Міністерство освіти і науки України

НТУУ «КПІ ім. Ігоря Сікорського»

Лабораторна робота №4

**«**Зв’язані списки»

Виконала:

Студентка 2-ого курсу ТЕФ

Групи ТВ-91

Таратонова Олександра

Перевірив:

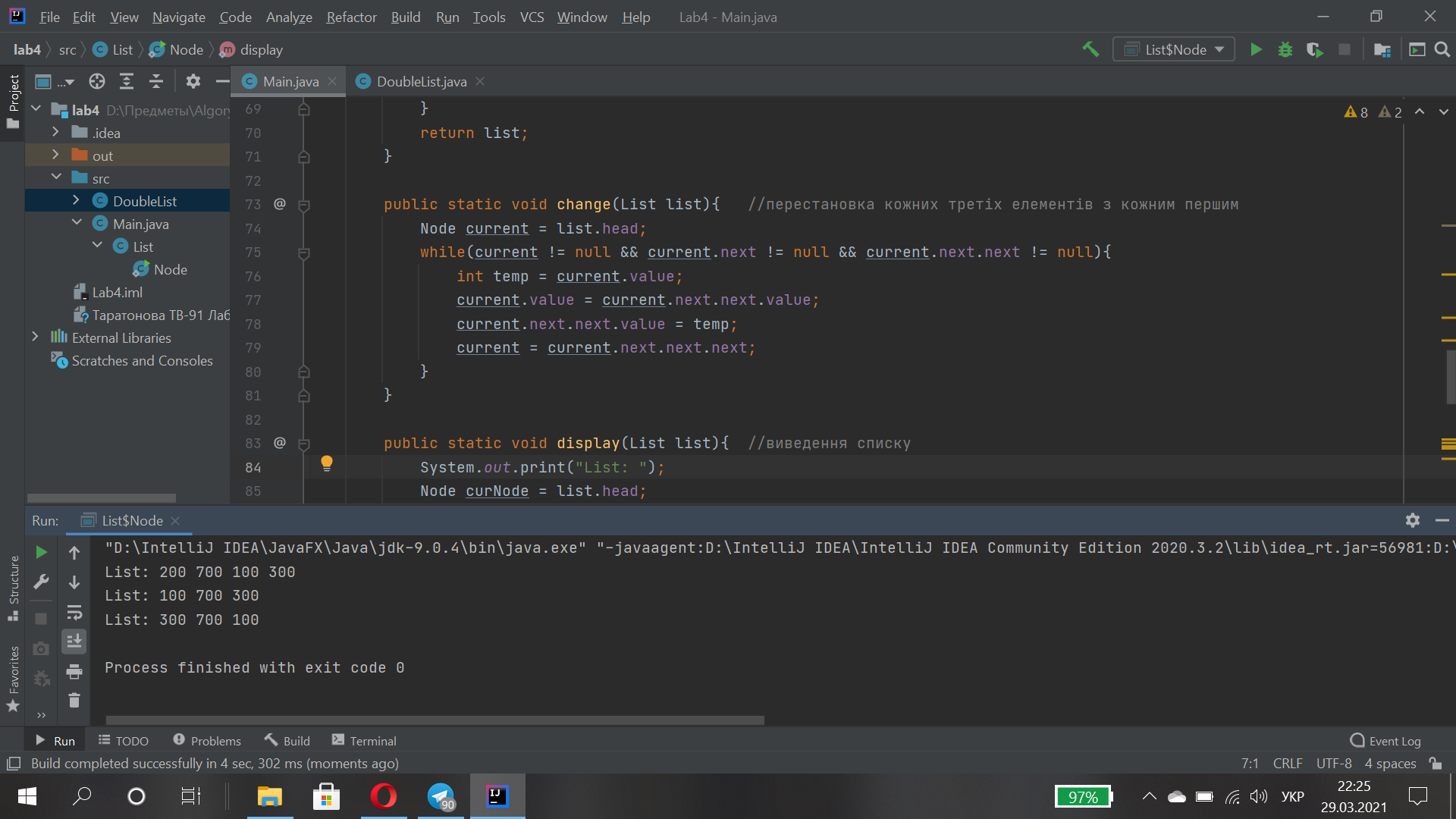
Кузьменко І. М.

КИЇВ 2021

Код програми однозв’язний список

class List {  
 Node head;  
  
 static class Node{  
 int value;  
 Node next;  
  
 Node(int v){  
 value = v;  
 next = null;  
 }  
  
 public static List insert(List list, int data, int pos){ //вставка елементів в будь-яке місце  
 pos = Math.*abs*(pos);  
 Node curNode = new Node(data); //новий вузел  
 curNode.next = null;  
  
 if(list.head == null){  
 list.head = curNode;  
 } else {  
 Node last = list.head;  
 if(pos == 0){ //якщо елемент буде першим  
 curNode.next = list.head;  
 list.head = curNode;  
 } else {  
 int i = 1;  
 while (last.next != null && i < pos) {  
 last = last.next;  
 i++;  
 }  
 if(last.next == null) last.next = curNode;  
 else {  
 curNode.next = last.next;  
 last.next = curNode;  
 }  
 }  
 }  
 return list;  
 }  
  
 public static List erase(List list, int data){ //видалення будь-якого елемента, наявного в списку  
 Node toErase = list.head;  
 if(toErase.value == data){ //перший елемент  
 list.head = toErase.next;  
 toErase = toErase.next;  
 if(toErase.next != null && toErase != null){ //перестановка елементів  
 int temp = toErase.value;  
 toErase.value = toErase.next.value;  
 toErase.next.value = temp;  
 }  
 } else {  
 while (toErase.next != null && toErase.next.value != data) {  
 toErase = toErase.next;  
 }  
 if (toErase.next != null) {  
 if(toErase.next.next != null){ //елемент в середині масива  
 toErase.next = toErase.next.next;  
 toErase = toErase.next;  
 if(toErase.next != null && toErase != null){ //переставляємо  
 int temp = toErase.value;  
 toErase.value = toErase.next.value;  
 toErase.next.value = temp;  
 }  
 } else{ //останній елемент  
 toErase.next = null;  
 toErase = toErase.next;  
 }  
 }  
 }  
 return list;  
 }  
  
 public static void change(List list){ //перестановка кожних третіх елементів з кожним першим  
 Node current = list.head;  
 while(current != null && current.next != null && current.next.next != null){  
 int temp = current.value;  
 current.value = current.next.next.value;  
 current.next.next.value = temp;  
 current = current.next.next.next;  
 }  
 }  
  
 public static void display(List list){ //виведення списку  
 System.*out*.print("List: ");  
 Node curNode = list.head;  
 while (curNode != null) {  
 System.*out*.print(curNode.value + " ");  
 curNode = curNode.next;  
 }  
 System.*out*.println();  
 }  
  
 public static void main(String[] args) {  
 List firstList = new List();  
  
 firstList = *insert*(firstList,100, 1);  
 firstList = *insert*(firstList,200, 0);  
 firstList = *insert*(firstList,300, 3);  
 firstList = *insert*(firstList,700, 1);  
  
 *display*(firstList);  
 firstList = *erase*(firstList, 200);  
 *display*(firstList);  
  
 *change*(firstList);  
 *display*(firstList);  
 }  
 }  
}

Результат роботи



Код програми двозв’язний список

public class DoubleList {  
 DoubleList.Node head;  
  
 static class Node {  
 int value;  
 DoubleList.Node prev;  
 DoubleList.Node next;  
  
 Node(int v) {  
 value = v;  
 prev = null;  
 next = null;  
 }  
  
 public static DoubleList insert(DoubleList list, int data, int pos){ //вставка елементів в будь-яке місце  
 pos = Math.*abs*(pos);  
 DoubleList.Node curNode = new DoubleList.Node(data); //новий вузел  
 curNode.next = null;  
 curNode.prev = null;  
  
 if(list.head == null){ //якщо елемент буде першим  
 list.head = curNode;  
 } else {  
 DoubleList.Node last = list.head;  
 if(pos == 0){  
 list.head.prev = curNode;  
 curNode.next = list.head;  
 list.head = curNode;  
 } else {  
 int i = 1;  
 while (last.next != null && i < pos) {  
 last = last.next;  
 i++;  
 }  
 if (last.next == null){  
 curNode.prev = last;  
 last.next = curNode;  
 }  
 else {  
 curNode.prev = last.prev;  
 curNode.next = last.next;  
 last.next = curNode;  
 }  
 }  
 }  
 return list;  
 }  
  
 public static DoubleList erase(DoubleList list, int data){ //видалення будь-якого елемента, наявного в списку  
 DoubleList.Node toErase = list.head;  
 if(toErase.value == data){ //перший елемент  
 list.head = toErase.next;  
 toErase = toErase.next;  
 toErase.prev = null;  
 } else {  
 while (toErase.next != null && toErase.next.value != data) {  
 toErase = toErase.next;  
 }  
 if (toErase.next != null) {  
 if(toErase.next.next != null){ //елемент в середині масива  
 toErase.next.next.prev = toErase;  
 toErase.next = toErase.next.next;  
 toErase = toErase.next;  
 } else{ //останній елемент  
 toErase.next = null;  
 toErase = toErase.next;  
 }  
 }  
 }  
 return list;  
 }  
  
 public static void display(DoubleList list){ //виведення списку  
 System.*out*.print("List: ");  
 DoubleList.Node curNode = list.head;  
 while (curNode != null) {  
 System.*out*.print(curNode.value + " ");  
 curNode = curNode.next;  
 }  
 System.*out*.println();  
 }  
  
 public static void main(String[] args) {  
 DoubleList firstList = new DoubleList();  
  
 firstList = *insert*(firstList,100, 1);  
 firstList = *insert*(firstList,200, 0);  
 firstList = *insert*(firstList,300, 3);  
 firstList = *insert*(firstList,700, 1);  
  
 *display*(firstList);  
 firstList = *erase*(firstList, 300);  
 *display*(firstList);  
 }  
 }  
}

Результат роботи

