

## Laboratorinis darbas Nr.1: Bazinės duomenų struktūros

Atliko: Tadas Laurinaitis, IFF 6/8, 2017 12 10

### 1.1 Darbo užduotis

Pasinaudojant duotu projektu **Lab1\_BazinesStrukturos**, kuriame yra pateiktos bazinių struktūrų demonstracinės klasės su valdymo perjungimo metodais, reikia atlikti metodų tyrimus ir paaiškinti gautus rezultatus. Išbandyti sistemos NetBeans teikiamas galimybes.

### 1.2 Praktinio darbo tyrimas ir analizė

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

<i>Tiriamas metodas – E_Komanda.kiekŽaidėjųPelnėTaškus()</i>
<pre> int kiekŽaidėjųPelnėTaškus() {     // surašykite reikiamus veiksmus     int count = 0;     for(int al: taškai){         if(al != 0){             count++;         }     }     return count; } </pre>
Gauti rezultatai
<pre> ***** Komandos Sakalai ataskaita Komandos žaidėjų skaičius 12 Surinkti taškai [5, 2, 12, 21, 26, 0, 0, 7, 2, 11, 17, 0] Taškų suma Senu Stiliumi 103 Taškų suma ForEach Stiliumi 103 Max taškų Senu Stiliumi 26 Max taškų ForEach Stiliumi 26  Zaidejai Pelne Taskus: 9 Vidurkis: 8.0 ***** Komandos Ereliai ataskaita Komandos žaidėjų skaičius 8 </pre>

```

Surinkti taškai [44, 8,
0, 21, 26, 0, 0, 7]
Taškų suma Senu Stiliumi
106
Taškų suma ForEach
Stiliumi 106
Max taškų Senu Stiliumi
44
Max taškų ForEach
Stiliumi 44

Zaidejai Pelne Taskus: 5
Vidurkis: 13.0
Rungtynes laimėjo Ereliai

```

**Komentarai:**

Surandamas žaidėjų, pelniusių taškus, kiekis

***Tiriamas metodas -- E\_Komanda.koksVidurkis()***

```

double koksVidurkis(){
    // surašykite reikiamus veiksmus
    double vid = 0;
    int suma = 0;
    int count = 0;
    for(int al: taškai){
        count++;
        suma = suma + al;
    }
    vid = suma/count;
    return vid;
}

```

**Gauti rezultatai**

===== metodas6 ...

```

***** Komandos Sakalai ataskaita
Komandos žaidėjų skaičius 12
Surinkti taškai [5, 2, 12, 21, 26, 0, 0, 7, 2, 11, 17, 0]
Taškų suma Senu Stiliumi    103
Taškų suma ForEach Stiliumi 103
Max taškų Senu Stiliumi     26
Max taškų ForEach Stiliumi  26

Zaidejai Pelne Taskus: 9
Vidurkis: 8.0
***** Komandos Ereliai ataskaita
Komandos žaidėjų skaičius 8
Surinkti taškai [44, 8, 0, 21, 26, 0, 0, 7]
Taškų suma Senu Stiliumi    106
Taškų suma ForEach Stiliumi 106
Max taškų Senu Stiliumi     44
Max taškų ForEach Stiliumi  44

Zaidejai Pelne Taskus: 5
Vidurkis: 13.0
Rungtynes laimėjo Ereliai

```

**Komentarai:**

Surandamas įmestų taškų vidurkis

***Tiriamas metodas -- E\_Komanda.kasLaimejo(E\_Komanda kita)***

```
String kasLaimėjo(E_Komanda kita){
    // surašykite reikiamus veiksmus, kad sužinotume laimėjusią komandą
    // numatykite lygiųjų atvejį
    if(sumaForEachStiliumi() > kita.sumaForEachStiliumi()){
        return pavadinimas;
    }
    else
        return kita.pavadinimas;
}
```

**Gauti rezultatai**

```
***** Komandos
Sakalai ataskaita
Komandos žaidėjų
skaičius 12
Surinkti taškai
[5, 2, 12, 21, 26,
0, 0, 7, 2, 11,
17, 0]
Taškų suma Senu
Stiliumi 103
Taškų suma ForEach
Stiliumi 103
Max taškų Senu
Stiliumi 26
Max taškų ForEach
Stiliumi 26

Zaidejai Pelne
Taskus: 9
Vidurkis: 8.0
***** Komandos
Ereliai ataskaita
Komandos žaidėjų
skaičius 8
Surinkti taškai
[44, 8, 0, 21, 26,
0, 0, 7]
Taškų suma Senu
Stiliumi 106
Taškų suma ForEach
Stiliumi 106
Max taškų Senu
Stiliumi 44
Max taškų ForEach
Stiliumi 44

Zaidejai Pelne
Taskus: 5
Vidurkis: 13.0
Rungtynes laimėjo Ereliai
```

**Komentarai:**

Surandama komanda laimėjusi rungtynes

### 1.3 Pirmo kurso laboratorinio užduoties programa JAVA kalba

**Užduotis:** 19. WCG turnyras. Kaune vyksta atrankinis kompiuterinio žaidimo „League of Legends“ turnyras. Žaidime dvi penkių žaidėjų komandos kovoja tarpusavyje valdydamos skirtingus čempionus, siekdamos sunaikinti priešininkų bazės gilumoje stovintį „Nexus“. Duomenų faile pateikta informacija apie pirmo rato dalyvius ir jų rezultatus: vardas, pavardė, komanda, pozicija, čempionas, sunaikinimai(K), dalyvavimai sunaikinimuose(A).

- Raskite aktyviausią „Jungle“ pozicijoje žaidusį žaidėją. Žaidėjo aktyvumui palyginti naudokite nužudymų ir dalyvavimų nužudymuose sumą (K+A).
- Raskite geriausiai bendradarbiavusią komandą (didžiausia bendra dalyvavimų nužudymuose suma), komandos pavadinimą atspausdinkite ekrane.
- Sudarykite panaudotų čempionų sąrašą, čempionų pavadinimus surašykite į failą „Čempionai.csv“.

**Sprendimas:** aprašomos sudarytos klasės ir metodai -> *pilni kodai pateikiami kartu su projektu.*

**public class Player** // Pagal šią klasę kuriamas Player objektas

**Laukai:**

```
private String name;  
private String team;  
private String position;  
private String champion;  
private int kills;  
private int assists;
```

**Metodai:**

```
public String getName() //grąžinamas Player vardas  
public String getTeam() //grąžinama Player komanda  
public String getPosition() //grąžinama Player pozicija  
public String getChampion() //grąžinamas Player čempionas  
public int getKills() //grąžinamas Player kill'ų skaičius  
public int getAssists() //grąžinamas Player assist'ų skaičius
```

**public class Lab1Laurinaitis**// Šioje klasėje atliekami visi veiksmai ir spausdinami rezultatai

**Metodai:**

```
public static void main(String[] args) //Pagrindinis metodas  
  
public static Player[] getPlayers()//Sudaro zaideju sarasa  
  
static Player[] getJunglers(Player[] players)//Gauna visus junglerius is players masyvo  
  
static Player mostActiveJungler(Player[] junglers) //Randa geriausia jungle pozicijos zaideja  
  
static String[] championsList(Player[] players) //Championu sarasas  
  
static String bestTeam(Player[] players) //Randa geriausia komanda
```

**Pradinis duomenų rinkinys:**

Sudaromas metodo getPlayers() pagalba

**Gauti rezultatai:**

Aktyviausias jungle pozicijos zaidejas: 5

Championu Sarasas:

Champion No. 1

Champion No. 2

Champion No. 3

Champion No. 4

Champion No. 5

Champion No. 6

Champion No. 7

Champion No. 8

Champion No. 9

Champion No. 10

Pirmos komandos TSM visu Kills/Assists suma: 75

Antros komandos CLG visu Kills/Assists suma: 135

Geriausia komanda CLG

## 2. Klasių String ir ArrayList išeities kodų analizė

Atlikau klasės Timer kodo analizę. Ji turi tokius laukus:

```
private final char value[];
```

```
private final int offset; // The offset is the first index of the storage that is used.
```

```
private final int count; //raidžių skaičius
```

```
private int hash; // Default to 0 //hash kodas
```

```
private static final long serialVersionUID = -6849794470754667710L; //JDK versija
```

Sudėtingiausias yra metodas:

Public boolean regionMatches

Atlikau klasės ArrayList kodo analizę. Ji turi tokius laukus:

```
private static final long serialVersionUID = 8683452581122892189L;
```

```
private static final int DEFAULT_CAPACITY = 10;//nustatyta talpa
```

```
private static final Object[] EMPTY_ELEMENTDATA = {};//skirtas atvejams kai tuscias
```

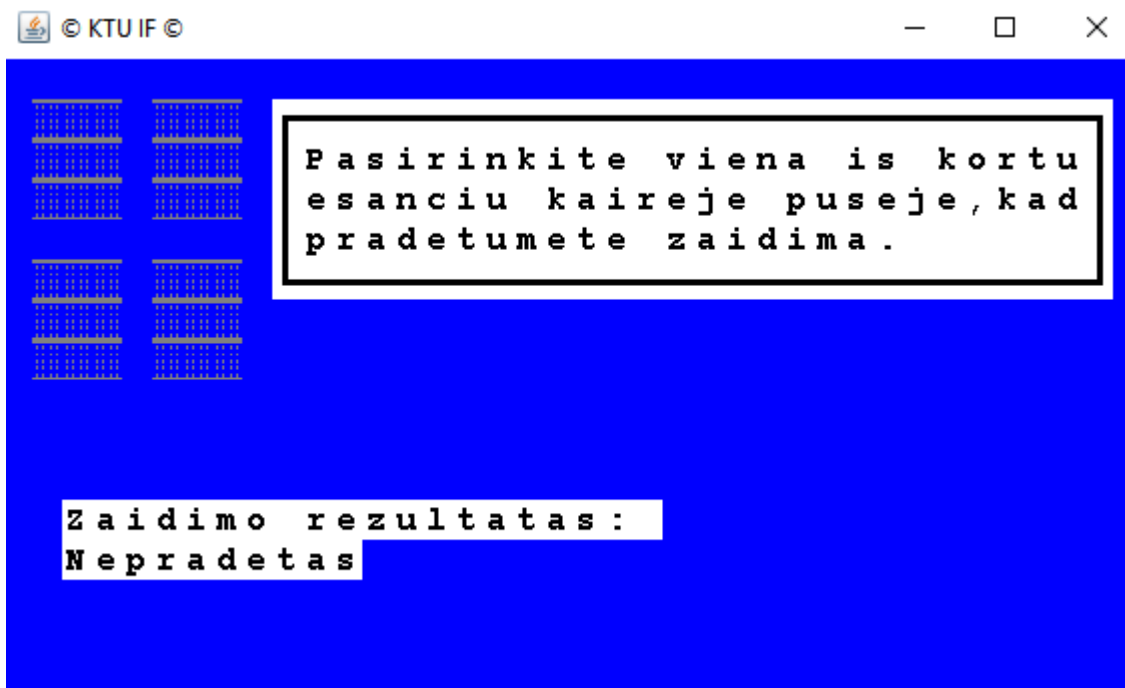
```
private static final Object[] DEFAULTCAPACITY_EMPTY_ELEMENTDATA = {};//vietos  
paskyrimas kai i tuscia pridedamas pirmas elementas
```

Sudėtingiausias yra metodas:

Private boolean batchRemove()

## 2. Vizualios struktūros

Sukūriau žaidimą, kurio pagrindinis tikslas atspėti didžiausios vertės kortą



**Vartotojo instrukcija:** Kairiu pelės mygtuku pasirenkama korta. Jeigu ši korta yra didesnė už kitas, tada jūs laimite ir atvirkščiai. Jeigu dviejų kortų vertės sutampa, tada žiūrima spalva: jeigu spalvos skiriasi, jūs pralaimite, jeigu vienodos, jūs turite 50% šansą laimėti. Pasirinkus kortą rezultatas matomas po užrašu „Zaidimo rezultatas:“. Gavus rezultatą ir norint žaisti dar kartą, reikia spausti 'r' klavišą ir tada 'n' klavišą. Šiuo atveju atsidarys naujas langas kuriame galėsite iš naujo rinktis kortą. Jeigu žaidimo nebenorite tęsti, tada spauskite viršuje dešiniame kampe esantį 'X' simbolį šitaip uždarydami langą ir nutraukdami programos veikimą.

### Išvados

Labai patogiu, kad nereikia pačiam rašyti rūšiavimo burbuliuku metodų. Java yra grynai objektinė kalba, todėl kodas atrodo tvarkingesnis, nėra atskirų, išmėtytų funkcijų.

String objektas turi savyje dažnai naudotinus metodus: replace, split, toUpperCase, toLowerCase.

*Apie NetBeans:* Turi gera klaidų taisymą/rodymą. Gali suformatuoti negražiai parašytą kodą. Turi gražią kodo spalvinimo paletę. Automatiškai atnaujina išorėje pakeistą kodą.

Bendros laboratorinio darbo atlikimo laiko sąnaudos buvo 6 valandų.