**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №11**

з дисципліни

# «Об’єктно-орієнтоване програмування»

**Виконав:**

студент групи КН-107

Бєлан В.Ю

**Прийняв:**

Асистент кафедри СШІ

Швороб І.Б

Львів – 2018 р.

**Код програми:**

# 1)

**package** week11;

**public** **class** Career { **public** String place; **public** String position; **public** String price; **public** String passport; **public** String date;

}

# 2)

**package** week11;

**public** **class** Execute {

**public** **static** **void** main(String[] args) { Career career = **new** Career(); career.place = "MicrosoftWindows"; career.position = "Java Developer"; career.price = "100000";

LinkedContainer<Career> list = **new** LinkedContainer<Career>(); list.add(career.place); list.add(career.position); list.add(career.price); list.delete(1);

System.***out***.println(list.toString()); list.clear();

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

} }

# 3)

**package** week11; **import** java.util.Iterator;

**public** **class** LinkedContainer<E> **implements** Iterable<E>{

**private** Node<E> tail; **private** Node<E> head; **private** **int** size = 0;

**public** LinkedContainer() {

}

**public** **void** add(String place) { Node<E> newNode = **new** Node<E>(); newNode.setData(place);

**if** (size == 0)

head = newNode; **else**

tail.setNext(newNode);

tail = newNode; size++;

}

**public** E get(**int** index) {

**return** findNodeByIndex(index).getData();

}

**public** **boolean** delete(**int** index) { **if** (findNodeByIndex(index) != **null** ) {

**if** (index != 0)

findNodeByIndex(index - 1).setNext(findNodeByIndex(index+1)); **else**

head = head.getNext();

size--;

**return** **true**;

}

**return** **false**;

}

**public** **int** size() { **return** size;

}

**private** Node<E> findNodeByIndex(**int** index) { **if** (index < size && index >= 0) { Node<E> curNode = head; **int** curIndex = 0; **while** (curIndex < index){ **if** (curNode.getNext() != **null**) { curNode = curNode.getNext(); curIndex++;

}

}

**return** curNode;

} **else**

**return** **null**;

}

**public** String toString() {

StringBuilder resultString = **new** StringBuilder("[");

**for** (**int** i = 0; i < size; i++) { resultString.append(get(i));

**if** (i < size-1) resultString.append(", ");

}

resultString.append("]"); **return** resultString.toString();

}

**public** **void** clear() { size = 0; tail = **null**; head = **null**;

}

**public** Iterator<E> iterator() { **return** **new** Iterator() {

@Override

**public** **boolean** hasNext() { // **TODO** Auto-generated method stub **return** **false**;

}

@Override **public** Object next() { // **TODO** Auto-generated method stub **return** **null**;

}

};

}

}

# 4)

**package** week11; **public** **class** Node<E> {

**private** Node<E> next; **private** E data;

**public** Node() {

}

**public** Node<E> getNext() {

**return** next;

}

**public** **void** setNext(Node<E> next) {

**this**.next = next;

}

**public** E getData() {

**return** data;

}

**public** **void** setData(String place) { **this**.data = (E) place;

}

}

**Результат роботи програми:**

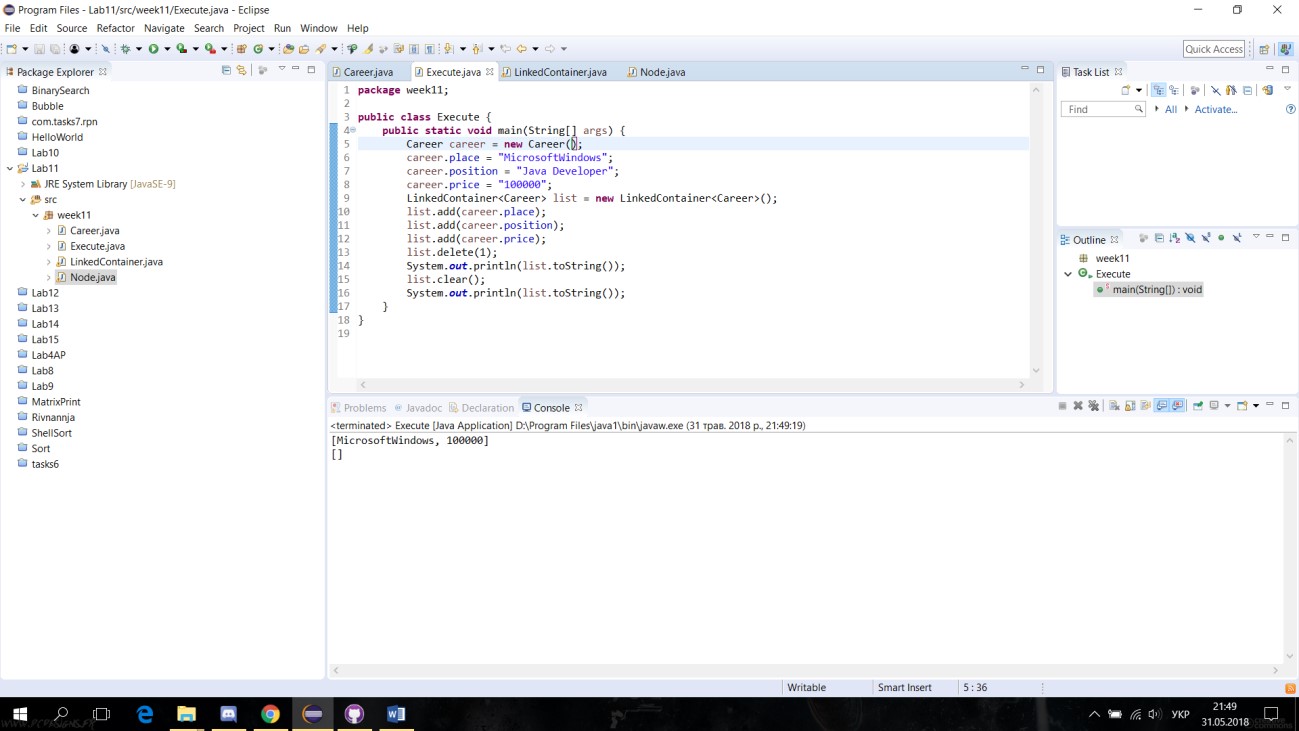


Рис.1 Результат роботи програми