As Listas Encadeadas são estruturas de dados que irão nos ajudar a resolver alguns desses problemas!!!

Listas Encadeadas

- Listas Encadeadas podem substituir vetores em muitas estruturas de dados!
- Entendendo Listas Encadeadas
  - Cada item em uma lista encadeada é representado por um **nó**, que além dos dados dele, contém uma **referência para o próximo nó**!!!



Listas Encadeadas

- Tem uma característica de JAVA quanto a REFERÊNCIA de um objeto, que torna para nós as listas encadeadas interessante!!!
  - Considere:
    - Pessoa pessoa1 = new Pessoa();
    - Aqui estamos criando um objeto to tipo pessoa na memória. Pessoa1 não esta localizado junto da área da memória onde foi criado este novo objeto … ele é somente "aponta" para essa área, ou seja é só uma referência, então quando fazemos:
    - Pessoa pessoa2 = pessoa1.
    - Não estamos fazendo cópia do objeto, somente estamos colocando uma referência a um objeto do tipo Pessoa chamado pessoa1 em pessoa2. É só uma referência!!!

Um exemplo simples de Lista Encadeada

Listas Encadeadas – Exemplo 1

- **Passo 1**: Criamos a classe que será nosso nó

```
class No
   {
   private int iData;                // vai armazenar um valor numerico qualquer
   private double dData;             // vai armazenar um valor numerico qualquer
   private No next;                 // referência para o próximo item da lista
// --------------------------------------------------------------
   public No(int id, double dd) // metodo construtor
      {
      iData = id;                   // initializa os dois atributos do No
      dData = dd;                   // o No next a principio fica como null
      }                             //
// --------------------------------------------------------------
   public void displayNo()      // mostra os valores do proprio nó
      {
      System.out.print("{" + iData + ", " + dData + "} ");
      }

   public void setNext(No n){
       next = n;
   }

   public No getNext(){
       return next;
   }
   }  // fim da classe nó
////////////////////////////////////////////////////////////////
```

Listas Encadeadas – Exemplo 1

- **Passo 2**: Criamos os métodos que controlarão nossa lista

```
class ListaDeNos
   {
   private No first;           // criamos uma referencia para o primeiro nó da lista
// ----------------------------------------------------------
   public ListaDeNos()             // metodo construtor
      {
      first = null;               // quando construido o primeiro nó fica como NULL, porque não tem nada!
      }
// ----------------------------------------------------------
   public boolean isEmpty()       // se o primeiro (first) nó estiver como NULL
      {                             // estará vazia
      return (first==null);
      }
// ----------------------------------------------------------
   public void insertFirst(int id, double dd)  // insere o primeiro nó
      {
      No newNo = new No(id, dd);
      newNo.setNext(first);       // colocamos o antigo "primeiro nó" na referência do que esta sendo ins
      first = newNo;             // e aí colocamos o que esta sendo inserido como o primeiro
      }
// ----------------------------------------------------------
   public No deleteFirst()       // removemos o primeiro nó
      {                             // assumimos que a lista não esta vazia!
      No temp = first;           // o primeiro nó vai p/ variavel temporária
      first = first.getNext();         // o segundo nó se torna o novo primeiro
      return temp;               // retornarmos para classe usuária o item removido
      }
```

Listas Encadeadas – Exemplo 1

- **Passo 3**: Criamos uma classe usuária

```
class ListaDeNosApp
   {
   public static void main(String[] args)
      {
      ListaDeNos theList = new ListaDeNos();  // criamos uma nova lista

      theList.insertFirst(22, 2.99);      // inserimos quatro itens nela
      theList.insertFirst(44, 4.99);
      theList.insertFirst(66, 6.99);
      theList.insertFirst(88, 8.99);

      theList.displayList();              // mostramos a lista

      while( !theList.isEmpty() )         // until it's empty,
         {
         No aLink = theList.deleteFirst();   // deletamos o primeiro nó da lista e colocamos em aLink
         System.out.print("Deletado ");          //
         aLink.displayNo();                      // mostramos o nó deletado
         System.out.println("");
         }
      theList.displayList();              // mostramos a lista
      }  // final do método main()
   }  // final da classe usuária
```

Exercício - E7

- Crie utilizando Listas Encadeadas, uma lista que armazene em cada nó, o nome de cada colega e a “pessoa” querida que este escolheu!
- Consultando o capítulo 5 do livro do Lafore, implemente um mecanismo de busca linear em seu algoritmo

Listas Encadeadas – Parte II

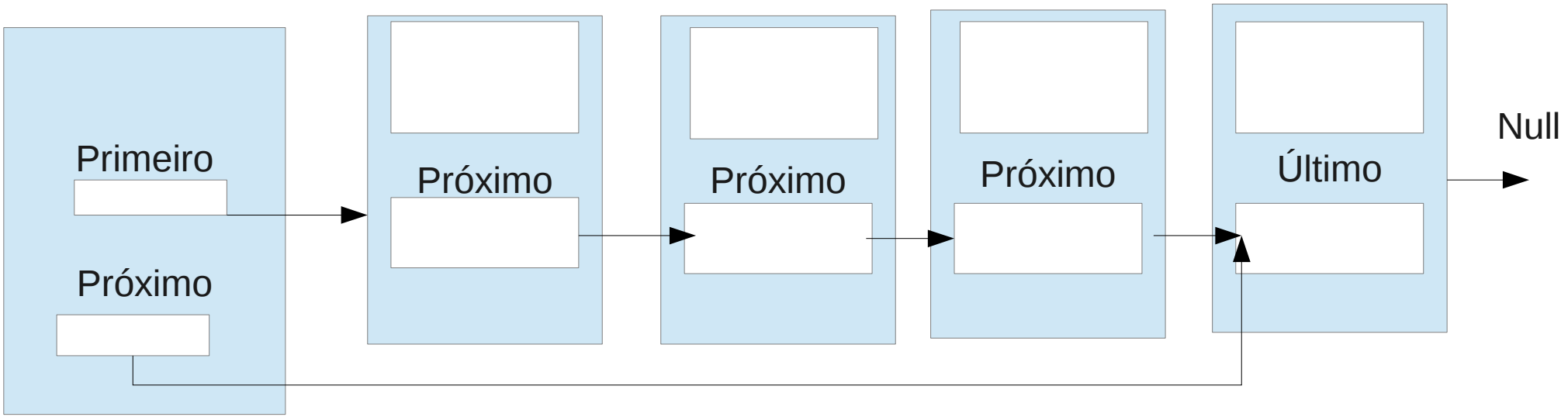
Desafio 1

Como adicionar um método para excluir um item específico da lista?

Listas de Extremidades Duplas

Listas de Extremidades Duplas

- A lista de extremidade dupla funciona de forma bastante parecida com a lista simples, porém, possui uma referência para o último nó inserido, além de uma referência para o primeiro nó!



Listas de Extremidades Duplas

- Quais as vantagens de se ter uma Lista de Extremidades Duplas?

Vamos implementar uma lista com extremidades duplas?

Listas de Extremidades Duplas

- Prática:
  - Um jeito de entender bem como funciona uma lista de extremidade dupla é criar através dela as TAD Pilha e Fila!!!
  - Boa Sorte!!!

Listas Ordenadas

Listas Ordenadas

- Listas ordenadas podem nos auxiliar em uma busca binária futuramente!!!
  - O único método que precisamos alterar para criar uma lista ordenada é o método INSERT().
  - Qual a lógica?

Exercício - E8

- Utilizando seus conceitos e todas as técnicas aprendidas no Exercício 7, implemente no Exercício 7:
  - O parametro **idade** no nó;
  - Desenvolva seu sistema de tal forma que a inserção seja ordenada!!!
  - Implemente um método de busca por nome e outro por idade!
- Desafio HARD: Implementar busca binária por idade no sistema!!!

Referências

LAFORE, Robert. Estruturas de Dados & algoritmos em Java. Rio de Janeiro. 2004

Visão Geral da Estrutura de Dados

Dúvidas? Questionamentos?
ale.garcia.aguado@gmail.com