

Лабораторна робота №1.

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. 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);

б. 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

Жеребецький Олег, КН-108, номер варіанту індивідуального завдання – 11.

1.2 Задача

Ввести текст. У тексті слова заданої довжини замінити зазначеним рядком. 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 та змінні редакування рядка.

2.1 Засоби ООП

Для виконання завдання був використані утилітарні класи `List` і `Menu`, які містив певні статичні методи необхідні для обрахунків.

1. Клас Main, який містить 1 функцію – main. 2. Клас List з декількома функціями: (str_work , scan_text, scan_list, scan_word, out_d, вигляд якого показано в наступному пункті)

```
package ua.lpnuai.oop.surname01;

public class Main {

    public static void main(String[] args) {
        boolean Point = false;

        if (args.length > 0) Point = Menu.prehelp(args[0]) ;

        String str_text = new String();
        String str_list = new String();
        int word = 0;

        while (true) {
            switch (Menu.comscanwork())
            {
                case(1):
                    str_text = List.scan_text();
                    str_list = List.scan_list();
                    word = List.scan_word();
                    break;
                case(2):
                    List.Out_d(str_text, str_list, word);
                    break ;
                case(3):
                    str_text = List.str_work(str_text, str_list, word,Point);
                    if (Point==true) List.Out_d(str_text, str_list, word);
                    break ;
                case(4):
                    System.out.println("\nText:\t"+str_text);
                    break ;
                case(0):
                    System.out.println( ".\n\t\t\t\t\t\t\t\tGoooodbye! \t" );
                    System.exit(0);
                    break ;
                default:
                    break;
            }
        }
    }
}
```

2.3 Важливі фрагменти програми.

Функція обробки тексту:

```
35
36 public static String str_work(String s1,String s2,int s3,boolean deb) {
37     String str_text = new String();
38
39     s1+=" \0";
40     if (deb==true) System.out.println("Add to text end sign (+\" \\0\")");
41     char wor[]=s1.toCharArray();
42     int i=0,numl= 0 , point=0;
43     if (deb==true) System.out.println( "\nVariable for moving srougt ( i = 0 ), word length (numl = 0), calculation mod point = 0\n"
44     do
45     {
46         if (point == 0 )
47         {
48             if(wor[i]!=' ')
49             {
50
51                 point=1;
52                 numl++;
53                 if (deb==true) System.out.println("\npoint: "+point+"\tnuml: "+numl);
54             }
55             else
56             {
57                 str_text += wor[i]; if (deb==true) System.out.println("\nText:\t "+str_text);
58             }
59         }
60         else
61         {
62             if(wor[i]==' ')
63             {
64
65                 point=0;if (deb==true) System.out.println("\npoint: "+point+"\tEnd of word");
66                 if (numl==s3)
67                 {
68
69                     str_text += s2;
70                     str_text += wor[i];
71                 }
72             }
73         }
74     }
75 }
```

```
2
3         if(wor[i]==' ')
4         {
5
6             point=0;if (deb==true) System.out.println("\npoint: "+point+"\tEnd of word");
7             if (numl==s3)
8             {
9
10                 str_text += s2;
11                 str_text += wor[i];
12                 if (deb==true) System.out.println("\nText:\t"+str_text);
13             }
14             else
15             {
16                 for(int j=numl;j>=0; j--)
17                 {
18                     str_text += wor[i-j];if (deb==true)System.out.println("\nText:\t"+str_text);
19                 }
20             }
21             numl=0; if (deb==true)System.out.println("\nnuml:\t"+numl+"\tEnd of word");
22         }
23         else
24         {
25             numl++;if (deb==true)System.out.println("\nnuml:\t"+numl+"\tWord lenght calculating
26         }
27     }
28
29     i++; if (deb==true)System.out.println("\ni:\t"+i+"\t-char number");
30 }
31 while (wor[i]!='\0');
32 System.out.println( ".\nWork done! \t" );
33 return str_text;
34 }
35 }
```

3. Варіанти використання

Дана програма може використовуватись для редагування рядків, написаних з помилковими словами та сторонніми символами.

ВИСНОВКИ

У ході роботи розвинулись навички написання власних утилітарних класів та вирішення з їхньою допомогою задач над рядками. Також реалізовано можливість команд з боку користувача, а, крім цього, обробки командного рядка для визначення режиму роботи програми.