

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту

Лабораторна робота №8

на тему:

“Утилітарні класи Java SE.

Обробка масивів і рядків.

Інтерактивні консольні програми для платформи”

з курсу:

“Об’єктно-орієнтоване програмування ”

Виконала:
ст. гр. КН-110
Турколяк
Вікторія-Ірина
Викладач:
Гасько Р.Т

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 для вирішення прикладної задачі за номером, що відповідає номеру студента в журналі групи з поверненням до початку. Наприклад 1 ->1, 2->2, ..., 15->15, 16->1, 17->2, ..., 30->15, 31-1 і т.д.
2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:
 - a. введенняданих;
 - b. перегляд даних;
 - c. виконанняобчислень;
 - d. відображення результату;
 - e. завершенняпрограмиіт.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 (<http://docs.oracle.com/javase/8/docs/api/java/lang/...>)
6. Застосувати функціональну (процедурну) декомпозицію - розробити

власні утилітарні класи (особливий випадок допоміжного класу, див. https://en.wikipedia.org/wiki/Helper_class) та для обробки даних використовувати відповідні статичні методи. <https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>

7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету `java.util.regex` (`Pattern`, `Matcher` та ін.), docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html> <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Matcher.html>

а також відповідні методи класу `String` (`matches`, `replace`, `replaceFirst`, `replaceAll`, `split`) - docs.oracle.com/javase/7/docs/api/java/lang/String.html

Варіант 2

Ввести декілька рядків. Упорядкувати, а потім вивести рядки за алфавітом (перший пріоритет) та в порядку зростання їх довжини (другий пріоритет).

Використовується `enum` для збереження пунктів меню,

Клас `StringHelper` який маніпулює масивом рядків

Клас `MenuTool` для обробки вхідних даних

Текст основних моментів програми

```
public class MenuTool {
    private static final MenuTool INSTANCE;
    private StringHelper manager = new StringHelper();
    private Scanner scanner = new Scanner(System.in);
    static {
        System.out.println("FINAL");
        INSTANCE = new MenuTool();
    }

    private MenuTool(){
    }

    public static MenuTool getInstance(){
        return INSTANCE;
    }

    public void run() {
        System.out.println("Hello ");
        while (true) {
```

```

showMenu();
int act = scanner.nextInt();
Menu menu = Menu.getMenu(act);
switch (menu) {
    case ADD_TEXT: {

        manager.add();
        break;
    }
    case ORDER_TEXT: {
        System.out.println("How do you want to
order 1-letter 2-length");
        int k=scanner.nextInt();
        if(k==1){
            manager.orderByLetter();
        }else if(k==2){
            manager.orderByLength();
        }
        else System.out.println("try again");
        break;
    }
    case DISPLAY_TEXT: {
        manager.displayText();
        manager.order();
        break;
    }
    case HELP:{
        System.out.println("The program is done by

" +
            " VictoriaTurkolyak " +
            " KN-110 " +
            " Variant 2 ");

        System.out.println("1-Enter strings to
work with");

        System.out.println("2-Work with them");
        System.out.println("3-See result");
        System.out.println("4-exit from system");
        break;}
    case DEBUG:{
        System.out.println("Debug is active");
        manager.debugmod(true);
        break;
    }
    case EXIT: {
        System.out.println("Good bye");
        return;
    }
    case DEFAULT: {
    }
    default: {

```

```

        System.out.println("Unknown command !!!");
    }
}

}

private void showMenu() {
    System.out.println("Choose your act");
    System.out.println("1) Add text");
    System.out.println("2) Order by letter/length");
    System.out.println("3) Display all students");
    System.out.println("4) Help");
    System.out.println("5) Debug mode");
    System.out.println("6)Exit");
}

}

public final class StringHelper {
    int size;
    boolean flag=false;
    private String[] strings = new String[20];

    private Scanner scanner = new Scanner(System.in);

    private void copyArray() {
        String[] tmp = new String [strings.length * 2];
        System.arraycopy(strings, 0, tmp, 0, strings.length);
        strings = tmp;
    }

    public void debugmod(boolean t) {
        System.out.println("debug");
        flag=true;
    }

    public void add()
{
    if(flag) System.out.println("add text to work with");
    System.out.println("How much strings you want to add");
    if(flag) System.out.println("Waiting for answeare");
    size=scanner.nextInt();
    System.out.println("enter");
    if(flag) System.out.println("If size>20 wait because i change
the size of array");
    while (size >= strings.length) {
        copyArray();
    }
    if(flag) System.out.println("Waiting for text ");
    for(int i=0;i<=size;i++){
        strings[i] = scanner.nextLine();
    }
}
}

```

```

}

    public String[] orderByLetter() {
        if(flag) System.out.println("If size>20 wait because i
change the size of array");
        while (size >= strings.length) {
            copyArray();
        }
        if(flag) System.out.println("sorting...");
        int steps=0;
        String[] tmp=strings.clone();
        String temp;
        for (int i = 0; i <=size; i++)
        {
            for (int j = i + 1; j <=size; j++)
            {
                if (tmp[i].compareTo(tmp[j])>0)
                {
                    temp = tmp[i];
                    tmp[i] = tmp[j];
                    tmp[j] = temp;
                    steps++;
                }
            }
        }
        System.out.println("Sorted");
        if(flag) System.out.println("It took "+steps+" steps to
sort");
        return tmp;
    }

    public String[] orderByLength() {
        if(flag) System.out.println("If size>20 wait because i
change the size of array");
        while (size >= strings.length) {
            copyArray();
        }
        int steps=0;
        if(flag) System.out.println("sorting...");
        String[] tmp=strings.clone();
        for (int i=1 ;i<=size; i++)
        {
            String temp = tmp[i];

            // Insert s[j] at its correct position
            int j = i - 1;
            while (j >= 0 && temp.length() < tmp[j].length())
            {
                tmp[j+1] = tmp[j];
                j--;
                steps++;
            }
        }
    }

```

```

        tmp[j+1] = temp;
    }
    System.out.println("Sorted");

    if(flag) System.out.println("It took "+steps+" steps to
sort");
    return tmp;
}

public void displayText(){
    if(flag) System.out.println("If size>20 wait because i
change the size of array");
    while (size >= strings.length) {
        copyArray();
    }
    System.out.println("Your entered");
    for(int i=0;i<=size;i++){
        System.out.println(strings[i]);
    }
    if(flag) System.out.println("Sorted in both ways(we are
sorting copies!)");
    System.out.println("Do you want to see sorted array
separetely? 1-Yes,2-N0");
    int d=scanner.nextInt();
    if(d==1) {
        System.out.println("Sorted by length");
        for (int i = 0; i <= size; i++) {
            System.out.println(orderByLength()[i]);
        }
        for(int i=0;i<=size;i++){
            System.out.println(orderByLetter()[i]);}
    }
    for(int i=0;i<=size;i++){
        System.out.println(order()[i]);
    }
}

public String[] order(){
    int steps=0;
    String[] tmp=strings.clone();
    String temp;
    for (int i = 0; i <=size; i++)
    {
        for (int j = i + 1; j <=size; j++)
        {
            if (tmp[i].compareTo(tmp[j])>0)
            {
                temp = tmp[i];
                tmp[i] = tmp[j];
                tmp[j] = temp;
                steps++;
            }
        }
    }
}

```

```

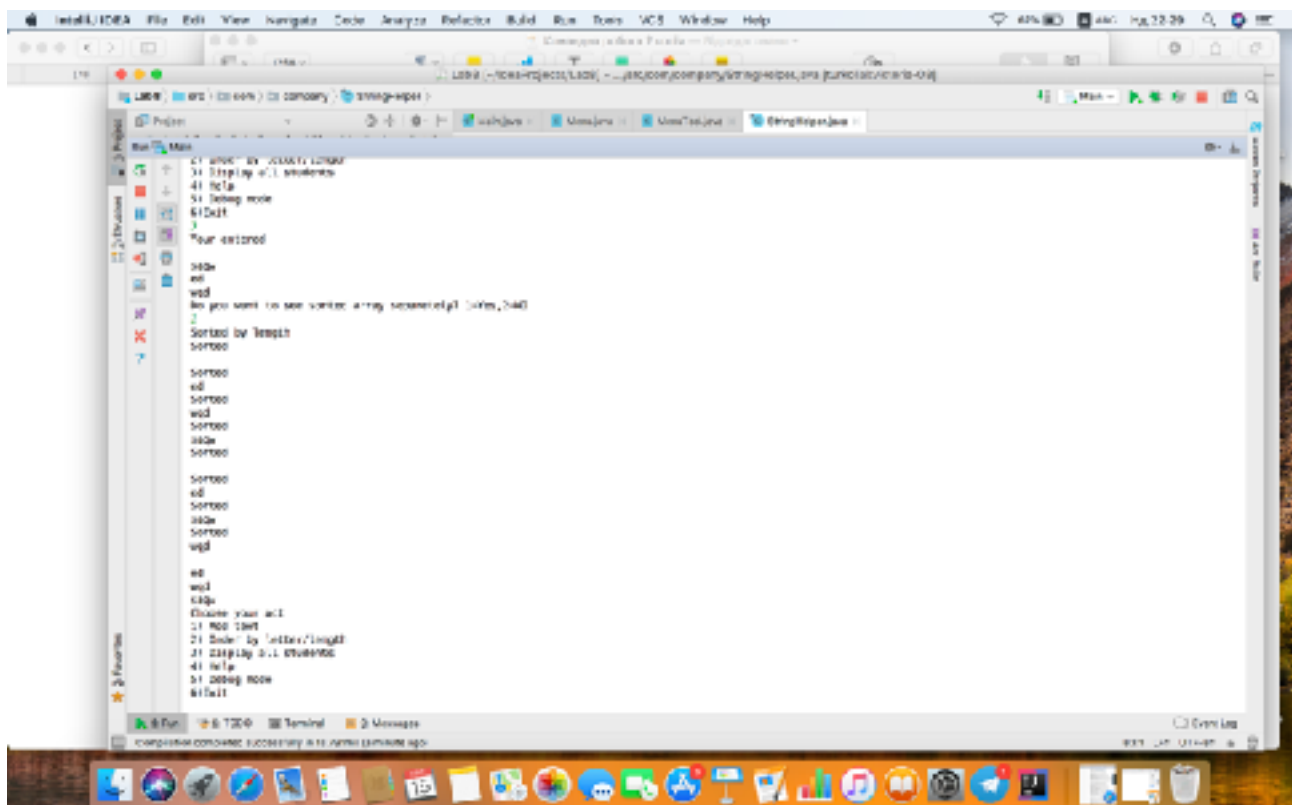
    }
    for (int i=1 ;i<=size; i++)
    {
        String t = tmp[i];

        // Insert s[j] at its correct position
        int j = i - 1;
        while (j >= 0 && t.length() < tmp[j].length())
        {
            tmp[j+1] = tmp[j];
            j--;
            steps++;
        }
        tmp[j+1] = t;
    }
    return tmp;
}

private String getInputString(String message){
    System.out.print(message);
    String input = "";
    while (input.isEmpty()) {
        input = scanner.nextLine();
    }
    return input;
}
}
}

```

Приклад використання програми



Висновок на даній лабораторній роботі я навчилась працювати з допоміжними класами і створювати інтерактивні меню

Г