

Лабораторна робота №3

Утилітарні класи. Обробка масивів і рядків

Мета: Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

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. При вирішенні прикладних задач використовувати латинку.
3. Продемонструвати використання об'єктів класу `StringBuilder` або `StringBuffer`.
4. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. `Helper Class`) та для обробки даних використовувати відповідні статичні методи.
5. 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

- П.І.Б: Заночкин. Є. Д.
- Група: КІТ-119а
- Варіант: 7

1.2 Загальне завдання

Ввести текст. З тексту видалити всі слова заданої довжини, що починаються на приголосну літеру. Вивести початковий текст та результат.

2 ОПИС ПРОГРАМИ

2.1 Було використано наступні засоби:

`StringBuilder sbStr = new StringBuilder(str.substring(0))` – створення рядку типу `StringBuilder`;

`sbStr.length()` – визначення довжини рядка;

`sbStr.charAt()` – визначення символу, який стоїть на заданому місці.

2.2 Ієрархія та структура класів

Було створено 2 класи:

- `public class Main` – містить метод `main`, в якому задаються параметри, для роботи з класом `DeleteWords`;
- `public class DeleteWords` – клас для вирішення загального завдання.

2.3 Важливі фрагменти програми

```
package ua.khpi.oop.zanochkyn03;
```

```
/**
 * Клас DeleteWords, який видаляє всі слова заданої довжини,
 * що починаються на приголосну літеру.
 *
 * @author Zanochkyn Yehor
 * @version 1.0 22/10/20
 */
public class DeleteWords
{
    /**
     * Метод findAndDelete, який знаходить та видаляє потрібні слова
     */
    public static void findAndDelete(String str, int count)
    {
        StringBuilder sbStr = new StringBuilder(str.substring(0));
        System.out.println("Default string: " + sbStr);
        int endPos = 0;
        for (int i = 0; i < sbStr.length(); i++)
        {
            endPos = findPunctMark(sbStr,i);
            if (isVowel(sbStr.charAt(i)) == false)
            {
                StringBuilder sb = new StringBuilder(sbStr.substring(i, endPos));
                int endPos2 = findPunctMark(sb,0);
                if(endPos2 != -1)
                    sb.delete(endPos2,sb.length());
                if(i != 0)
                    if(sb.length() == count)
                    {
                        if(endPos2 != -1)
                            sbStr.delete(i-1, endPos-1);
                        else
                            sbStr.delete(i-1, endPos);
                        i = i - 1;
                    }
                    else
                        i = endPos;
                else
                    if(sb.length() == count)
                    {
                        if(endPos2 != -1)
                            sbStr.delete(i, endPos-1);
                        else
                            sbStr.delete(i, endPos+1);
                        i = i - 1;
                    }
                    else
                        i = endPos;
            }
            else
                i = endPos;
        }
        System.out.println("Edited string: " + sbStr);
    }
    /**
     * Метод isVowel, який перевіряє першу літеру слова
     */
    public static boolean isVowel(char c)
    {

```

```

switch (Character.toLowerCase(c))
{
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
    case 'y':
        return true;
    default:
        return false;
}
}

/**
 * Метод findPunctMark, який знаходить позицію, на якій є пунктуаційний знак
 */
public static int findPunctMark(StringBuilder a, int num)
{
    int endPos = a.indexOf(" ", num);
    if(endPos == -1)
        endPos = a.indexOf(".", num);
    if(endPos == -1)
        endPos = a.indexOf("!", num);
    if(endPos == -1)
        endPos = a.indexOf("?", num);
    if(endPos == -1)
        endPos = a.indexOf(",", num);
    if(endPos == -1)
        endPos = a.indexOf(";", num);
    if(endPos == -1)
        endPos = a.indexOf(":", num);
    if(endPos == -1)
        endPos = a.indexOf("-", num);
    if(endPos == -1)
        endPos = a.indexOf("\\", num);
    if(endPos == -1)
        endPos = a.indexOf("/", num);
    if(endPos == -1)
        endPos = a.indexOf("|", num);
    return endPos;
}
}

```

3 Результати роботи програми

```

Count letters in word to delete:
5
Default string: Welcome, to our - to welcome club, buddy. 12345 Welcome to our to welcome club, buddy.
Edited string: Welcome, to our - to welcome club,. Welcome to our to welcome club,.

```

Рисунок 3.1 – Результат роботи програми у середовищі Eclipse

Висновок

Під час виконання лабораторної роботи було набуто навички роботи з алгоритмами обробки масивів та рядків у середовищі Eclipse IDE.