**Laboratorinis darbas Nr.1: Bazinės duomenų struktūros**

Atliko: Paulius Vizbara, IFF-7/9, 2018 10 07

[Kodas](https://github.com/PauliusVizbara/Java_laboratoriniai)

# Individuali užduotis JAVA kalba

<https://www.codingame.com/ide/puzzle/defibrillators>

**Užduotis:** Iš defibriliatorių sąrašo, pateikto ilgumom ir platumom, išrinkti esantį arčiausiai kliento.

**Kintamieji:**

TaskData taskData = FileHandler.readTaskData("data2"); // Iš failo nuskaityti užd. duomenys

List<Defibrillator> defibs = taskData.getDefibrillators(); // Visų galimų defib. duomenys

float userLatitude = taskData.getUserLatitude(); // Kliento platuma

float userLongtitude = taskData.getUserLongitude(); // Kliento ilguma

**Metodai:**

static double distanceBetweenPoints(float lat1, float long1, float lat2, float long2) // Apskaičiuoja atstumą tarp dviejų koordinačių (plokštuma/ilguma). Statinis metodas statinėj klasėje

TaskData readTaskData(String filename) // Įveda duomenis iš failo

**Pradiniai duomenys:**

3,879483

43,608177

3

1;Maison de la Prevention Sante;6 rue Maguelone 340000 Montpellier;;3,87952263361082;43,6071285339217

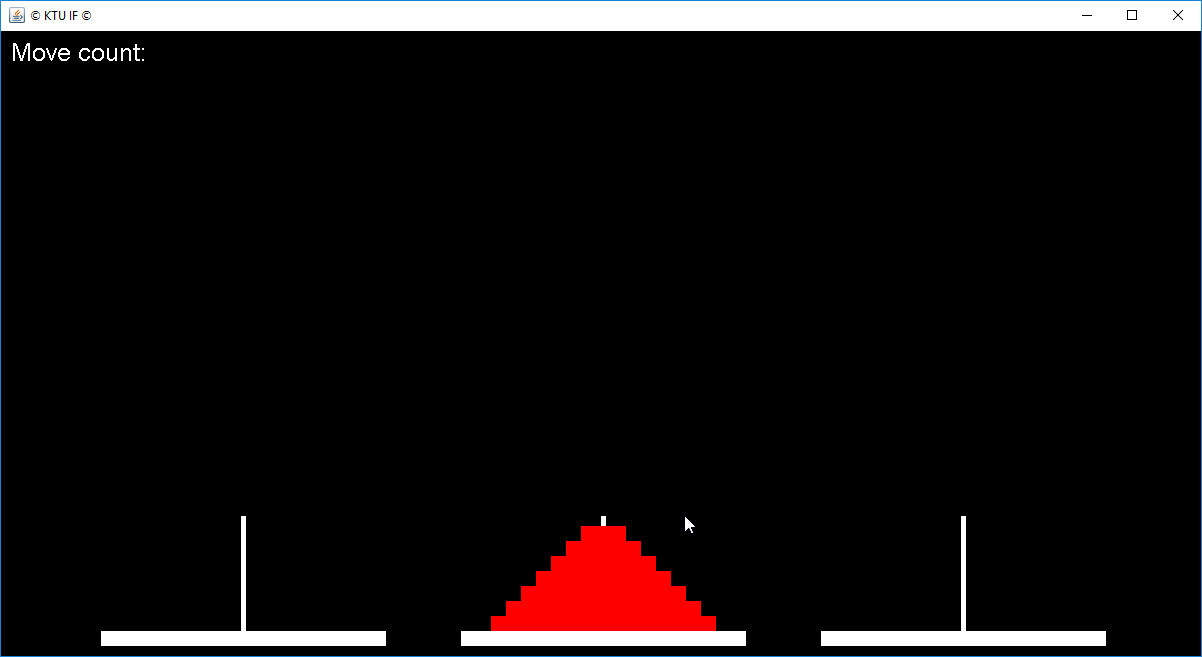
2;Hotel de Ville;1 place Georges Freche 34267 Montpellier;;3,89652239197876;43,5987299452849

3;Zoo de Lunaret;50 avenue Agropolis 34090 Mtp;;3,87388031141133;43,6395872778854

**Rezultatai:**

Maison de la Prevention Sante

# Vizualios struktūros



Žaidimas padarytas „Tower of Hanoi“ pagrindu. Plačiau apie jį galima rasti [čia.](https://en.wikipedia.org/wiki/Tower_of_Hanoi)

**Instrukcija:** visus „žiedus“ perkelti ant bet kurio kito stulpo.

# LinkedList individualios užduotys ir greitaveika

**LinkedList individualiai realizuoti metodai:**

### public E removeFirst() {

if (size == 0) {

return null;

}

E removedElement = null;

removedElement = first.element;

if ( size == 1) first = null;

else first = first.next;

return removedElement;

}

### public boolean containsAll(E... e){

for(E temp:e){

if(!contains(temp) ) return false;

}

return true;

}

### public boolean contains(E e){

for (start(); !lastElement(); next()) {

if ( current.element.compareTo(e) == 0) return true;

}

return false;

}

### public E set(int k, E e) {

int counter = 0;

for (start(); !lastElement(); next()) {

if (counter == k) {

E returnedElement = current.element;

current.element = e;

return returnedElement;

}

counter++;

}

return null;

}

A public E remove(int index) {

int counter = 0;

E removedElement = null;

if (index == 0) {

removedElement = first.element;

first = first.next;

}

for (start(); !lastElement(); next()) {

if (counter + 1 == index) {

if (index == size) {

current.next.element = removedElement;

current.next = null;

} else {

removedElement = current.next.element;

current.next = current.next.next;

}

return removedElement;

}

counter++;

}

return null;

}

public boolean add(E e) {

if (e == null) {

return false; // nuliniai objektai nepriimami

}

if (first == null) {

first = new Node<>(e, first);

last = first;

} else {

Node<E> e1 = new Node(e, null);

last.next = e1;

last = e1;

}

size++;

return true;

}

## Greitaveikos testavimas

Reikėjo ištestuoti:

* sin VS sqrt ( sinusas VS kvadratinė šaknis)
* elemento įdėjimas į sąrašo vidurį LinkedList ir ArrayList sąrašuose.

## Sinusas VS Kvadratinė šaknis

1| kiekis(\*k) Saknis Sinusas

2| 2000( 1) 0.0002 0.0009

3| 4000( 2) 0.0002 0.0009

4| 8000( 4) 0.0002 0.0009

5| 16000( 8) 0.0002 0.0006

Bendras tyrimo laikas 0.111 sekundžių

Išmatuotas tyrimo laikas 0.004 sekundžių

t.y. 96.3% sudaro pagalbiniai darbai

6|

7| Normalizuota (santykinė) laikų lentelė

8| kiekis(\*k) Saknis Sinusas

9| 2000( 1) 1.00 3.98

10| 4000( 2) 0.66 3.89

11| 8000( 4) 0.64 3.94

12| 16000( 8) 0.69 2.46

## ArrayList VS LinkedList

1| kiekis(\*k) Array Linked

2| 2000( 1) 0.0004 0.0011

3| 4000( 2) 0.0002 0.0018

4| 8000( 4) 0.0003 0.0016

5| 16000( 8) 0.0004 0.0026

Bendras tyrimo laikas 0.174 sekundžių

Išmatuotas tyrimo laikas 0.008 sekundžių

t.y. 95.2% sudaro pagalbiniai darbai

6|

7| Normalizuota (santykinė) laikų lentelė

8| kiekis(\*k) Array Linked

9| 2000( 1) 1.00 2.66

10| 4000( 2) 0.54 4.51

11| 8000( 4) 0.75 4.00

12| 16000( 8) 0.92 6.33

## Išvados:

Programavimas JAVA kažkuo išskirtinai nesiskiria, nes programavom Object Oriented paradigmu nuo pirmo kurso. Šaunu, kad laboratoriniame galima laisvai pasirinkti savo temą, taip yra daug didesnė motyvacija kurti tai, ką nori.