ГОУ ВПО – Донецкий Национальный Университет

Физико-Технический факультет

Лабораторная работа по дисциплине

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

КОЛЛЕКЦИИ В JAVA

Подготовил студент

3-уск. курса, группы ИВТ-5

Петренко Никита

**Цель**: рассмотреть основные интерфейсы, классы и алгоритмы Collection framework. Изучить существующие структуры данных.

**Файл** HeavyBox.java

public class HeavyBox implements Comparable<HeavyBox> {

private int weight;

public HeavyBox(int w) {

weight = w;

}

public void ChangeWeight(int w) {

weight = w;

}

public int mass() {

return weight;

}

public int compareTo(HeavyBox anotherBox) {

if (this.weight == anotherBox.weight) {

return 0;

} else if (this.weight < anotherBox.weight) {

return -1;

} else {

return 1;

}

}

}

**Файл** Main.java

public class main {

public static void main(String[] args) {

First();

Second();

Third();

Fourth("1, 2, 3, 4, 4, 5");

Fifth();

Sixth();

}

static void First() {

//a)

var DynMass = CreateHBoxList();

//б)

//Method 1

for(HeavyBox item : DynMass){

System.out.print(item.mass() + "\n");

}

//в)

DynMass.getFirst().ChangeWeight(1);

//г)

DynMass.remove(DynMass.getLast());

//д)

Second();

//е)

DynMass.clear();

}

static void Second() {

var DynMass = CreateHBoxList();

//Method 1

System.out.println("Method 1");

for(HeavyBox item : DynMass){

System.out.print(item.mass() + "\n");

}

//Method 2

System.out.println("Method 2");

Iterator<HeavyBox> iter = DynMass.listIterator();

while(iter.hasNext()){

System.out.println(iter.next().mass());

}

//Method 3

System.out.println("Method 3");

Stream<Integer> stream = Stream.of(DynMass.getFirst().mass());

stream.parallel().forEach(s->System.out.println(s));

}

static void Third() {

var DynMass = CreateHBoxList();

TreeSet<HeavyBox> HevBoxTree = new TreeSet<HeavyBox>(DynMass);

for(HeavyBox item : HevBoxTree){

System.out.print(item.mass() + "\n");

}

}

static void Fourth(String str) {

Set<Integer> numbersSet = new HashSet<>();

Matcher m = Pattern.compile("[0-9]+").matcher(str);

while (m.find()) {

numbersSet.add(Integer.valueOf(str.substring(m.start(), m.end())));

}

for (Integer number : numbersSet) {

System.out.print(number + " ");

}

}

static void Fifth() {

var DynMass = CreateHBoxList();

ArrayDeque<HeavyBox> Hboxes = new ArrayDeque<HeavyBox>();

DynMass.forEach(s->Hboxes.offer(s)); // using offer

DynMass.forEach(s->Hboxes.poll()); // using poll

}

static void Sixth() {

var DynMass = new LinkedList<HeavyBox>();

for(int i = 150; i<=450;i+=50)

DynMass.add(new HeavyBox(i));

sort(DynMass,300).forEach(s->System.out.println(s.mass()));

}

static LinkedList<HeavyBox> sort(LinkedList<HeavyBox> boxes, int weight) {

LinkedList<HeavyBox> resultList = new LinkedList<>();

Iterator<HeavyBox> iterator = boxes.listIterator();

while (iterator.hasNext()) {

HeavyBox currentBox = iterator.next();

if (currentBox.mass() > weight) {

resultList.add(currentBox);

iterator.remove();

}

}

return resultList;

}

static LinkedList<HeavyBox> CreateHBoxList(){

var DynMass = new LinkedList<HeavyBox>();

for(int i = 2; i<6;i++)

DynMass.add(new HeavyBox(i));

return DynMass;

}

}