

Objetos Distribuídos



Aula 02

➤ Algumas configurações:

➤ Mysql:

- instalar o banco de dados

(<http://www.mysql.com/downloads>)

- Baixar o conector

(<http://dev.mysql.com/downloads/connector/j/>)

➤ PostgreSQL:

- Instalar o banco de dados PostgreSQL

(<http://www.postgresql.org/download/>)

- Instalar o conector

(<http://jdbc.postgresql.org/download.html>)

Coleções – Iniciando por List

- interfaces-chave da API: List, Set e Map
- List: Uma coleção é, dito de forma simples, uma estrutura que agrupa vários objetos em um único
- Características: pode repetir, funcionam com vetores (arrays)
- Principais: ArrayList, LinkedList

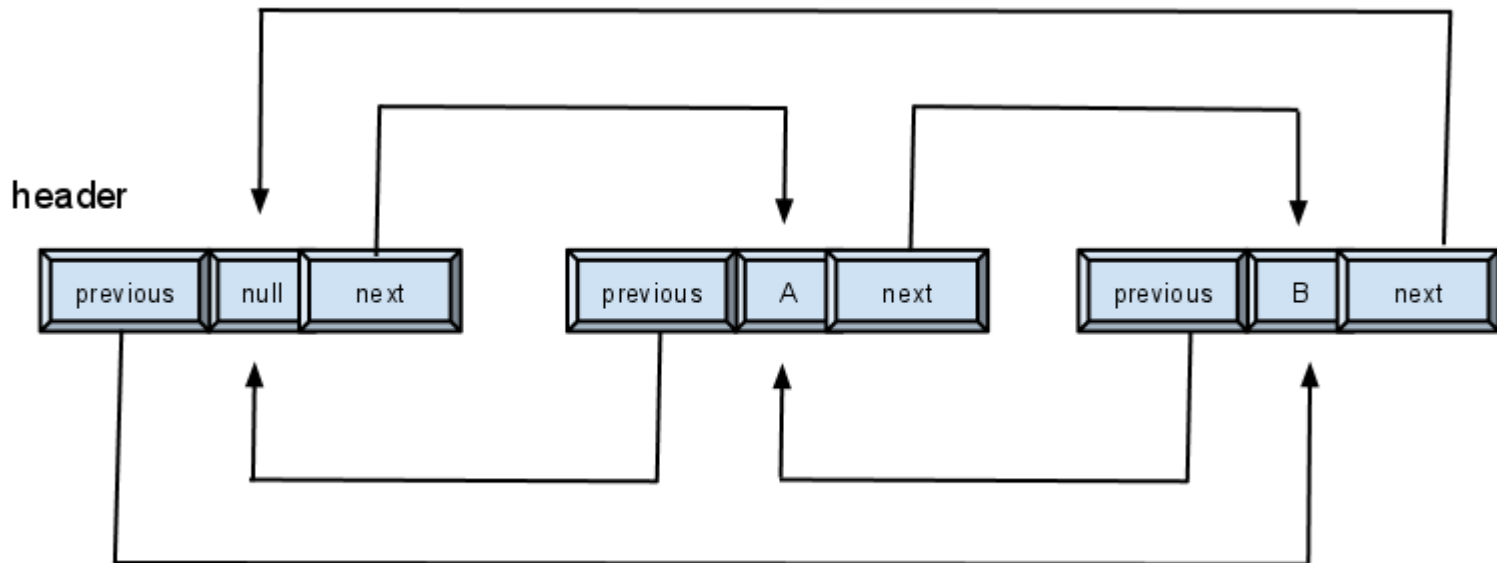
ArrayList

- É uma lista simples. Utilizar para tamanhos definidos ou pequenos
- Principais métodos: `get()`, `set()` e `add()`

Exercício 2

LinkedList

- lista duplamente encadeada ou ligada.



significa que cada elemento da lista possui um link para o próximo elemento e para o elemento anterior a ele mesmo

Continuação...

- Desvantagem: bem mais lenta que o ArrayList
- Métodos: push, pop, set, remove
- suportam inserção e remoção nas duas extremidades
- e sempre o primeiro a ser atendido é o primeiro da fila (FIFO)

Exercício 3

Qual o melhor?

- caso seu programa faça muitos acessos randômicos a lista, recomenda-se utilizar ArrayList e LinkedList quando precisar fazer muito add / remove.

Set

- Esta interface representa um conjunto.
- Características: não possuem elementos duplicados.
- Principais implementações: HashSet e TreeSet
- Diferenças: TreeSet é classificada, HashSet, não é classificada nem ordenada

HashSet

- É mais rápida que a TreeSet para as operações de inserção e exclusão
- Métodos : add, remove,...
- Não estabelece nenhum tipo de ordem

TreeSet

- Ordena os elementos
 - Exercício
 - Se for inserir e remover, este é mais lento
-
- Exercício 4

Map

- A interface `Map` especifica as funcionalidades necessárias para manipular um grupo de objetos onde cada elemento é um par que associa um objeto chave a um objeto valor.
- Principal: `HashMap`
- utilizamos um `HashMap` quando queremos associar chaves e valores, para posteriormente, recupera-los por meio de chaves.

HashMap (continuação)

Funciona literalmente como um “mapa”. Através de um endereço (chave) você localiza o local (valor)

Exercício 5