

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Кафедра математичного моделювання та аналізу даних

ПРАКТИЧНА РОБОТА № 2

на тему: «Використання нормалізації і вдосконалення програми для обробки великого обсягу даних»

ВАРИАНТ № 1

Виконав: студент 2 курсу групи 22
Спеціальності 124 «Комп'ютерні науки»
Грецький Д. В.
Прийняв: Старший викладач ЗВО
Бережний А.А.

Мета роботи: здобути навички нормалізації і вдосконалення програми для обробки великого обсягу даних

Постановка задачі:

Завдання складається з двох частин. Перша частина завдання дається вам зараз, друга частина буде надана команді, як тільки команда повідомить про виконання першого завдання. Вам надається список учнів місцевої початкової школи разом із їх класом. Список зберігається у файлі **students.txt** (його можна скачати з **stm.khai.edu**). У кожному рядку файлу зберігається інформація про студента.

Формат рядка:

StLastName, StFirstName, Grade, Classroom, Bus, TLastName, TFirstName

Тут, StLastName і StFirstName ідентифікують студента; TFirstName ідентифікує вчителя учня. Bus - це маршрут шкільного автобуса, яким учень їде, щоб добратися до школи. Bus, Grade і Classroom - цілі числа (якщо значення цього поля 0, то це означає **-дитячий садок**, учень, який самостійно прибуває до школи – значення відмінне від 0), а всі інші поля - це рядки.

Ось зразок рядка з файлу:

DROP, SHERMAN, 0, 104, 51, NIBLER, JERLENE

Не дивно, що рядок слід читати «Sherman Drop, який їде автобусом 51 маршрутом, є вихованцем дитячого садка, Mrs. Jerlene Nibler у класі 104».

Ваша мета - написати програму, яка здійснює пошук студентів у текстовому файлі та видає результати пошуку.

Потрібно здійснити такі пошукові запити:

- Враховуючи прізвище студента, знайдіть клас студента, та клас викладача (якщо є більше одного студента з однаковим прізвищем, знайдіть цю інформацію для всіх студентів);
- Враховуючи прізвище студента, знайдіть автобусний маршрут, яким їде студент (якщо є більше одного студента з однаковим прізвищем, знайдіть цю інформацію для всіх студентів);
- Дан викладач, знайдіть список його учнів;
- Враховуючи автобусний маршрут, знайдіть усіх учнів, які ним користуються;
- Знайдіть усіх учнів на певному рівні класу;

Текст програми з коментарями:

```
import java.io.*;
import java.util.Scanner;

public class Schoolsearch {
    public static void main(String[] args) {
        int choice, exit=1;
        File list = new File("list.txt");
        File teachers = new File("teachers.txt");
        if(teachers.exists() && list.exists()){
            System.out.println("File "+teachers.getName()+" and
"+list.getName()+" opened successfully");
        } else{
            System.out.println("File not found!");
            exit=0;
        }
        Scanner scanner = new Scanner(System.in);
        while(exit!=0){
            System.out.println("""
            Choose option by what to find info:
            1) Student
```

```

        2)Teacher
        3)Classroom
        4)Bus
        0)Exit
        """);
choice = scanner.nextInt();
scanner.nextLine(); // додатковий виклик для споживання зайвого
символу \n
switch (choice){
    case 1://Пошук по прізвищу
        System.out.println("Enter student`s last name:");
        String lastname = scanner.nextLine();
        searchByLastname(lastname);
        break;
    case 2://Пошук за викладачем
        System.out.println("Enter teacher`s last name:");
        String teacher = scanner.nextLine();
        searchByTeacher(teacher);
        break;
    case 3://Пошук по прізвищу
        System.out.println("Enter classroom`s number:");
        String classroom = scanner.nextLine();
        searchByClassroom(classroom);
        break;
    case 4://Пошук за викладачем
        System.out.println("Enter bus`s number:");
        String bus = scanner.nextLine();
        searchByBus(bus);
        break;
    case 0:
        exit=0;
        break;
    default:
        System.out.println("Invalid option");
}
}

}
// LIST: 0StLastName, 1StFirstName, 2Grade, 3Classroom, 4Bus
// TEACHERS: 0TLastName, 1TFIRSTName, 2Classroom
private static void searchByLastname(String lastname){
    String line_student, line_teacher;
    Scanner scanner = new Scanner(System.in);
    System.out.println("Choose variant(student info = 1 / student bus =
2)");
    int option = scanner.nextInt();
    scanner.nextLine();// додатковий виклик для споживання зайвого символу
\n
    if(option==1){
        try {
            long starting_time = System.currentTimeMillis();
            BufferedReader list_reader = new BufferedReader(new
FileReader("list.txt"));
            BufferedReader teacher_reader = new BufferedReader(new
FileReader("teachers.txt"));
            //Прохід по всьому файлу
            while ((line_student = list_reader.readLine()) != null){ //поки
не досягнемо кінця файлу
                //розділ рядків на слова
                String[] parts_student = line_student.split(",");
                //Пошук співпадінь прізвищ
                if(parts_student[0].equalsIgnoreCase(lastname)){//ігноруючи

```

```

капіталізацію
        System.out.println("Student: "+parts_student[0]+"
"+parts_student[1]+"  Class: "+
        parts_student[3]+"\\nTechers from class
"+parts_student[3]+": " );
        while((line_teacher = teacher_reader.readLine()) !=
null){
            String [] parts_teacher = line_teacher.split(",");
if(parts_student[3].equalsIgnoreCase(parts_teacher[2])){
            System.out.println(parts_teacher[0]);
        }
    }
    }
    long total_time = System.currentTimeMillis() - starting_time;
    System.out.println("\\nTime it took to find:
"+total_time+"ms\\n");
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
    } else{
        try {
            long starting_time = System.currentTimeMillis();
            System.out.println("Enter bus number:");
            String bus = scanner.nextLine();

            BufferedReader list_reader = new BufferedReader(new
FileReader("list.txt"));
            //Прохід по всьому файлу
            while ((line_student = list_reader.readLine()) != null){ //поки
не досягнемо кінця файлу
                //розділ рядків на слова
                String[] parts_student = line_student.split(",");
                //Пошук співпадінь прізвищ
                if(parts_student[0].equalsIgnoreCase(lastname) &&
parts_student[4].equalsIgnoreCase(bus)){//ігноруючи капіталізацію
                    System.out.println("Student: "+parts_student[0]+"
"+parts_student[1]+"  Bus: "+
                    parts_student[4]);
                }
            }
            long total_time = System.currentTimeMillis() - starting_time;
            System.out.println("\\nTime it took to find:
"+total_time+"ms\\n");
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
// LIST: 0StLastName, 1StFirstName, 2Grade, 3Classroom, 4Bus
// TEACHERS: 0TLastName, 1TFirstName, 2Classroom
private static void searchByTeacher(String teacher){
    String line_list, line_teachers;
    try {
        long starting_time = System.currentTimeMillis();
        BufferedReader teachers_reader = new BufferedReader(new
FileReader("teachers.txt"));

        //Пошук вчителя у файлі
        while((line_teachers = teachers_reader.readLine())!=null){
            // розділ лайну на частини

```

```

        String[] parts_teacher = line_teachers.split(",");
        //пошук класрума по прізвищу
        if(parts_teacher[0].equalsIgnoreCase(teacher)){
            //запам'ятовуємо класрум
            String classroom = parts_teacher[2];
            System.out.println("\n"+parts_teacher[0]+" "+
parts_teacher[1]+"`s students from classroom"+parts_teacher[2]+":");
            //відкриваємо лист в лупі, щоб кожен раз читання починалося
спочатку і
            // не пропускали студентів інших викладачів з однаковими
прізвищами
            BufferedReader list_reader = new BufferedReader(new
FileReader("list.txt"));
            //шукаємо по класруму вчителя студентів, які навчаються у
цьому класрумі
            while ((line_list = list_reader.readLine()) != null){

                // розділ лайну на частини
                String[] parts_student = line_list.split(",");
                //якщо класрум студента співпадає з класрумом в пам'яті
if(parts_student[3].equalsIgnoreCase(classroom)){//ігноруючи капіталізацію
                //виписуємо прізвище студента
                System.out.println(parts_student[0]+"
"+parts_student[1]);
            }
        }
        list_reader.close();
    }
    }
    long total_time = System.currentTimeMillis() - starting_time;
    System.out.println("\nTime it took to find: "+total_time+"ms\n");
} catch (IOException e) {
    throw new RuntimeException(e);
}
}
// LIST: 0StLastName, 1StFirstName, 2Grade, 3Classroom, 4Bus
// TEACHERS: 0TLastName, 1TFirstName, 2Classroom
private static void searchByClassroom(String classroom){
    String line;
    int i = 1;
    try {
        long starting_time = System.currentTimeMillis();
        BufferedReader list_reader = new BufferedReader(new
FileReader("list.txt"));
        //Прохід по всьому файлу
        System.out.println(classroom + "`s students:");
        while ((line = list_reader.readLine()) != null){ //поки не досягнемо
кінця файлу
            //розділ рядків на слова
            String[] parts = line.split(",");
            //Пошук співпадінь прізвищ
            if(parts[3].equalsIgnoreCase(classroom)){//ігноруючи
капіталізацію
                System.out.println(i+" "+parts[0]+" "+parts[1]);
                i++;
            }
        }
        long total_time = System.currentTimeMillis() - starting_time;
        System.out.println("\nTime it took to find: "+total_time+"ms\n");
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

```

```

    }
}
// LIST: 0StLastName, 1StFirstName, 2Grade, 3Classroom, 4Bus
// TEACHERS: 0TLastName, 1TFirstName, 2Classroom
private static void searchByBus(String bus){
    String line;
    int i = 1;
    try {
        long starting_time = System.currentTimeMillis();
        BufferedReader reader = new BufferedReader(new
FileReader("list.txt"));
        //Прохід по всьому файлу
        System.out.println(bus + "`s students:");
        while ((line = reader.readLine()) != null){ //поки не досягнемо
кінця файлу
            //розділ рядків на слова
            String[] parts = line.split(",");
            //Пошук співпадінь прізвищ
            if(parts[4].equalsIgnoreCase(bus)){//ігноруючи капіталізацію
                System.out.println(i+" "+parts[0]+" "+parts[1]);
                i++;
            }
        }
        long total_time = System.currentTimeMillis() - starting_time;
        System.out.println("\nTime it took to find: "+total_time+"ms\n");
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
}
}

```

Результат виконання:

```

1
Enter student's last name:
CIGANEK
Choose variant(student info = 1 / stu
1
Student: CIGANEK MANIE Class: 107
Techers from class 107:
FAFARD
BODZIONY
HANTZ
STEIB
ALPERT
GAMBREL
COOL
KERBS
NISTENDIRK
KERBS
STEIB
BODZIONY
COOL
FAFARD
BODZIONY
GAMBREL
NISTENDIRK
BODZIONY
FAFARD

Student: CIGANEK JANNETTE Class: 107
Techers from class 107:
Student: CIGANEK ZANDRA Class: 102
Techers from class 102:
Student: CIGANEK ZANDRA Class: 105
Techers from class 105:
Student: CIGANEK WAN Class: 106
Techers from class 106:
Student: CIGANEK HYE Class: 105
Techers from class 105:
Student: CIGANEK DICK Class: 102
Techers from class 102:
Student: CIGANEK FLOY Class: 109
Techers from class 109:
Student: CIGANEK DIMPLE Class: 109
Techers from class 109:
Student: CIGANEK HYE Class: 103
Techers from class 103:
Student: CIGANEK JANNETTE Class: 103
Techers from class 103:
Student: CIGANEK RANDOLPH Class: 102
Techers from class 102:
Student: CIGANEK WAN Class: 104
Techers from class 104:

Time it took to find: 9ms

```

Рисунок 1-2 – скріншот виконання пошуку студентів за прізвищем і виведення інформації про студента, а також часу, що зайняли розрахунки

```

1
Enter student's last name:
BUSSMANN
Choose variant(student info = 1 / student bus = 2)
2
Enter bus number:
55
Student: BUSSMANN XUAN Bus: 55
Student: BUSSMANN MANIE Bus: 55
Student: BUSSMANN INGER Bus: 55

Time it took to find: 7ms

```

Рисунок 3 – скріншот виконання пошуку студента за прізвищем і виведення інформації про його автобусний маршрут, а також часу

```

2
Enter teacher's last name:
COOL
COOL REUBEN's students from c
DURAND CARLEE
CLECKLER TOMAS
CORKER HYE
CREMEANS JANNETTE
BUSSMANN XUAN
BREVO XUAN
BREVO ZANDRA
CORKER FLOY
CORONADO INGER
BUSSMANN TOMAS
COVINGTON JANNETTE
BERBES MANIE
CREMEANS ZANDRA
BEX HYE
CORKER INGER
BREVO TOMAS
CORONADO DIMPLE
BOYTER MANIE
BUSSMANN BILLY
COVINGTON JANNETTE
CLECKLER HYE
CORONADO JANNETTE
BEX MANIE
BEX FLOY
CORKER DICK
COMO WAN
CIGANEK RANDOLPH
BUSSMANN INGER
CLECKLER FLOY
CREMEANS CARTER

Time it took to find: 41ms

```

Рисунок 4-5 – скріншот виконання пошуку студентів за прізвищам їх викладача, а також часу, що зайняли розрахунки


```
Enter classroom`s number:
107
107`s students:
1)COOKUS XUAN
2)ELHADDAD SHANTE
3)SWEDLUND SHARRI
4)CIGANEK MANIE
5)COVINGTON TOMAS
6)EARLY TORY
7)LINHART LELA
8)BERBES DICK
9)CORONADO INGER
10)CORONADO JANNETTE
11)CORKER TAMESHA
12)BOYTER WAN
13)BEX JANNETTE
14)CIGANEK JANNETTE
15)COMO FLOY
16)CORKER JANNETTE
17)COVINGTON DICK
18)CLECKLER BILLY
19)BUSSMANN RANDOLPH
20)COOKUS HYE
21)BEX TAMESHA
22)BEAN DICK
23)BEX HYE
24)CORKER ZANDRA
25)BUSSMANN HYE
26)COMO DIMPLE
27)COVINGTON CARTER
28)BUSSMANN XUAN
29)BEAN FLOY
30)BREVO RANDOLPH
31)BOYTER XUAN
32)BOYTER CARTER
33)BOYTER RANDOLPH
34)CORKER TAMESHA

Time it took to find: 12ms
```

Рисунок 6 – скріншот виконання пошуку студентів за номером їх класу, а також часу виконання пошуку.

```

4
Enter bus`s number:
55
55`s students:
1)SWEDLUND SHARRI
2)ZAPPAVIGNA ELLIOTT
3)SAENPHIMMACHA NUMBERS
4)BOYTER WAN
5)KOZOLA BUSTER
6)DEMARTINI DEWAYNE
7)BEX TAMESHA
8)WERST LEON
9)LAMKIN TAMA
10)BREVO FLOY
11)COOKUS XUAN
12)BEAN JANNETTE
13)COVINGTON WAN
14)BEAN MANIE
15)COVINGTON CARTER
16)BEAN TAMESHA
17)COVINGTON FLOY
18)BEX BILLY
19)COMO DICK
20)COVINGTON DIMPLE
21)BERBES TOMAS
22)COVINGTON JANNETTE
23)BOYTER WAN
50)BEAN BILLY
51)CORONADO HYE
52)COVINGTON INGER
53)CLECKLER HYE
54)BREVO RANDOLPH
55)COOKUS BILLY
56)CREMEANS WAN
57)BRODERSEN BILLY
58)CORKER HYE
59)COMO INGER
60)BEX MANIE
61)CREMEANS MANIE
62)CIGANEK HYE
63)BREVO BILLY
64)CLECKLER JANNETTE
65)COMO XUAN
66)COMO WAN
67)BUSSMANN INGER
68)CLECKLER FLOY

Time it took to find: 1ms

```

Рисунок 7-8 – скріншот виконання пошуку студентів за маршрутом автобуса, , а також часу виконання пошуку.

Висновки: здобув навичок нормалізації і вдосконалення програми для обробки великого обсягу даних

Посилання на Вашу Git-сторінку з повним лістингом проєкту (репозиторій повинен бути закритий але з доступом для викладача):

<https://github.com/Torvald3/PR1.git>

