|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №7-8**

Выполнил студент группы ИНБО-02-20 Судариков Д.А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Изучение работы с очередями

Изучить различные виды списков ожидания

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab7;  
  
import java.util.LinkedList;  
import java.util.Scanner;  
public class Game {  
 // Условие, определяющее, чья карта сильнее  
 public static boolean win (int first, int second) { return (first > second && first != 9 && second != 0 || first == 0 && second == 9);  
 }  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 // Игроки представлены в виде связанных списков  
 LinkedList<Integer> first = new LinkedList<>();  
 LinkedList<Integer> second = new LinkedList<>();  
 for (int i = 0; i < 5; i++)  
 first.addLast(sc.nextInt());  
 for (int i = 0; i < 5; i++)  
 second.addLast(sc.nextInt());  
 int count = 0;  
 while (!first.isEmpty() && !second.isEmpty()) {  
 if (*win*(first.get(0), second.get(0))) {  
 second.addLast(first.get(0));  
 second.addLast(second.get(0));  
 first.remove(0);  
 second.remove(0);  
 } else {  
 first.addLast(first.get(0));  
 first.addLast(second.get(0));  
 first.remove(0);  
 second.remove(0);  
 }  
 if (count++ == 105) {  
 System.*out*.println("botva");  
 return;  
 }  
 }  
 System.*out*.println((first.isEmpty() ? "first" : "second") + " " + count);  
 }  
 }

**8 практика.** В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab8;  
  
public class BoundedWaitList<E> extends WaitList<E>{  
 private int capacity;  
 BoundedWaitList(int capacity){};  
  
 public int getCapacity() {  
 return capacity;  
 }  
  
 @Override  
 public void add(E element) {  
 super.add(element);  
 }  
  
 @Override  
 public String toString() {  
 return "BoundedWaitList{" +  
 "capacity=" + capacity +  
 '}';  
 }  
}

package ru.mirea.lab.lab8;  
  
import java.util.ArrayList;  
  
public interface IWaitList<E> {  
 public void add(E element);  
 boolean remove(E element);  
 boolean contains(String E);  
 boolean containsAll(ArrayList<String> Q);  
 boolean isEmpty();  
  
}

package ru.mirea.lab.lab8;  
  
public class UnfairWaitList<E> extends WaitList<E>{  
 UnfairWaitList(){};  
  
 @Override  
 public boolean remove(E element) {  
 return super.remove(element);  
 }  
 void moveToBack(E element){};  
}

package ru.mirea.lab.lab8;  
  
import java.util.ArrayList;  
import java.util.Queue;  
import java.util.concurrent.ConcurrentLinkedQueue;  
  
public class WaitList<E> implements IWaitList<E> {  
 protected Queue<E> deque = new ConcurrentLinkedQueue<>();  
 public WaitList(){}  
 public WaitList(E element) {}  
 public void add(E element) {}  
 public boolean remove(E element) {return false;}  
 public boolean contains(String E) {return false;}  
 public boolean containsAll(ArrayList<String> Q) {return false;}  
 public boolean isEmpty(){return false;};  
}

## **Вывод**

В данной практической работе мной были получены знания по работе с очередями. Также мной были получены знания по работе со списками ожидания.