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

Atliko: Tadas Laurinaitis, IFF 6/8, 2017 12 19

# 1.1 Darbo užduotis

*Duota*: Projektas **Lab4\_MaisosLenteles**, kuriame yra pateiktos toliau naudojamos sisteminės klasės, grafinės Swing (paketas laborai.gui.swing) ir JavaFX (paketas laborai.gui.fx) sąsajos ir Atvaizdžio realizacija maišos lentele , kolizijas sprendžianti atskirų grandinėlių metodu (klasės **MapKTU** ir **MapKTUx**).

*Atlikti*: Perkelkite reikiamus paketus ir klases iš ankstesnių laboratorinių darbų ir atlikite reikalingas užduotis, susijusias su maišos lentelių kūrimu bei panaudojimu.

*Tyrimo ir analizės dalis:*

1. Išnagrinėkite klasės **MapKTU** metodus, ištirkite jų panaudojimą operacijų atlikimui su atvaizdžiu.
2. Atlikite maišos lentelės greitaveikos tyrimo eksperimentus su faile zodynas.txt surašytais žodžiais.

*Individuali programavimo dalis:*

1. Pakete **lab4pavarde** sudarykite individualių elementų atvaizdžių panaudojimo klasę, kurioje būtų atvaizdžio formavimas, poros šalinimas atvaizdyje, raktą atitinkančios reikšmės paieška ir pan., panaudojant klasę **MapKTU** (3 metodai). Sukurtų metodų veikimą demonstruokite pateiktuose Swing arba JavaFX dialoguose arba sukurkite nuosavą, pasinaudodami paskaitų medžiaga.
2. Klasėje **MapKTU** realizuokite metodą ***boolean containsValue(Value value)***, patikrinantį ar atvaizdyje egzistuoja duota reikšmė, ir metodus, skaičiuojančius maišos lentelės charakteristikas: vidutinį grandinėlės ilgį ir maišos lentelės masyvo tuščių elementų kiekį.
3. Pakete **lab4pavardė** sukurkite **MapADT** interfeisą maišos lentele realizuojančią klasę **MapKTUOA**, kolizijas sprendžiant vienu atviros adresacijos metodu: tiesinio dėstymo, kvadratinio dėstymo ar dvigubos maišos.
4. Ištirkite ir su **MapKTU** realizacija palyginkite 4 punkte sukurtos **MapADT** realizacijos metodų greitaveiką, sudarykite vykdymo laikų grafikus ir atlikite rezultatų analizę. Tai atlikite su Jūsų individualios klasės atvaizdžių poromis ir su faile zodynas.txt esančiais žodžiais (tyrimui užtenka, kad raktas ir reikšmė būtų tas pats žodis).
5. Realizuoti individualiai nurodytus metodus **MapKTU** klasėje.

# 1.2 Praktinio darbo tyrimas ir analizė

Pateikiami **studento sukurtų metodų** eksperimentai (viso 3-5 variantai)

|  |
| --- |
| ***Tiriamas metodas –* MapKTU.** containValue(V value) |
| public boolean containsValue(V value){  for(Node<K, V> n1 : this.table){  for(Node<K, V> n2 = n1; n2 != null; n2 = n2.next){  if(n2.value.equals(value)){  return true;  }  }  }  return false;  } |
| **Gauti rezultatai** |
| true |
| **Komentarai:**  Patikrinama ar šioje lentelėje yra elementas turintis value reikšmę |
| ***Tiriamas metodas --* MapKTU.** numberOfEmpties() |
| public int numberOfEmpties(){  int count = 0;    for(Node<K, V> n : this.table){  if(n == null){  count++;  }  else  continue;  }  return count;  } |
| **Gauti rezultatai** |
| ===== metodas6 ... |

|  |
| --- |
| 15 |
| **Komentarai:**  Surandamas tuščių lentelės elementų skaičius. |
| ***Tiriamas metodas --* MapKTU.** replace(K key, V value) |
| public V replace(K key, V value){  if(this.get(key) != null){  this.put(key, value);  return null;  }  return null;  } |
| **Gauti rezultatai** |
|  |
| **Komentarai:**  Surandamas elementas turintis raktą key, o jo reikšmė pakeičiama i value. |

# 1.3 Klasės panaudotos atvaizdžio gavimui

**public class Kate** // Pagal šią klasę kuriamas Kates objektas

**Laukai:**

private String vardas;

private String veisle;

private String lytis;

private double amzius;

private double kaina; private int assists;

**Metodai:**

public String getName() //gražinamas Player vardas

public String getTeam() //gražinama Player komanda

public String getPosition() //gražinama Player pozicija

public String getChampion() //gražinamas Player čempionas

public int getKills() //gražinamas Player kill‘ų skaičius

public int getAssists() //gražinamas Player assist‘ų skaičius

**Metodai:**

public Kate create(String dataString)

public final void parse(String dataString)

public String validate()

public String toString()

public String getVardas()

public String getLytis()

public String getVeisle()

public double getAmzius()

public double getKaina()

**public class KaciuTestai**// Šioje klasėje atliekami veiksmai su katemis

**Metodai:**

public static void main(String[] args)

public static void atvaizdzioTestas()

**public class KaciuGamyba**// Šioje klasėje generuojami kaciu objektai ir ju duomenys

**Laukai:**

private Kate[] kates;

private String[] raktai;

private int kiekis = 0, idKiekis = 0;

**Metodai:**

public static Kate[] gamintiKates(int kiekis)

public static String[] gamintiKaciuIds(int kiekis)

public Kate[] gamintiIrParduotiKates(int aibesDydis, int aibesImtis)

public Kate parduotiKate()

public String gautiIsBazesKatesId()

**Gauti rezultatai:**

========Porų išsidėstymas atvaizdyje pagal raktus=======

[ 0 ]

[ 1 ]

[ 2 ]

[ 3 ]

[ 4 ]

[ 5 ] --> TA178 --> TA156

[ 6 ] --> TA102

[ 7 ]

[ 8 ]

[ 9 ] --> TA105

[ 10 ] --> TA106

[ 11 ] --> TA107

[ 12 ]

[ 13 ]

[ 14 ] --> TA171

[ 15 ]

\*\*\*\*\*\* Bendras porų kiekis yra 7

======== Atvaizdžio pabaiga =======

1| Ar egzistuoja pora atvaizdyje?

2| true

3| false

4| Pašalinamos poros iš atvaizdžio:

5| Mole\_Sfinksas\_Mot: 5.0 2500.0

6| null

========Porų išsidėstymas atvaizdyje pagal raktus=======

[ 0 ]

[ 1 ]

[ 2 ]

[ 3 ]

[ 4 ]

[ 5 ] --> TA178 --> TA156

[ 6 ]

[ 7 ]

[ 8 ]

[ 9 ] --> TA105

[ 10 ] --> TA106

[ 11 ] --> TA107

[ 12 ]

[ 13 ]

[ 14 ] --> TA171

[ 15 ]

\*\*\*\*\*\* Bendras porų kiekis yra 6

======== Atvaizdžio pabaiga =======

7| Atliekame porų paiešką atvaizdyje:

8| Juode\_Abisinija\_Mot: 6.0 7800.0

9| null

10| Išspausdiname atvaizdžio poras String eilute:

TA178=Juode\_Abisinija\_Mot: 6.0 7800.0

TA156=Melga\_Abisinija\_Mot: 4.0 500.0

TA105=Selme\_Sfinksas\_Mot: 4.0 27672.0

TA106=Julia\_Abisinija\_Mot: 4.0 500.0

TA107=Mole\_Sfinksas\_Mot: 5.0 2500.0

TA171=Prisma\_Sfinksas\_Mot: 10.0 3600.0

**1.4 Greitaveikos Tyrimas ir rezultatai:**

Laboratorinio darbo užduotyje buvo duoti du tyrimai, kuriuos reikėjo atlikti: 1) Palyginti klasių TreeSet<Integer> ir HashSet<Integer> metodus remove(Object o); 2) ) Palyginti klasių TreeSet<Integer> ir HashSet<Integer> metodus containsAll(Collection<?> c);

**Greitaveikos tyrimo rezultatai:**

Greitaveikos tyrimas:

kiekis(\*k) put put

10000( 1) 0.0042 0.0007

20000( 2) 0.0070 0.0028

40000( 4) 0.0079 0.0023

80000( 8) 0.0188 0.0045

Bendras tyrimo laikas 1.508 sekundžių

Išmatuotas tyrimo laikas 0.048 sekundžių

t.y. 96.8% sudaro pagalbiniai darbai

Normalizuota (santykinė) laikų lentelė

kiekis(\*k) put put

10000( 1) 1.00 0.18

20000( 2) 1.65 0.66

40000( 4) 1.88 0.55

80000( 8) 4.46 1.06

# Išvados

Šio laboratorinio darbo metu išsiaiškinome dar vieną naudingą duomenų struktūrą – hash lenteles, bei jos realizaciją ir panaudojimą.

Bendros laboratorinio darbo atlikimo laiko sąnaudos buvo 6 valandų.