KAUNO TECHNOLOGIJOS UNIVERSITETAS

Duomenų struktūros (P175B014) Laboratorinio darbo Nr.1 ataskaita

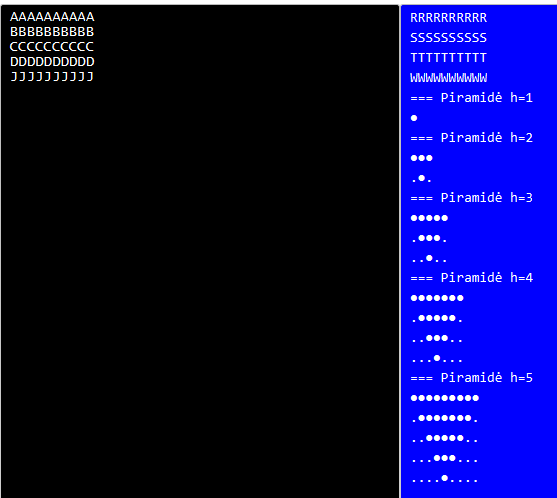
Atliko Arnas Švenčionis gr. IFF-8/11

Lab1A. Įvado dalis – JAVA FX

sudarykite naują metodą funkcijų sin, cos ir tan skaičiavimui, kai duotas kampas yra nurodomas laipsniais:



sudarykite pasirinktinai metodus apverstai ar pasuktai piramidei spausdinti:



pakeiskite šabloną, kad jis atpažintų:

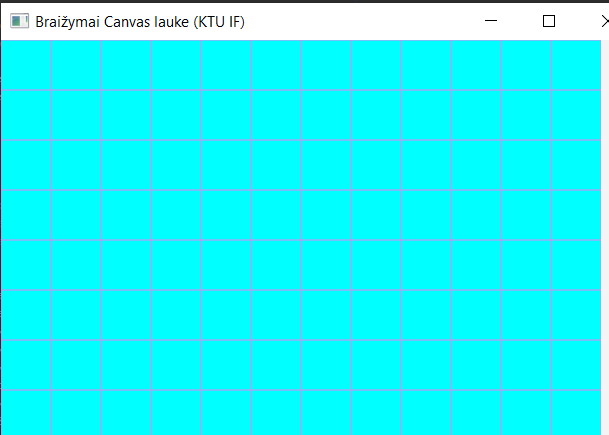
minusą

žodžius su lotyniškomis raidėmis

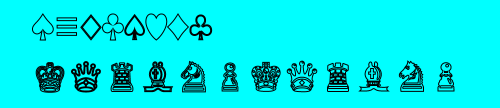
žodžius su lietuviškomis raidėmis

informaciją tarp skliaustų ()

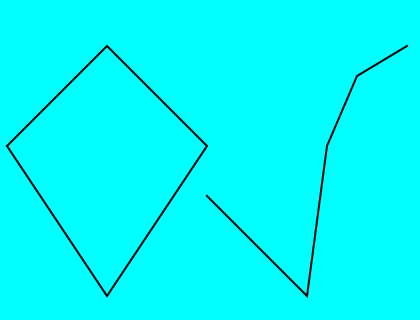
plonomis linijomis su žingsniu step=50 nubrėžkite tinklelį



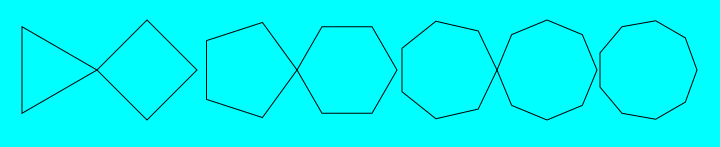
išbandykite kitus simbolius:



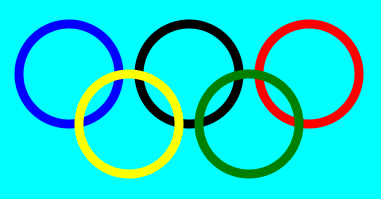
nubrėžkite polilinijas ir poligonus:



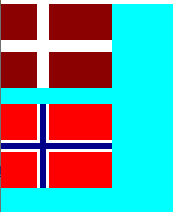
nubrėžkite taisyklingus 3, 4, 5, ..., 9-kampius (parašykite funkciją, kuri paskaičiuoja skaičių masyvus kuriuose surašomos taisyklingo daugiakampio koordinatės)



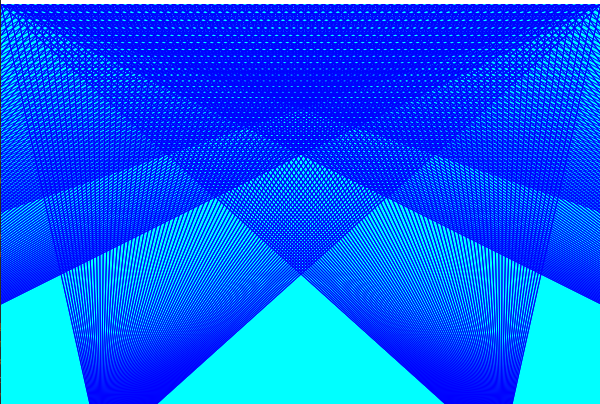
nubrėžkite žiedus (Olimpiniai simboliai):



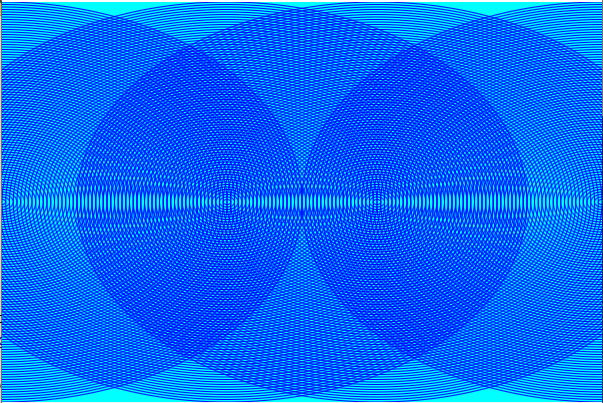
pasirinktinai nubrėžkite kelių valstybių sudėtingesnes vėliavas:



sukurkite metodą, kuris linijų pluoštus skleistų iš įvairių kampų:

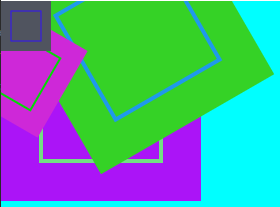


sukurkite metodą, kur apskritimų centrai būtų keturiuose taškuose:



sukurkite metodą, kur rekursijos metu gaunami ovalai būtų nukreipti į apačią:

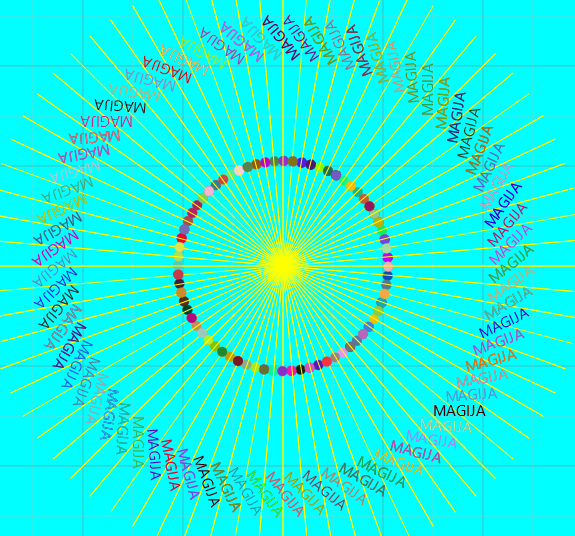
sukurkite savo figūrų rinkinį, kurį transformuosite:



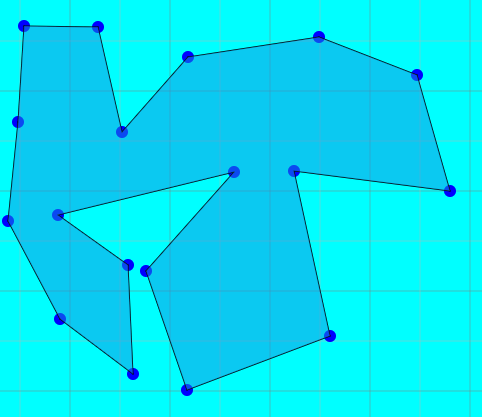
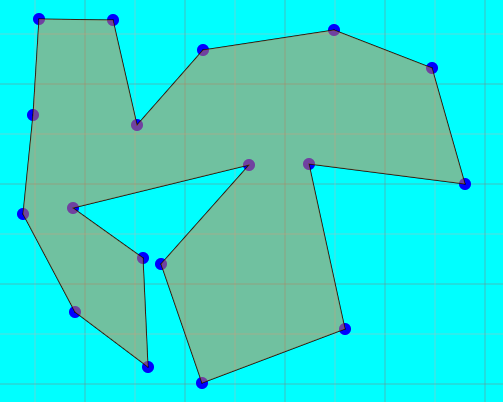
gautus bėžinius sudėkite į laboratorinio ataskaitą:



išbandykite su savo baziniais rinkiniais:



išbandykite pakartotinius figūrų paspaudimus, kokia reakcija?:

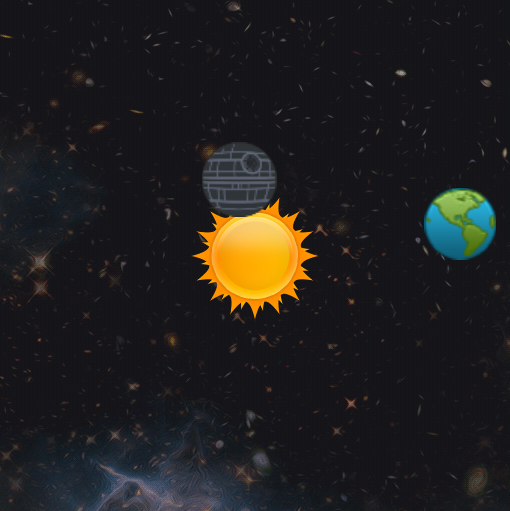
Keiciasi spalva.

paaiškinkite kintamojo t skaičiavime naudojamą konstantą:

t apibudina žemės sukimosi aplink saulę greitį.

paskaičiuokite, per kiek laiko žemė apskrieja aplink saulę:

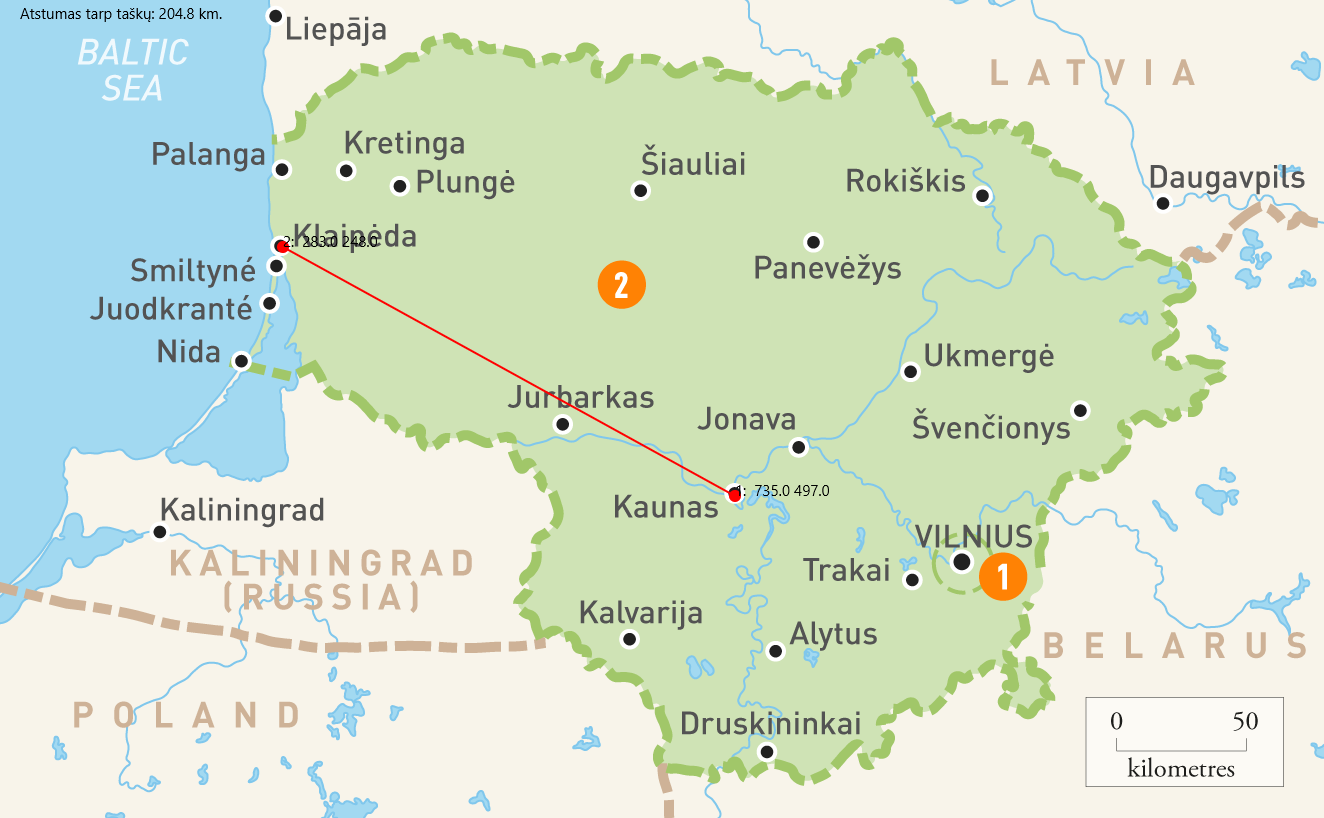
per 6 sekundes  
papildykite erdvę kitais kosminiais ar fantastiniais kūnais:

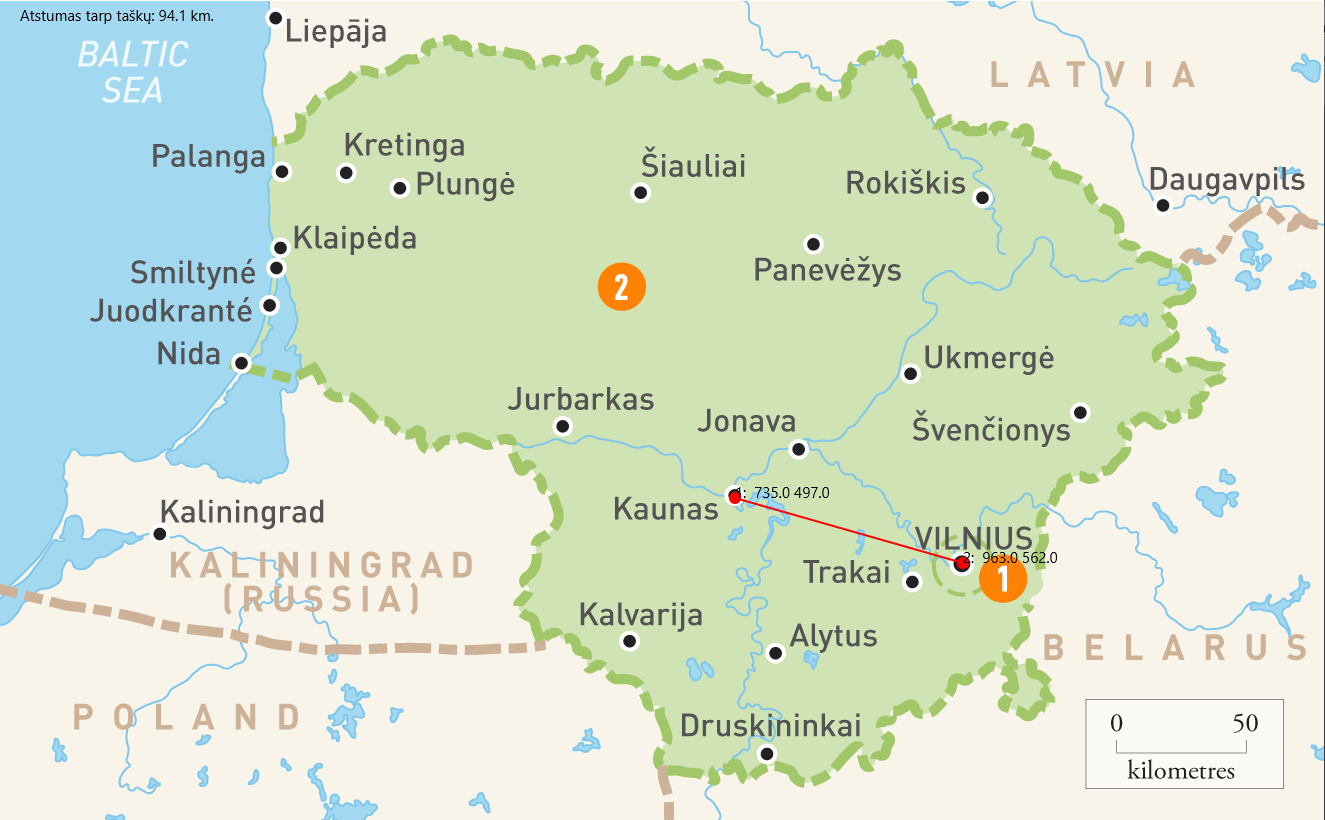


**Papildomi taškai:**

Digitalizavimo uždavinys: paruoškite žemėlapio iškarpą .png ar .jpg formatu. Išmatuokite etaloninį atstumą tarp 2 taškų. Pelės paspaudimu sukurkite poliliniją ir paskaičiuokite jos atstumą ir paverskite į realų atstumą (papildomi 3 balai).







Lab1B. Susieti bendriniai sąrašai

**Pradiniai duomenys:**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Lausogis 25 1512 125125,0 8500**

**8| Midrakadis 74 12516 20569,8 8500**

**Tiriamas metodas – LinkedList.add(E e)**

**public boolean add(int k, E e) {**

**if (k < 0 || k > this.size) {**

**return false;**

**}**

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

**if (k == 0) {**

**temp.next = first;**

**first = temp;**

**} else if (k == this.size) {**

**last.next = temp;**

**last = temp;**

**}**

**else {**

**Node<E> nn = first;**

**for (int i = 0; i < k - 1; i++) {**

**nn = nn.next;**

**}**

**temp.next = nn.next;**

**nn.next = temp;**

**}**

**size++;**

**return true;**

**}**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Lausogis 25 1512 125125,0 8500**

**8| Midrakadis 74 12516 20569,8 8500**

**9| Puikioji 255 20021 20212,1 3500**

**Žvaigždė “Puikioji” buvo pridėta į sąrašo gala.**

**Tiriamas metodas – LinkedList.remove(int index);**

**@Override**

**public E remove(int k) {**

**if (first == null) {**

**return null;**

**}**

**if (k < 0 || k >= size) {**

**return null;**

**}**

**if (k == 0) {**

**if (first == last) {**

**first = last = null;**

**size = 0;**

**} else {**

**first = first.next;**

**size--;**

**return first.element;**

**}**

**} else if (k == size - 1) {**

**Node<E> temp = first;**

**for (int i = 0; i < k - 1; i++) {**

**temp = temp.next;**

**}**

**temp.next = null;**

**last = temp;**

**size--;**

**return first.element;**

**} else {**

**Node<E> temp = first;**

**for (int i = 0; i < k - 1; i++) {**

**temp = temp.next;**

**}**

**temp.next = temp.next.next;**

**size--;**

**return temp.next.element;**

**}**

**return null;**

**}**

**1| Zvaigziu 45 1251 2523,1 3500**

**2| Lapignas 25 1512 125125,0 8500**

**3| Salamina 74 12516 20569,8 8500**

**4| Naujorita 13 3021 5011,1 1700**

**5| Akaulosjus 45 1251 2523,1 3500**

**6| Lausogis 25 1512 125125,0 8500**

**7| Midrakadis 74 12516 20569,8 8500**

**Ištrintas nulinis objektas;**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Midrakadis 74 12516 20569,8 8500**

**Ištrintas šeštas objektas;**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Lausogis 25 1512 125125,0 8500**

**Ištrintas septintas objektas;**

**Tiriamas metodas – LinkedList.set(int index, E e);**

**@Override**

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

**if(k < 0 || k >= size)**

**return null;**

**if(first == null)**

**return null;**

**if(k == 0){**

**first.element = e;**

**return first.element;**

**}**

**Node<E> node = first;**

**while (--k > 0){**

**node = node.next;**

**}**

**node.next.element = e;**

**return node.next.element;**

**}**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Puikioji 255 20021 20212,1 3500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Lausogis 25 1512 125125,0 8500**

**8| Midrakadis 74 12516 20569,8 8500**

**Antras objektas buvo pakeistas į „Puikioji“**

**1| Puikioji 255 20021 20212,1 3500**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Akaulosjus 45 1251 2523,1 3500**

**7| Lausogis 25 1512 125125,0 8500**

**8| Midrakadis 74 12516 20569,8 8500**

**Nulinis objektas nuvo pakeistas į „Puikioji“.**

**Tiriamas metodas – addAll(int index, LinkedList<? Extends E> c):**

**public boolean addAll(int index, LinkedList<? extends E> c){**

**if(this.first == null) return false;**

**else if(index < 0 || index > size) return false;**

**else if(c.size() == 0) return false;**

**for (int i = 0; i < c.size(); i++) {**

**this.add(index + i, c.get(i));**

**}**

**return true;**

**}**

**Testavimui sukursime dar vieną sarašą, kurį pridėsime prie senojo:**

**1| Puikioji 255 20021 20212,1 3500**

**2| Loyalitas 212 12155 20000,2 8500**

**3| Kazanopolis 1003 6541 9125,3 8500**

**Rezultatas pridėjus sąrašą, indexui esant 0:**

**1| Puikioji 255 20021 20212,1 3500**

**2| Loyalitas 212 12155 20000,2 8500**

**3| Kazanopolis 1003 6541 9125,3 8500**

**4| Malarija 13 3021 505111,1 1700**

**5| Zvaigziu 45 1251 2523,1 3500**

**6| Lapignas 25 1512 125125,0 8500**

**7| Salamina 74 12516 20569,8 8500**

**8| Naujorita 13 3021 5011,1 1700**

**9| Akaulosjus 45 1251 2523,1 3500**

**10| Lausogis 25 1512 125125,0 8500**

**11| Midrakadis 74 12516 20569,8 8500**

**Rezultatas pridėjus sąrašą, indexui esant 5:**

**1| Malarija 13 3021 505111,1 1700**

**2| Zvaigziu 45 1251 2523,1 3500**

**3| Lapignas 25 1512 125125,0 8500**

**4| Salamina 74 12516 20569,8 8500**

**5| Naujorita 13 3021 5011,1 1700**

**6| Puikioji 255 20021 20212,1 3500**

**7| Loyalitas 212 12155 20000,2 8500**

**8| Kazanopolis 1003 6541 9125,3 8500**

**9| Akaulosjus 45 1251 2523,1 3500**

**Tiriamas metodas – LinkedList.removeFirst():**

**public E removeFirst(){**

**if(first == null) return null;**

**Node<E> temp = first;**

**if(size == 1){**

**first = null;**

**last = null;**

**size = 0;**

**}**

**else{**

**first = first.next;**

**size--;**

**}**

**return temp.element;**

**}**

**1| Zvaigziu 45 1251 2523,1 3500**

**2| Lapignas 25 1512 125125,0 8500**

**3| Salamina 74 12516 20569,8 8500**

**4| Naujorita 13 3021 5011,1 1700**

**5| Akaulosjus 45 1251 2523,1 3500**

**6| Lausogis 25 1512 125125,0 8500**

**7| Midrakadis 74 12516 20569,8 8500**

**Tiriamas objektas – LinkedList.RemoveFirstOccurrence(E e):**

**public boolean RemoveFirstOccurrence(E e) {**

**if (first == null) {**

**return false;**

**}**

**if (first.element.equals(e) && last.element.equals(e)) {**

**first = last = null;**

**size = 0;**

**return true;**

**}**

**if (first.element.equals(e)) {**

**first = first.next;**

**size--;**

**return true;**

**}**

**for (Node<E> temp = first; temp.next != null; temp = temp.next) {**

**if (temp.next.element.equals(e)) {**

**if (temp.next == last) {**

**temp.next = null;**

**last = temp;**

**} else {**

**temp.next = temp.next.next;**

**}**

**size--;**

**return true;**

**}**

**}**

**return false;**

**}**

**Rezultatas, gautas į metodo parametrą padavus antrą Star objektą:**

**1| Malarija 13 3021 505111,1 1700**

**2| Lapignas 25 1512 125125,0 8500**

**3| Salamina 74 12516 20569,8 8500**

**4| Naujorita 13 3021 5011,1 1700**

**5| Akaulosjus 45 1251 2523,1 3500**

**6| Lausogis 25 1512 125125,0 8500**

**7| Midrakadis 74 12516 20569,8 8500**

Greitaveikos tyrimas

Reikia palyginti metodus Math.pow(x, 1.0/3) ir Math.cbrt(x). Testavimui sugeneruosiu įvairių dydžių int sąrašus, kad galėtume matyti sirtumą aiškiau.

Reikia palyginti ArrayList<Integer> metodus indexOf(Object o) ir lastIndexOf(Object o). Tam sugeneruosiu įvairių dydžių sąrašus ir patestuosiu greitaveiką su kiekvienu.

Rezultatai:

1| memTotal= 257425408

2| memMax= 477626368

3| 0 - Elementų kiekis ArrayList<Integer> sąraše

4| 1 - Pasiruošimas tyrimui - x sukurimas

5| 2 - Math.pow(x, 1.0/3)

6| 3 - Math.cbrt(x)

7| 4 - ArrayList<Integer> sugeneravimas

8| 5 - indexOf(Object o)

9| 6 - lastIndexOf(Object o)

0 1 2 3 4 5 6

10000 0,00001 0,00199 0,00407 0,00261 0,00062 0,00048

20000 0,00000 0,00336 0,00345 0,00354 0,00079 0,00077

40000 0,00000 0,00397 0,00531 0,00510 0,00174 0,00160

80000 0,00001 0,00101 0,00926 0,00377 0,00084 0,00083

Iš greitaveikos rezultatų galime matyti, jog metodas Math.pow(x, 1.0/3) yra greitesnis už Math.cbrt(x) (2 ir 3 stulpeliai). Nors skirtumas nėra didelis. Metodai indexOf(Object o) ir lastIndexOf(Object o) yra labai panašūs savo veikimo greičiu (5 ir 6 stulpeliai).

Išvados

Darant laboratorinį darbą, sudėtingiausia buvo priprasti prie naujos, bet turinčios panašumų su C# kalba. Turbūt sunkiausia laboratorinio darbo dalis buvo A, nes reikėjo priprasti prie naujos programavimo aplinkos, ir javos skirtumų nuo C#. B dalis pasirodė lengvesnė, nes vienkrypčius sąrašus darėme praeitą pusmetį. Dauguma dalykų reikėjo tiesiog prisimint. Sužinojau, kad Javos kalboje yra daug lengviau programuoti vizualiai – kurti įvairias figūras ir linijas. „Žaidimuko“ kūrimas buvo labai įdomi darbo dalis, nes galėjau pamatyti kur šias funkcijas galima pritaikyti realiame gyvenime. Greitaveikos užduotis taip pat labai patiko. Buvo įdomu lyginti metodus pagal jų veikimo greitį.