Програмування Java. Інформатика 3-ій курс.

Лабораторна робота №2

З файлу, назва якого вводиться користувачем, прочитати дані в один із стандартних контейнерів. Обробити дані згідно варіанту завдання і записати результати у інший файл.

1. Зчитати рядки з файлу, вилучити усі порожні рядки (ті, які не містять символів або складаються лише з пробілів). Записати у файл, що задається користувачем з консолі, зчитані рядки, попередньо вилучивши з усіх рядків символи, які не є маленькими латинськими літерами, і відсортувати їх.
2. Читаючи з файлу рядки, ті з них, що представляють цілі числа (містять лише цифри), просумувати і вивести на консоль (суму і усі доданки), у решті рядків змінити порядок слів на протилежний, та вивести у файл, заданий користувачем.
3. Читаючи з файлу рядки, виділити з них цифри, просумувати і вивести на консоль (суму і усі доданки). У перший файл, заданий користувачем, вивести відсортований список отриманих чисел. У інший файл вивести рядки, в яких цифр не було.
4. Кожен рядок задає ціле або дробове число(якщо рядок містить лише цифри то вважати його цілим у іншому випадку дробовим, якщо числа не правильний то цей рядок відкидається), порахувати суму дробових чисел, добуток усіх цілих. Вивести відсортовані цілі та дробові числа в файли задані користувачем. Суму і добуток вивести на консоль.
5. Отримати від користувача імена двох файлів кожен з яких містить текст. Обидва тексти розібрати на слова (відкидати пробіли та розділові знаки „ . , : ; ”) привести їх до нижнього регістру(великі букви замінити на маленькі). У файл заданий користувачем вивести слова що зустрічаються одночасно в обох файлах у вигляді відсортованого списку. На консоль вивести два відсортовані списки слів з обох файлів.
6. Кожен рядок задає дробове число, порахувати суму найбільших десяти, або усіх, якщо кількість рядків <10. Вивести відсортовані за спаданням числа в файл, заданий користувачем. Суму і числа, з яких вона складається, вивести в інший файл. Числа відсортувати за зростанням.
7. З файлу заданого користувачем зчитати список слів. Підрахувати кількість входжень цих слів в текст що міститься в іншому файлі який також задається користувачем. Вивести на консоль відсортований список слів за якими здійснювався пошук з кількістю входження кожного з них. В файл заданий користувачем вивести відсортований список слів з тексту в якому здійснювався пошук попередньо вилучивши усі дублікати.
8. З файлу заданого користувачем зчитати список слів(слова це послідовність латинських літер) . Вивести на консоль відсортований перелік латинських літер що зустрічаються у файлі в великому і маленькому регістрі з кількістю їх входжень. У файл заданий користувачем вивести відсортований список слів літери яких різні.
9. З файлу, заданого користувачем, зчитати список слів(слова – це послідовність латинських літер), з іншого файлу зчитати текст. Вивести на консоль відсортований перелік слів, що зустрічаються у другому файлі більш ніж один раз з кількістю їх входжень. У файл, заданий користувачем, вивести список слів які не зустрічаються в першому файлі.
10. Кожен рядок задає ціле число(якщо формат числа неправильний, то цей рядок відкидається), потрібно підрахувати кількість таких рядків, порахувати середнє арифметичне усіх чисел. Вивести відсортовані за зростанням числа, що менші за середнє, в файл, заданий користувачем решту чисел вивести на консоль.
11. З файлу, заданого користувачем, зчитати список слів. Підрахувати кількість входжень цих слів в текст, що міститься в іншому файлі, який також задається користувачем. Вивести на консоль відсортований по спаданню список слів, за якими здійснювався пошук. В файл, заданий користувачем вивести відсортований список слів з тексту, в якому здійснювався пошук, попередньо вилучивши усі дублікати.
12. Отримати від користувача імена двох файлів кожен з яких містить текст. Обидва тексти розібрати на слова (відкидати пробіли та розділові знаки „ . , : ; ”) привести їх до верхнього регістру. У файл, заданий користувачем вивести слова, що зустрічаються окремо в першому і другому файлах у вигляді відсортованого списку. На консоль вивести слова, що зустрічаються в обох файлах.
13. З файлу, заданого користувачем зчитати список слів. Підрахувати кількість входжень цих слів в текст, що міститься в іншому файлі, який також задається користувачем. Вивести на консоль відсортований за спаданням список слів, за якими здійснювався пошук, з кількістю входження кожного з них. В файл, заданий користувачем, вивести відсортований список слів з тексту, в якому здійснювався пошук, попередньо вилучивши усі дублікати.
14. Вилучити порожні рядки, послідовності пробілів замінити одним пробілом. На консоль вивести статистику перетвореного тексту: кількість рядків, кількість символів, кількість пробільних символів. Перетворений текст вивести в файл якій задається користувачем.
15. Вилучити повторення рядків. Відсортувати рядки в алфавітному порядку. Вивести результат в файл, якій задається користувачем.

Приклад. З файлу заданого користувачем зчитати текст(слова – це послідовність латинських літер). Вивести на консоль відсортований перелік латинських літер що не зустрічаються в тексті. У файл заданий користувачем вивести відсортований список слів літери яких усі різні.

**import** java.io.BufferedReader;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Scanner;

**public** **class** MainClass {

**public** **static** **void** main(String[] args) {

**try** {

Scanner in = **new** Scanner(System.***in***); // об'єкт для введення даних

System.***out***.print("Введіть назву файлу:");

String fileName = in.next();

// відкриття файду для читання

BufferedReader reader = **new** BufferedReader(**new** FileReader(fileName));

String line;

// колекція для збереження рядків прочитаних з файлу

ArrayList<String> lines = **new** ArrayList<String>();

**while** ((line = reader.readLine()) != **null**) // читання рядків з файлу

{

lines.add(line);

}

reader.close();

/\*інший варіант введення

Scanner in = new Scanner(new File("filename.txt"));

while (in.hasNextLine())

lines.add(in.nextLine());

in.close(); \*/

**int** i;

// колекція букв, які є у файлі

ArrayList<String> letters = **new** ArrayList<String>();

**for** (i=0;i<lines.size();i++)

{

line = lines.get(i);

**int** j = 0;

**while** (j<line.length()) // цикл по буквах рядка

// додаємо букви до колекції

letters.add(String.*valueOf*(line.charAt(j++)));

}

// вивід у консоль

System.***out***.println("Літери, яких немє в тексті:");//проходимо по усіх буквах

**for** (**char** c='A'; c<='Z';c++)

**if** (!letters.contains(String.*valueOf*(c))) //перевіряємо чи входять у колекцію

System.***out***.print(c); // друкуємо

**for** (**char** c='a'; c<='z';c++)

**if** (!letters.contains(String.*valueOf*(c)))

System.***out***.print(c);

System.***out***.println();

ArrayList<String> words = **new** ArrayList<String>(); // колекція слів у файлі

**for** (i=0;i<lines.size();i++)

{

line = lines.get(i);

String[] tmp = line.split("[ ,.:;-?!]+"); // вибираємо слова

**for** (**int** j=0;j<tmp.length;j++)

words.add(tmp[j]); // додаємо їх до колекції

}

// вилучення слів у яких є однакові букви

i=0;

**while** (i<words.size()) // цикл по словах колекції

{

line = words.get(i);

**boolean** f = **false**;

**for** (**int** j=0;j<line.length();j++)

**for** (**int** l=j+1;l<line.length();l++)

**if** (line.charAt(j)==line.charAt(l)) //перевірка чи є однакові букви

f = **true**;

**if** (f)

words.remove(i); / вилучення слова з колекції

**else**

i++;

}

Collections.*sort*(words); // сорування слів у колекції

// виведення результату у файл

System.***out***.print("Введіть назву вихідного файлу: ");

fileName = in.next();

in.close();

// відкриття файлу для запису

FileWriter writer = **new** FileWriter(fileName);

// змінна для побудови результату (кожне слово у новому рядку)

StringBuilder strB = **new** StringBuilder();

**for** (i=0;i<words.size();i++)

strB.append(words.get(i)).append(System.*getProperty*("line.separator"));

writer.write(strB.toString()); // запис у файл

writer.close();

System.***out***.println("Готово");

}

// обробка виключної ситуації, коли не буде файлу

**catch** (FileNotFoundException ex) {

System.***out***.println("Файл не знайдено");

}

// обробка решта виключних ситуацій

**catch** (Exception ex) {

System.***out***.println("Помилка!!!");

}

}

}