***Dt : 12/11/2022***

***=>Limitation of ArrayList<E> can be overcomed using LinkedList<E>.***

***LinkedList<E>:***

***=>In LinkedList<E> the elements are available in NonSequence and the elements***

***are in the form of nodes.***

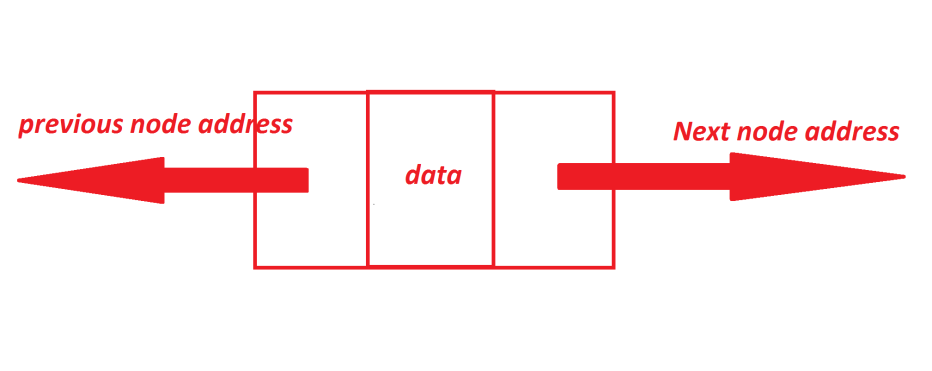
***=>LinkedList<E> node is divided into the following three partitions:***

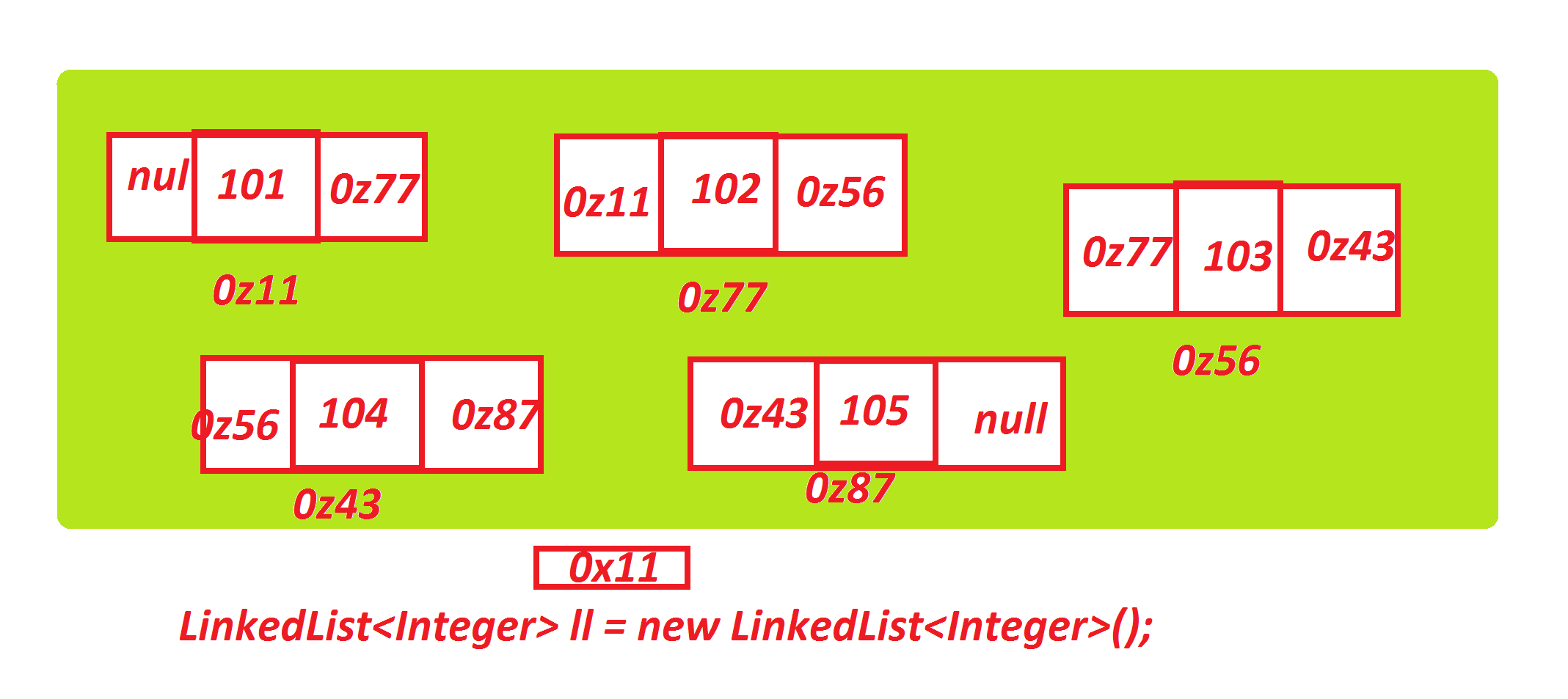
***(i)Previous Node Address***

***(ii)data***

***(iii)Next Node Address***

***Diagram:***

******

******

***=>In realtime LinkedList<E> is used in applications where we have more number***

***of add() and remove() operations.***

***=>LinkedList<E> in Java is known as Double-linked-list.***

***Ex:***

***package maccess;***

***import java.util.\*;***

***public class DemoLinkedList {***

***@SuppressWarnings("removal")***

***public static void main(String[] args) {***

***LinkedList<Integer> ll = new LinkedList<Integer>();***

***for(int i=101;i<=105;i++) {***

***ll.add(new Integer(i));***

***}//end of loop***

***System.out.println(ll.toString());***

***}***

***}***

***===================================================================***

***\*imp***

***Vector<E>:***

***=>Vector<E> organizes elements in Sequence and which is synchronized class.***

***=>Vector<E> is Legacy class in JCF and which is available from many versions.***

***=>The following are some important methods from Vector<E>:***

***public synchronized E elementAt(int);***

***public synchronized E firstElement();***

***public synchronized E lastElement();***

***public synchronized void setElementAt(E, int);***

***public synchronized void removeElementAt(int);***

***public synchronized void insertElementAt(E, int);***

***public synchronized void addElement(E);***

***public synchronized boolean removeElement(java.lang.Object);***

***public synchronized void removeAllElements();***

***Ex : DemoVector.java***

***package maccess;***

***import java.util.\*;***

***public class DemoVector {***

***@SuppressWarnings("removal")***

***public static void main(String[] args) {***

***Vector<Integer> v = new Vector<Integer>();***

***for(int i=201;i<=205;i++) {***

***v.addElement(new ~~Integer~~(i));***

***}//end of loop***

***System.out.println(v.toString());***

***System.out.println("ele at index 2:"+v.elementAt(2));***

***System.out.println("first element:"+v.firstElement());***

***System.out.println("last elements:"+v.lastElement());***

***v.setElementAt(new ~~Integer~~(500), 2);***

***System.out.println(v.toString());***

***v.removeElementAt(2);***

***System.out.println(v.toString());***

***v.insertElementAt(new ~~Integer~~(203), 2);***

***System.out.println(v.toString());***

***v.removeElement(new ~~Integer~~(203));***

***System.out.println(v.toString());***

***System.out.println("size:"+v.size());***

***v.removeAllElements();***

***System.out.println("size:"+v.size());***

***}***

***}***

***o/p:***

***[201, 202, 203, 204, 205]***

***ele at index 2:203***

***first element:201***

***last elements:205***

***[201, 202, 500, 204, 205]***

***[201, 202, 204, 205]***

***[201, 202, 203, 204, 205]***

***[201, 202, 204, 205]***

***size:4***

***size:0***

***--------------------------------------------------------------***

***Note:***

***=>In realtime Vector<E> is used in Connection Popling Concept,which means***

***organizing multiple Database connection among multiple users.***

***===========================================================================***

***define Stack<E>?***

***=>Stack<E> is a ChildClass of Vector<E> and which organizes elements based on***

***the algorithm First-In-Last-Out or Last-In-First-Out.***

***=>The following are some important methods of Stack<E>:***

***public E push(E);***

***public synchronized E pop();***

***public synchronized E peek();***

***public boolean empty();***

***public synchronized int search(java.lang.Object);***

***=>seach() method searches the ele from the stack and display position of an***

***ele.***

***Ex : DemoStack.java***

***package maccess;***

***import java.util.\*;***

***public class DemoStack {***

***public static void main(String[] args) {***

***Scanner s = new Scanner(System.in);***

***Stack<Integer> ob = new Stack<Integer>();***

***try(s;){***

***try {***

***while(true) {***

***System.out.println("\*\*\*\*\*Choice\*\*\*\*");***

***System.out.println("1.push\n2.pop\n3.peek\n4.search\n5.exit");***

***System.out.println("Enter the Choice:");***

***switch(s.nextInt()) {***

***case 1:***

***System.out.println("Enter the element:");***

***ob.push(new ~~Integer~~(s.nextInt()));***

***System.out.println(ob.toString());***

***break;***

***case 2:***

***if(ob.empty()) {***

***System.out.println("Stack is empty...");***

***}else {***

***ob.pop();***

***System.out.println("Ele removed from Stack...");***

***System.out.println(ob.toString());***

***}***

***break;***

***case 3:***

***if(ob.empty()) {***

***System.out.println("Stack is empty...");***

***}else {***

***System.out.println("peek ele : "+ob.peek());***

***System.out.println(ob.toString());***

***}***

***break;***

***case 4:***

***if(ob.empty()) {***

***System.out.println("Stack is empty...");***

***}else {***

***System.out.println("Enter the ele to be Searched:");***

***Integer ele = new ~~Integer~~(s.nextInt());***

***int p = ob.search(ele);***

***if(p>0) {***

***System.out.println("ele found at position : "+p);***

***}else {***

***System.out.println("Ele not founded....");***

***}***

***}***

***break;***

***case 5:***

***System.out.println("Stack operations Stopped..");***

***System.exit(0);***

***default:***

***System.out.println("Invalid Choice...");***

***}//end of switch***

***}//end of loop***

***}catch(Exception e) {e.printStackTrace();}***

***}//end of try***

***}***

***}***

***o/p:***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***4***

***Stack is empty...***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***1***

***Enter the element:***

***11***

***[11]***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***1***

***Enter the element:***

***12***

***[11, 12]***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***1***

***Enter the element:***

***13***

***[11, 12, 13]***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***1***

***Enter the element:***

***14***

***[11, 12, 13, 14]***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***1***

***Enter the element:***

***15***

***[11, 12, 13, 14, 15]***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***4***

***Enter the ele to be Searched:***

***11***

***ele found at position : 5***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***4***

***Enter the ele to be Searched:***

***15***

***ele found at position : 1***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***4***

***Enter the ele to be Searched:***

***200***

***Ele not founded....***

***\*\*\*\*\*Choice\*\*\*\****

***1.push***

***2.pop***

***3.peek***

***4.search***

***5.exit***

***Enter the Choice:***

***5***

***Stack operations Stopped..***

***=========================================================================***