

Інтерактивні консольні програми для платформи Java SE

Мета роботи: Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

Вимоги

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.

Розробник: Рябов Олексій Володимирович КІТ119а №18.

Опис програми

Засоби ООП: клас, метод класу.

Структура класів: один публічний клас Main, один утилітарний клас Helper.

Важливі фрагменти програми:

```
private static int mean = 0;

static ArrayList<String> line = new ArrayList<String>();

static private void findmean(ArrayList<String> line) {
    int m = 0;
    for(int i =0;i<line.size();i++)
```

```

        m+=line.get(i).length();
        setMean(m/line.size());
        if(debug)
            System.out.println("Mean: " + mean);
        return;
    }

    private static boolean debug = false;

    public static void debugMod(boolean debugMod) {
        debug = debugMod;
        System.out.printf("Debug mod ");
        if (debug) {
            System.out.printf("on\n");
        }else
        {
            System.out.printf("off\n");
        }
    }

    public static void start(StringBuilder ln) {

        if(debug) {
            System.out.println("Start line: " + ln);
        };

        if(ln==null||ln.length()==0) {
            System.out.println("Line is empty");
            return;
        }

        clottingSpace(ln);
        line = findArray(ln);
        findmean(line);
        if(getMean() == 0) {
            throw new NullPointerException("Line have nott element or size of
line is zero");
        }
    }

```

```
}
```

```
static private ArrayList<StringBuilder> findArray(StringBuilder ln){
    ArrayList<StringBuilder> line = new ArrayList<StringBuilder>();
    ArrayList<Integer> IndexSpace = new ArrayList<Integer>();
    int length = ln.length();
    for (int i = 0; i<length;i++) {
        if(ln.charAt(i)==' ')
            IndexSpace.add(i);
    }
    if(IndexSpace.size() == 0) {
        line.add(new StringBuilder(ln));
    }
    else
    {
        line.add(new StringBuilder(ln.substring(0, IndexSpace.get(0))));
        IndexSpace.add(length);
        for(int i=1;i<IndexSpace.size();i++) {

            if(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))!="")
                line.add(new StringBuilder(ln.substring(IndexSpace.get(i-1)+1,
IndexSpace.get(i))));
        }
    }
    return line;
}
```

```
static private StringBuilder clottingSpace(StringBuilder line) {
```

```
    for(int i = 0; i< line.length()-1;i++)
    {
        if(line.charAt(i)==' '&&line.charAt(i+1)==' ') {
```

```

        line.deleteCharAt(i+1);
        i--;
    }
}
if(line.charAt(line.length()-1)==' ')
    line.deleteCharAt(line.length()-1);
if(line.charAt(0)==' ')
    line.deleteCharAt(0);
if(debug)
    System.out.println("Line after clotting space: " + line);
return line;
}

```

```

static public void printline() {
    if(line.isEmpty()) {
        System.out.println("Line is empty");
        return;
    }
    if(debug)
        System.out.println("Count line: " + line.size());
    System.out.println("Lines are bigger than average: ");
    for(StringBuilder n : line) {
        if(n.length() >= mean) {
            System.out.println(n+"\nKenght of line:" + n.length());
        }
    }
    System.out.println("Lines smaller than average: ");
    for(StringBuilder n : line) {
        if(n.length() < mean) {
            System.out.println(n+"\nLenght of line:" + n.length());
        }
    }
}

```

```

public static int getMean() {
    return mean;
}

public static void setMean(int mean) {
    Helper.mean = mean;
}

```

Результати роботи

```

1.Enter line
2.Show data
3.Start
4.Show result
5.End
Command id:
1
13412341234 341234 3241
1.Enter line
2.Show data
3.Start
4.Show result
5.End
Command id:
2
13412341234 341234 3241
1.Enter line
2.Show data
3.Start
4.Show result
5.End
Command id:
3
1.Enter line
2.Show data
3.Start
4.Show result
5.End
Command id:
4
Lines are bigger than average:
13412341234
Kenght of line:11
Lines smaller than average:
341234
Lenght of line:6
3241
Lenght of line:4
1.Enter line
2.Show data
3.Start
4.Show result
5.End
Command id:
5
.

```

Висновки

Оволодів навичками роботи з консольним компілюванням, створення консольного меню та режиму “debug”.