Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 4

з дисципліни Алгоритми та структури даних

Варіант 7

Виконав студент групи СА-32

Мурашов О.О.

Прийняв викладач Щербак О.Д.

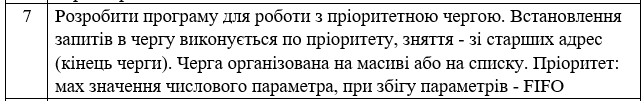
*Львів 2017*

**Тема:** Стек і черга. Хеш таблиця.

**Мета роботи:** набуття навичок моделювання зв’язаних динамічних структур даних та роботи з ними.

**Завдання на роботу:** Розробити додатки, які забезпечують запити на запис або читання даних з черги, стека або дека. Для організації вказаних структур використовувати масиви або списки. Перевірити працездатність розроблених додатків. Послідовність виконання операцій запису або читання вибираються випадково. Порівняти результати роботи, зробити висновки.

**Індивідуальне завдання:**



# Хід роботи

Текст програми на мові JAVA:

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Collectors;

public class PriorityQueue<T> {

private List<Element<T>> queue;

public PriorityQueue() {

queue = new ArrayList<>();

}

public T findMin() {

return queue.get(queue.size()-1).getValue();

}

public T findMax() {

return queue.get(0).getValue();

}

public void push(T value, int priority) {

queue.add(new Element<T>(value,priority));

reorganize();

}

public void deleteMin() {

queue.remove(queue.size()-1);

reorganize();

}

public void deleteMax() {

queue.remove(0);

reorganize();

}

public void output(){

queue.stream().forEach(x-> System.out.println(x.getValue()+"\t\tpriority: "+x.getPriority()));

}

private void reorganize() {

queue = queue.stream().sorted(((o1, o2) -> o2.getPriority().compareTo(o1.getPriority()))).

collect(Collectors.toList());

}

class Element<T> {

private T value;

private Integer priority;

public T getValue() {

return value;

}

public void setValue(T value) {

this.value = value;

}

public Integer getPriority() {

return priority;

}

public void setPriority(Integer priority) {

this.priority = priority;

}

public Element(T value, Integer priority) {

this.value = value;

this.priority = priority;

}

}

}

public class Main {

public static void main(String[] args) {

PriorityQueue<String> queue = new PriorityQueue<>();

queue.push("zzz1",5);

queue.push("zzz2",1);

queue.push("zzz3",8);

queue.push("zzz4",1);

queue.push("zzz5",5);

queue.output();

queue.deleteMin();

System.out.println("without min");

queue.output();

System.out.println("without max");

queue.deleteMax();

queue.output();

}

}

Результати комп'ютерної реалізації:

