

Звіт

Лабораторна робота 3. Утилітарні класи. Обробка масивів і рядків

Мета

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

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. Пропридемонструвати 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: Буй Зуї , КІТ119-а, варіант №26.

1.2. Ввести текст. У тексті слова заданої довжини замінити зазначеним рядком. Вивести початковий текст та результат.

2. ОПИС ПРОГРАМИ

2.1. Засоби ООП: клас, метод класу, поле класу.

2.2. Важливі фрагменти програми:

```
3. import java.text.BreakIterator;  
import java.util.ArrayList;  
import java.util.Scanner;  
import java.lang.StringBuilder;  
  
import java.util.*;  
public class HelperClass {
```

```

public void outputText(StringBuilder line)
{
    System.out.println(line);
}

public StringBuilder inputText()
{
    StringBuilder sb=new StringBuilder();
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter text: ");

    String inputText = scan.nextLine();
    scan.close();
    int temp = 0;

    for(int i =0 ;i<inputText.length();i++) {
        sb.append(inputText.charAt(i));
    }

    return sb;
}

public void getWords(String text,int k, String sentence) {
    List<String> words = new ArrayList<String>();
    BreakIterator breakIterator = BreakIterator.getWordInstance();
    breakIterator.setText(text);
    int lastIndex = breakIterator.first();
    while (BreakIterator.DONE != lastIndex) {
        int firstIndex = lastIndex;
        lastIndex = breakIterator.next();
        if (lastIndex != BreakIterator.DONE &&
            Character.isLetterOrDigit(text.charAt(firstIndex))) {
            words.add(text.substring(firstIndex, lastIndex));
        }
    }
    for(int i=0; i< words.size();i++){
        if(words.get(i).length() == k){
            words.set(i,sentence);
        }
        System.out.print(words.get(i)+" ");
    }
}

public static void main(String[] args) {
    int k = 4;
    String sen = ". ДУБ ЗЕЛЕННЫЙ ";

    HelperClass help = new HelperClass();
    StringBuilder text =help.inputText();
    String str = text.toString();
    help.outputText(text);

    help.getWords(str,k,sen);
}

```

```
}  
}
```

3.Результат

```
"C:\Program Files\AdoptOpenJDK\jdk-11.0.9.11-hotspot\bin\java.exe" "-javaagent:C:\Users\orrio\Downl  
Enter text: Мы вольные птицы пора брат Туда где за тучей белеет гора  
Мы вольные птицы пора брат Туда где за тучей белеет гора  
Мы вольные птицы . ДУБ ЗЕЛЕНИЙ , . ДУБ ЗЕЛЕНИЙ , . ДУБ ЗЕЛЕНИЙ , где за тучей белеет . ДУБ ЗЕЛЕНИЙ  
Process finished with exit code 0
```

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи с серіалізацією та десеріалізацією об'єктів .

Програма протестована, виконується без помилок.