Linked List

- Objectives:
  - Discuss linked lists
    - Syntax
    - Implementation
Linked List

- **Characteristics:**
  - A collection
    - Holds objects to store, retrieve and manipulate
  - Java Library
    - LinkedList
    - Store type Object
    - In package java.Util

- **Definition:**
  - It is a data structure including a list of nodes so that each node is composed of the object and a reference to another node.
Linked List

• We must declare the class Node

```java
class Node {
    <ObjectType> <ObjectName>;
    Node link;
}
```

– Example a Node of a Rational number.
Linked List

- Example a Node of a Rational number.

class RationalNode
{
    Rational myRat;
    RationalNode link;
}

Linked List

• Declaration:
  – Syntax
    RatNode rPtr;
  – Meaning:
    • Allocate a memory to hold a reference of a RatNode
    • NO RatNode is created therefore, the reference is null
  – Instantiate:
    rPtr = new RatNode();
**Constructor**

- **Constructor**
  
  ```java
  RatNode ( Rational r, RatNode lPtr)
  {
      myRat = r;
      link = lPtr;
  }
  ```

- **Invoke**
  
  ```java
  new RatNode( new Rational(2,3), null)
  ```
Linked List

• Look at this instruction
  
  RatNode head =
  
  new RatNode(new Rational(2,3), null);
  – A new object 2/3 (Rational) is created
  – A new object of RatNode is created
  – The reference of RatNode is stored in head.
Linked List

• Look at this instruction

```java
RatNode head =
    new RatNode(new Rational(2,3), head);
```
– Compare this instruction to the one before.
Linked List

• Loop

    For (int j= 0; j < max; j++)
    {
        // create a new object from Rational class,
        // this new object must have a reference in r

        // analyze this instruction
        head = new RatNode(r, head);
    }
Linked List

summary

- store many objects with references
- are created by new