

Лабораторна робота №1.

Утилітарні класи Java SE. Обробка масивів і рядків. Інтерактивні консольні програми для платформи

Мета

- Розробка власних утилітарних класів.
- Набуття навичок вирішення прикладних задач з використанням масивів і рядків.
- Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

Вимоги

1. Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи.
2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню :
 - a. введення даних;
 - b. перегляд даних;
 - c. виконання обчислень;
 - d. відображення результату;
 - e. завершення програми і т.д.
3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:
 - a. параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);
 - b. параметр “-d” чи “-debug”: в процесі роботи програми відображаються додаткові дані, що полегшують налагодження та перевірку працездатності програми: діагностичні повідомлення, проміжні значення змінних, значення тимчасових змінних та ін.
4. При вирішенні прикладних задач використовувати латинку .
5. Продемонструвати використання об’єктів класу StringBuilder або StringBuffer .

6. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. Helper Class) та для обробки даних використовувати відповідні статичні методи.

7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex (Pattern , Matcher та ін.), а також відповідні методи класу String (matches , replace , replaceFirst , replaceAll , split).

1.1 Розробник

Горностай Богдан, КН-108, номер варіанту індивідуального завдання – 8.

1.2 Задача

Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом напроки) іншого (наприклад: “abc”-“cba”, “def”-“fed”). Результат вивести у вигляді таблиці.

2 Опис програми

Дана розроблена програма дозволяє шукати слова обіги в введеному тексті, реалізована робота двох можливих режимів роботи: -h – режим при якому на початку програма з’являється певна інформація(інформація про розробника, суть індивідуального завдання, можливі функції консольного меню), -d або ж -debug – спеціальний режим роботи, при якому користувачу виводить на екран проміжковий результат редагування рядка.

2.1 Засоби ООП

Для виконання завдання був використаний клас Ex8, який містив певні статичні поля і методи необхідні для обрахунків й порівнянь.

2.2 Ієрархія та структура класів

Клас Ex8 з статистичними полями.

```
import java.util.Scanner;

public class Ex9 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input text: ");
        String text = ' '+in.nextLine()+' ';
        int count = 0;
        for(char c : text.toCharArray()){
            if(c == ' '){
```

```

        count++;
    }
}
count--;
System.out.printf("Result is: %s\n", text);
in.close();
String[] word = new String[count];
int space1 = 0, space2 = 0;
for(int i = 0; i < count; i++)
{
    space2 = text.indexOf(' ', space2+1);
    word[i] = text.substring(space1, space2);
    space1 = space2;
}
String[] antiword = new String[count];
for(int k = 0; k < count; k++)
{
    char[] charArray = word[k].toCharArray();
    for (int i = charArray.length - 1; i >= 0; i--) {
        antiword[k] += charArray[i];
    }
}
for(int i = 0; i < count; i++)
{
    for(int k = i; k < count; k++)
    {
        if((k==i) && (k < (count -1)))
        {
            k++;
        }
        if((word[i].substring(1) + " ").equals(antiword[k].substring(4)) ==
true)
        {
            System.out.printf("Rrr: %s %s\n",word[i].substring(1),
word[k].substring(1) );
            antiword[k] = word[i]+"*";
        }
    }
}
}
}

```

3. Варіанти використання

Дана програма може використовуватись для пошуку слів одне з яких є обігом іншого, а також обчислення їхньої кількості.

ВИСНОВКИ

У ході роботи у мене розвинулись навички написання власних класів та вирішення з їхньою допомогою операцій над рядками. Окрім цього я набув навичок вирішення прикладних задач з використанням масивів і рядків а також реалізував діалоговий режим роботи з користувачем в консольних програмах мовою Java.