

1. Тема роботи

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

1.1 Розробник: Мокрик Ярослав Любомирович; КН-108; варіант 19 (завдання 4).

1.2

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. Прогонювати 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. У кожному слові тексту k-ту літеру замінити заданим символом. Якщо k більше довжини слова, видати повідомлення та заміну не виконувати. Початкові дані та результат вивести у вигляді таблиці.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Я застосував 3 класи, кожен з яких містить статичні методи. Таким чином, їх можна викликати, не створюючи екземпляр класу. До того ж, я використовував клас `StringBuilder`, щоб створити рядок, який можна змінювати в ході роботи програми.

2.2 Ієрархія та структура класів

Клас `Main` - головний клас. Керує роботою всієї програми і викликає статичні методи інших класів.

Клас `Menu` - клас меню. Керує показом головного меню, приймає ввід користувача і викликає відповідні методи.

Клас `StringMangager` - клас менеджера рядків. Керує вводом рядка, числа і символу. Генерує новий рядок відповідно до вимог завдання.

2.3 Важливі фрагменти програми

Цей метод генерує рядок відповідно до вимог програми.

```
public static void computeResult() {
    res = new StringBuilder();
    for (int i = 0; i < str.length(); i++) {
        if ((i+1) % k != 0 || i == 0)
            res.append(str.charAt(i));
        else
            res.append(c);
        if (Main.debug)
            System.out.println(res);
    }
}
```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Головне меню:

```
a. Input data
b. Output data
c. Compute
d. Quit
Your option: a
```

Введення даних:

```
Your option: a
Enter the string: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
Enter the number k and the letter to replace: 4 I
```

Виведення в форматі таблиці:

```
Your option: b
|-----|
|Original string: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor|
|-----|
|New string: LorIm iIsumIdolIr sIt aIet,IconIectIturIadiIiscIng IlitI seI doIeiuImodItemIor Inci|
|-----|
```

Повідомлення аргументу -h (-help):

```
Author: Yaroslav Mokryk, Group KN-108
Task:
Ввести текст. У кожному слові тексту k-ту літеру замінити заданим
символом. Якщо k більше довжини слова, видати повідомлення та заміну
не виконувати. Початкові дані та результат вивести у вигляді таблиці.
Menu items:
Input data - input the data (string, number and character) through the console
Output data - output a table with the original and result strings
Compute - replace every k-th character with the input character
Quit - exit the program
Command line arguments:
-h or -help - display this help message
-d or -debug - run the program in debug mode
```

Обчислення результату в режимі -d (-debug):

```
H
He
Hel
HelI
HelIo
HelIo
HelIo w
HelIo wI
HelIo wIr
HelIo wIrl
HelIo wIrld
HelIo wIrldI
HelIo wIrldIh
HelIo wIrldIhe
HelIo wIrldIhel
```

ВИСНОВКИ Я навчився розробляти власні утилітарні класи. Я навчився виконувати прикладні завдання з використанням масивів та рядків. Я навчився реалізовувати діалогову програму в консолі Java.