

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.**

1.1 Розробник

Гірник Юрій, КН-108, номер варіанту індивідуального завдання – 6.

1.2 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:

- а. введення даних;
- б. перегляд даних;
- с. виконання обчислень;
- д. відображення результату;
- е. завершення програми і т.д.

3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:

- а. 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`).

1.3 Задача

Ввести текст. З тексту видалити всі символи, крім пропусків, які не є буквами. Пропуски, що повторюються, замінити на одиночні. Між послідовностями літер, де знаходяться розділові знаки, залишити хоча б один пропуск (“a,b,c” -> “a, b, c”). Вивести початковий текст та результат.

2 Опис програми

Дана розроблена програма дозволяє обробляти(відповідно до індивідуального завдання) введений текст, 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 утилітарний клас `edittext`, який містив певні статичні поля і методи необхідні для обрахунків.

2.2 Ієрархія та структура класів

1. Клас `Main`, який містить 1 функцію – `main`.

2. Клас Edittext з статистичними полями та декількома функціями: (setLstr, print та editor, вигляд якого показано в наступному пункті)

```
public class Edittext {
    static StringBuilder Lstr;
    static int k;

    static void setLstr(String str){
        Lstr = new StringBuilder(str);
        k = Lstr.length();
    }

    static void print(){
        System.out.println(Lstr);
    }
}
```

2.3 Важливі фрагменти програми.

Функція обробки тексту:

```
static void editor(int debug){
    for(int i = 0; i<k; i++){
        if (debug == 1) System.out.println("Проміжковий результат: "+Lstr);
        if ((Lstr.charAt(i) == ',' && Lstr.charAt(i+1) == ',')
            || (Lstr.charAt(i) == ',' && Lstr.charAt(i-1) == ','))
        {
            Lstr.deleteCharAt(i--);
            k--;
        }

        if (!Character.isAlphabetic(Lstr.charAt(i)) && Lstr.charAt(i) != ' ' && Lstr.charAt(i) != ',')
        {
            Lstr.deleteCharAt(i);
            i--;
            k--;
            continue;
        }
        if (Character.isAlphabetic(Lstr.charAt(i))) continue;
        else
            if ((Lstr.charAt(i) == ' ' && Character.isAlphabetic(Lstr.charAt(i+1)))
                {
                    Lstr.insert(++i, ' ');
                    k++;
                }
                if (Lstr.charAt(i) == ' ' && Lstr.charAt(i+1) == ' ')
                {
                    Lstr.deleteCharAt(i--);
                    k--;
                }
            }
    }
}
```

3. Варіанти використання

Дана програма може використовуватись для редагування рядків, написаних з помилками, надмірними пропусками та сторонніми символами.

ВИСНОВКИ

У ході роботи розвинулись навички написання власних утилітарних класів та вирішення з їхньою допомогою задач над рядками з використанням класу StringBuilder. Також реалізовано можливість команд з боку

користувача, а, крім цього, обробки командного рядка для визначення режиму роботи програми.