

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

Гринишин Анастасія, КН-108, номер варіанту індивідуального завдання – 10.

1.2 Задача

Ввести текст. Визначити та вивести, яких літер (голосних чи приголосних) більше в кожному реченні тексту. Результат вивести у вигляді таблиці.

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 редакування рядка.

2.1 Засоби ООП Для виконання завдання був використаний утилітарний клас `Audit`, який містив певні методи необхідні для обрахунків.

2.2 Ієрархія та структура класів

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

2. Клас `Audit` з декількома функціями:

3. Клас Table

4. Клас Param який містить функції для меню і параметрів

2.3 Важливі фрагменти програми.

Функція обробки тексту:

```
1. public class Audit {
    public int[] how_much(StringBuffer s,int debug){
        int vowels=0,consonant=0,count=0;
        char[] a={'a','e','y','u','i','o'};
        char[]
b={'q','w','r','t','p','s','d','f','g','h','j','k','l','z','x','c','v','b','n','m',};
        for (int i =0;i<s.length();i++){
            for (int j=0;j<a.length;j++){
                if (a[j]==s.charAt(i)){
                    vowels++;
                }
            }
            for (int f=0; f<b.length;f++){
                if (b[f]==s.charAt(i)){
                    consonant++;
                }
            }
        }
        int[] letters = {vowels,consonant};
        if (debug == 1) {
            for (int k = 0; k < a.length; k++) {
                System.out.print(a[k] + " ");
            }
            for (int k = 0; k < b.length; k++) {
                System.out.print(b[k] + " ");
            }
        }
        return letters;
    }
}
```

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 StringBuffer. Також реалізовано можливість команд з боку користувача, а, крім цього, обробки командного рядка для визначення режиму роботи програми.