MAIN

|  |
| --- |
| package com.company; |
|  | import java.util.ArrayList; |
|  | import java.util.Scanner; |
|  |  |
|  | public class Main { |
|  |  |
|  | public static void main(String[] args) { |
|  | MyArrayList<Integer> list = new MyArrayList<Integer>(); |
|  | MyLindekList<Integer> link = new MyLindekList<Integer>(); |
|  | int size = 5; |
|  | for (int i = 0; i < size; i++) { |
|  | list.add(1 + i, i); |
|  | link.add(2 + i, i); |
|  | } |
|  | System.out.println("Arraylist: " + list); |
|  | System.out.println("LinkedList: " + link + '\n'); |
|  |  |
|  | System.out.println("You found:" + list.find(4)); |
|  | System.out.println("You found:" + link.find(3)); |
|  |  |
|  | list.reverse(); |
|  | link.reverse(); |
|  | System.out.println("After reversing:"); |
|  | System.out.println("Arraylist: " + list); |
|  | System.out.println("LinkedList: " + link + '\n'); |
|  |  |
|  | System.out.println("Removed number: " + list.remove(2)); |
|  | System.out.println("Removed number: " + list.remove(2) + '\n'); |
|  | System.out.println("After removing"); |
|  | System.out.println("Arraylist: " + list); |
|  | System.out.println("LinkedList: " + link + '\n'); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | }  MYARRAY LIST |
| package com.company; |
|  |  |
|  | import java.util.Arrays; |
|  |  |
|  | public class MyArrayList<T> { |
|  | private Object[] arr; |
|  | private int Size = 0; |
|  | private int Cap = 5; |
|  |  |
|  | public MyArrayList() { |
|  | arr = new Object[Cap]; |
|  | } |
|  | public void add(T newItem, int index) { |
|  | if (Size == arr.length) { |
|  | resize(2 \* arr.length); |
|  | } |
|  | for (int i = Size - 1; i >= index; i--) { |
|  | arr[i + 1] = arr[i]; |
|  | } |
|  | arr[index] = newItem; |
|  | Size++; |
|  | } |
|  |  |
|  | public int find(T keyItem) { |
|  | for (int i = 0; i < Size; i++) { |
|  | if (arr[i].equals(keyItem)) { |
|  | return i; |
|  | } |
|  | } |
|  | return -1; |
|  | } |
|  |  |
|  | public int remove(int index) { |
|  | Object result = arr[index]; |
|  | for (int i = index; i < Size - 1; i++) { |
|  | arr[index] = arr[index + 1]; |
|  | } |
|  | Size--; |
|  | arr[Size] = null; |
|  | return (int) result; |
|  | } |
|  |  |
|  | public void reverse() { |
|  | int temp; |
|  | int start = 0; |
|  | int end = Size - 1; |
|  | while (start < end) { |
|  | temp = (int) arr[start]; |
|  | arr[start] = arr[end]; |
|  | arr[end] = temp; |
|  | start++; |
|  | end--; |
|  | } |
|  | } |
|  |  |
|  | public void resize(int Cap) { |
|  | T[] newData = (T[]) new Object[Cap]; |
|  | for (int i = 0; i < Size; i++) { |
|  | newData[i] = (T) arr[i]; |
|  | } |
|  | arr = newData; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "MyArrayList{" + |
|  | "arr=" + Arrays.toString(arr) + |
|  | ", Size = " + Size + |
|  | ", Cap = " + Cap + |
|  | '}'; |
|  | } |
|  | } |

Linked List

|  |
| --- |
| package com.company; |
|  |  |
|  | public class MyLindekList<T> { |
|  | private MyNode<T> head; |
|  | private MyNode<T> tail; |
|  | private int size; |
|  |  |
|  | public MyLindekList() { |
|  | size = 0; |
|  | } |
|  | public void add(T newItem, int index) { |
|  | MyNode<T> newNode = new MyNode<>(newItem); |
|  | if (head == null) |
|  | head = tail = newNode; |
|  | else { |
|  | tail.next = newNode; |
|  | tail = newNode; |
|  | } |
|  | size++; |
|  | } |
|  |  |
|  | public int find(T keyItem) { |
|  | MyNode<T> current = head; |
|  | for (int i = 0; i < size; i++) { |
|  | current = current.next; |
|  | } |
|  | return 0; |
|  | } |
|  |  |
|  | public void remove(int index) { |
|  |  |
|  | } |
|  |  |
|  | public void reverse() { |
|  |  |
|  | } |
|  |  |
|  | private static class MyNode<E> { |
|  | E data; |
|  | MyNode<E> next; |
|  |  |
|  | MyNode(E data) { |
|  | this.data = data; |
|  | } |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "MyLindekList{" + |
|  | "head = " + head + |
|  | ", tail = " + tail + |
|  | ", size = " + size + |
|  | '}'; |
|  | } |
|  | } |