## Звіт

**Лабораторна работа 3. Утилітарні класи. Обробка масивів і рядків**

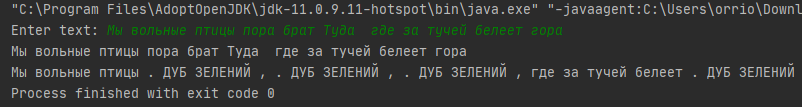
Мета

* 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](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%B0%D0%B1%D0%B5%D1%82%D0%BA%D0%B0).
3. Продемонструвати використання об'єктів класу [StringBuilder](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html) або [StringBuffer](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuffer.html).
4. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. [Helper Class](https://en.wikipedia.org/wiki/Helper_class)) та для обробки даних використовувати відповідні [статичні](https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html) методи.
5. Забороняється використовувати засоби обробки регулярних виразів: класи пакету [java.util.regex](https://docs.oracle.com/javase/8/docs/api/java/util/regex/package-summary.html) ([Pattern](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html), [Matcher](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html) та ін.), а також відповідні методи класу [String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) (matches, replace, replaceFirst, replaceAll, split).
   1. **Розробник**: Буй Зуі , КІТ119-а, варіант №26.
   2. Ввести текст. У тексті слова заданої довжини замінити зазначеним рядком. Вивести початковий текст та результат.
6. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: клас, метод класу, поле класу.
   2. **Важливі фрагменти програми:**
7. 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 ЗЕЛЕНИЙ ,";  
     
     
     
     
     
     
    HelperClass help = new HelperClass();  
    StringBuilder text =help.inputText();  
    String str = text.toString();  
    help.outputText(text);  
     
    help.getWords(str,k,sen);  
     
     
     
     
     
    }  
   }

**3.Результат**



**Висновки**

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

Програма протестована, виконується без помилок.