Ministerul Educaţiei Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică.

Departamentul Informatică și Ingineria Sistemelor

Raport

la

Lucrare de laborator Nr.7

la Tehnici Avansate de Programare

**A efectuat: st. gr. IA-181 Paniș Iulian**

**A verificat: asist. univ Bumbu Tudor**

**Chișinau 2020Crearea și parcurgerea colecțiilor**

**Scopul lucrării:**

Însuşirea modalităţilor de creare, realizare şi parcurgerea a colecţiilor în Java;

**Etapele de realizare:**

1. Crearea colecţiilor şi a hărţilor;
2. Metode de realizare a colecţiilor şi a hărţilor;
3. Metode de complectare şi exstragere a obiectelor din colecţii şi hărţi;
4. Crearea interfeţii programului;
5. Prezentarea lucrării.

**Lucru individual:**

Nu utilizaţi facilităţi conexe, rearanjaţi elementele negative ale listei la sfirsit, dar cele pozitive – la inceputul listei.

**Listingul programului:**

import java.util.\*;

class Main {

//functia pentru interschimbare a elementelor

static void switchPosition(Queue<Integer> q, int size) {

// conditia se indeplineste pana cand nu se va satisface conditia

if (size <= 0)

return;

// extragerea primului element si plasarea acestuia la coada

q.add(q.peek());

q.remove();

// apelul recursiv la aceasta functie

switchPosition(q, size - 1);

}

// functia pentru inserare a elementului in lista

static void insertQueue(Queue<Integer> q, int temp, int size) {

// functia se va indeplini panacand nu se va satisface conditia

if (q.isEmpty() || size == 0) {

q.add(temp);

return;

}

//daca elementul curent este mai mare, acesta se adauga in fata

else if (temp >= q.peek()) {

q.add(temp);

switchPosition(q, size);

} else {

// adaugarea elementului la sfarsitul listei

q.add(q.peek());

q.remove();

// apelul recursiv pentru inserare in lista

insertQueue(q, temp, size - 1);

}

}

// Functia pentru sortare a listei

static void sortQueue(Queue<Integer> q) {

// functia se indeplineste pana cand nu se va indeplini conditia

if (q.isEmpty())

return;

// selectarea ultimului element

int temp = q.peek();

// eliminarea acestuia

q.remove();

// apelul recursiv

sortQueue(q);

// inserarea elementului selectat in

insertQueue(q, temp, q.size());

}

public static void main(String[] args) {

// Adaugarea elementelor in lista noua

Queue<Integer> qu = new LinkedList<>();

qu.add(-14);

qu.add(8);

qu.add(-22);

qu.add(-5);

qu.add(11);

qu.add(25);

qu.add(5);

qu.add(3);

// Sortarea listei

sortQueue(qu);

// Afisarea listei sortate

System.out.print("\nLista sortata: ");

while (!qu.isEmpty()) {

System.out.print(qu.peek() + " ");

qu.poll();

}

}

}

**Rezultatul:**

****

**Concluzii:**

În lucrărea de laborator Nr.7 am studiat domeniul ***colecțiilor*** limbajului Java*.* Am creat o listă de numere întregi pozitive și negative. Am operat cu această listă și am adăugat elemente în cadrul acesteia utilizând metodele predefinite ale claselor **Queue** și **LinkedList**.Am sortat lista în ordine descrescătoare cu ajutorul metodelor individuale create care funcționează în mod recursiv.