

4. Інтерактивні консольні програми для платформи Java SE

Мета: Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів та рядків.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Шарма Олександр Раджнішович
- НТУ “ХПІ” КІТ-1196
- Варіант 25

1.2 Загальне завдання

1. Використовуючи програму рішення завдання лабораторної роботи №3, відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:
 - введення даних;
 - перегляд даних;
 - виконання обчислень;
 - відображення результату;
 - завершення програми і т.д.
2. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:
 - параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);
 - 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.

1.3 Задача

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

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі відсутні об’єктно-орієнтовані методи.

2.2 Ієрархія та структура даних

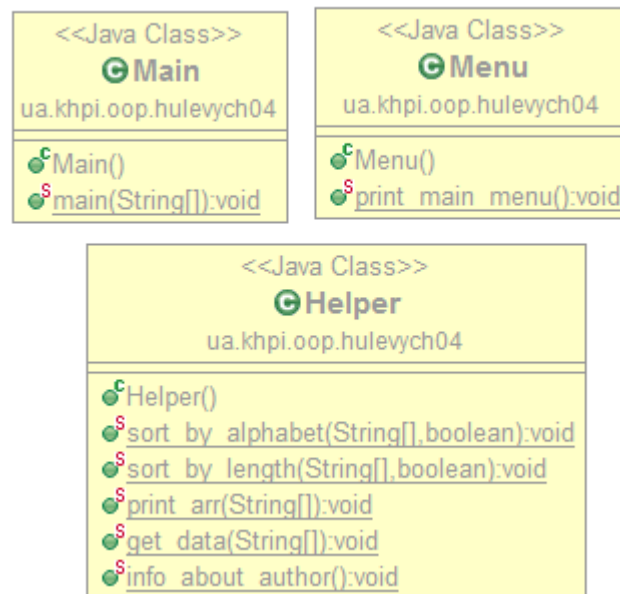


Рисунок 1 – діаграма класів

2.3 Важливі фрагменти програми

```
public class Helper {

    public static void sort_by_alphabet(String[] array, boolean flagg ) {
        if(flagg) Arrays.sort(array);
        else {
            System.out.println("This function sorts array of strings by alphabet");
        }
    }

    public static void sort_by_length(String[] array, boolean flagg) {
        if(flagg) {
            String temp = new String();

            int i, j, nMin;
            for( i=0; i < array.length; i++)
            { nMin=i;
              temp=array[i];
              for( j=i+1; j < array.length; j++) // цикл вибору найменшого елемента
                  if (array[j].length() < temp.length())
                  { nMin=j; // индекс найменшого элемента
                    temp=array[j]; // значение найменшого элемента
                  }
              array[nMin] = array[i];
              array[i] = temp; // меняем местами найменший с i[i]
            }
        }
    }
}
```

Рисунок 2 – Методи допоміжного класу

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма створена для роботи у двох режимах. У режимі `-d` програма працює у режимі налагоджування, з переглядом поточних даних та циклічного використання методів. У режимі `-h`, програма працює у інформативному режимі, з виведенням інформації про методи та інформації про автора.

```
Enter a couple of strings(5)
Car
Bear
Kangaroo
Ca
Wombat
1.Enter data
2.Show current data
3.Sort by alphabet
4.Sort by length
5.Exit
2
Your array is:
0.Car
1.Bear
2.Kangaroo
3.Ca
4.Wombat

1.Enter data
2.Show current data
3.Sort by alphabet
4.Sort by length
5.Exit
```

Рисунок 3 – результати роботи програми у режимі `-d`

```
Main(4) [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.222.1
@Author: Andrii Hulevych 1.KIT 102.8a
@Date: 01.11.2019
@Version 1.1
1.Enter data
2.Show current data
3.Sort by alphabet
4.Sort by length
5.Exit
1
The function get_gata scans the data from input
1.Enter data
2.Show current data
3.Sort by alphabet
4.Sort by length
5.Exit
2
This function prints array in console
1.Enter data
2.Show current data
3.Sort by alphabet
4.Sort by length
5.Exit
```

Рисунок 4 – результати роботи програми у режимі `-h`

ВИСНОВКИ

В даній лабораторній роботі було розроблено утилітарний клас та набуто навичок вирішення прикладних задач із використанням масивів та рядків