****Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

ЗВІТ   
про виконання лабораторної роботи №3  
з дисципліни «Технології конструювання програмного забезпечення»

**«Разработка и тестирование программ в среде IntelliJ IDEA»**

Виконав: студент групи TI-92

Черноусов Денис Ігорович

Перевірив: доцент, к.ф.-м.н.  
Тарнавський Ю.А.

Київ - 2020

Ход выполнения работы

1. Загрузите и установите IntelliJ IDEA.

2. Создайте проект Maven.

3. Напишите программу в соответствии с вариантом индивидуального задания.

4. Создайте unit-тесты для тестирования созданной программы.

5. Выполните тестирование программы.

**16 варіант**

5. Создайте программу для шифрования\расшифровки текста методом Цезаря. В нем ключом является целое число, а шифрование\расшифровка заключается в суммировании\ вычитании кодов символов открытого текста\криптотекста с ключом.

13.Создайте программу, которая осуществляет шифрование / расшифровка методом простой замены. В нем каждый символ незашифрованного текста из исходного алфавита заменяется другим из алфавита, символы в каком представлены в соответствии с ключом шифрования.

25.Анализ аргументов, задаваемых при запуске программы. Программа ищет наибольшую общую подстроку, содержащуюся во введенных аргументах (шаблон аргумента: строка либо латинских букв, либо букв кириллицы). Программа выводит количество заданных аргументов, их значения и наибольшую общую подстроку или сообщение о том, что общей подстроки нет.

31.Анализ аргументов, задаваемых при запуске программы. Программа сортирует введенные аргументы (шаблон аргумента: строка латинских букв) по первому символу аргумент. Программа выводит количество заданных параметров, их значения и список отсортированных аргументов.

Завдання 5

import static java.lang.Character.*isLetter*;  
  
public class task5 {  
  
 public static void main(String[] args){  
  
 String str = "HellOo there, I'm 18 !";  
  
 char[] input = str.toCharArray();  
  
 int key = -13 ;  
  
 System.*out*.printf("Encrypted : %n");  
  
 *Caesar*( key , input , true );  
  
 System.*out*.printf("%nDecrypted : %n");  
  
 *Caesar*( -key , input , true );  
 }  
  
 public static char[] Caesar(int key , char[] input , boolean bool ){  
  
 key = key % 26 ;  
  
 for(int i = 0 ; i < input.length ; i++){  
  
 if(*isLetter*(input[i])) {  
 if (input[i] + key < 'a' && Character.*isLowerCase*(input[i]))  
 input[i] = (char) (input[i] + key - 'a' + 'z' + 1);  
  
 else if (input[i] + key > 'z')  
 input[i] = (char) (input[i] + key + 'a' - 'z' - 1);  
  
 else if (input[i] + key < 'A')  
 input[i] = (char) (input[i] + key - 'A' + 'Z' + 1);  
  
 else if (input[i] + key > 'Z' && Character.*isUpperCase*(input[i]))  
 input[i] = (char) (input[i] + key + 'A' - 'Z' - 1);  
  
 else input[i] += key;  
 }  
  
 *withNum*(key , input , bool , i ) ;  
 }  
 System.*out*.println();  
 return input ;  
 }  
  
 public static void withNum(int key ,char[] input , boolean bool , int i){  
 int numkey = key % 11 ;  
 if (bool){  
 if(Character.*isDigit*(input[i])&& input[i] + numkey > '9' )  
 input[i] = (char)( input[i] + numkey - '9' + '0' - 1 ) ;  
  
 else if(Character.*isDigit*(input[i])&& input[i] + numkey < '0' )  
 input[i] = (char)( input[i] + numkey + '9' - '0' + 1 ) ;  
  
 else if(Character.*isDigit*(input[i])) input[i] += numkey ;  
  
 }  
 System.*out*.print(input[i]);  
 }  
}

Вивід:

Encrypted :

UryyBb gurer, V'z 96 !

Decrypted :

HellOo there, I'm 18 !

Завдання 13

public class task13 {  
  
 public static char[] changer ( char[] str , String abcStr , String keyStr ){  
  
 char[] abc = abcStr.toCharArray();  
 char[] key = keyStr.toCharArray();  
  
 for(int j = 0 ; j < str.length ; j++){  
 for(int i = 0 ; i < abc.length ; i++){  
  
 if ( str[j] == abc[i] ){  
 str[j] = key[i] ;  
 break;  
 }  
  
 if ( str[j] == Character.*toUpperCase*(abc[i]) ){  
 str[j] = Character.*toUpperCase*(key[i]) ;  
 break;  
 }  
 }  
 System.*out*.print(str[j]) ;  
 }  
  
 System.*out*.println();  
 return str ;  
 }  
  
 public static void main(String[] args){  
  
 String abcStr = "abcdefghijklmnopqrstuvwxyz" ;  
 String keyStr = "zyxcbawvutsrqponmlkjihgfed" ;  
  
 String myStr = "I'm going to feel";  
 char[] str = myStr.toCharArray();  
  
 *changer*( str , abcStr , keyStr );  
  
 *changer*( str , keyStr , abcStr );  
  
 }  
}

Вивід :

Encrypted :

U'q woupw jo abbr

Decrypted :

I'm going to feel

Додатковий файл:

public class Out {  
 public static void outArgs(String[] args){  
 System.*out*.println( "There are " + args.length + " arguments : ");  
 for (int i = 0; i < args.length; i++) System.*out*.println( args[i] );  
  
 System.*out*.println();  
  
 }  
}

Завдання 25

public class task25 {  
  
 public static String similar(String s1 , String s2 ){  
  
 if (s1.length() > s2.length()) { // s1 має бути меншим  
 String temp = s1 ;  
 s1 = s2 ;  
 s2 = temp ;  
 }  
  
 String temp = "" ;  
  
 for (int i = s1.length() ; i > 1 ; i--){ // мінімальна допустима довжина 2 substring’a  
 for ( int j = 0 ; j < s1.length() ; j++ )  
 for ( int k = 0 ; k < s2.length() ; k++ )  
 if( s1.regionMatches( true , j , s2 , k , i )) {  
 temp = String.*copyValueOf*( s1.toCharArray(), j , i );  
 return temp;  
 }  
  
 }  
 temp = null ;  
 return temp;  
 }  
  
 public static String findSubString(String[] args){  
 String max = "" ;  
  
 for(int i = 0 ; i < args.length - 1 ; i++){  
 if (max == null){  
 System.*out*.println("There isn't mutual substring!");  
 break ;  
 }  
 max = *similar* (args[i] , args[i+1] ); // знаходимо спільне в двох словах  
 args[i+1] = max ;  
 }  
 System.*out*.println("Mutual substring : " + max);  
 return max ;  
 }  
  
 public static void main(String[] args) {  
  
 Out.*outArgs*( args );  
  
 *findSubString*( args ) ;  
  
 }  
}

Вивід:

D:\Objects\JavaLab\lab3\target\classes>java task25 There are not substrings

There are 4 arguments :

There

are

not

substring

There isn't mutual substring!

Mutual substring : null

D:\Objects\JavaLab\lab3\target\classes>java task25 Second section gensec

There are 3 arguments :

Second

section

gensec

Mutual substring : Sec

Завдання 31

import static java.lang.reflect.Array.*getChar*;  
  
public class task31 {  
  
 public static String Sort(String[] args) {  
  
 String temp = " " ;  
 for(int j = 0 ; j < args.length - 1 ; j++ )  
 for (int i = 0 ; i < args.length - 1 ; i++){  
  
 if( (int)Character.*toLowerCase*(*getChar*( args[i].toCharArray() ,0 )) - (int)Character.*toLowerCase*(*getChar*( args[i+1].toCharArray() ,0 )) > 0 ){  
 temp = args[i] ;  
 args[i] = args[i+1];  
 args[i+1] = temp ;  
 }  
 }  
 System.*out*.println("Sorted : ");  
 Out.*outArgs*(args) ;  
 return args[0] ;  
 }  
  
 public static void main(String[] args) {  
  
 Out.*outArgs*(args) ;  
  
 *Sort* ( args ) ;  
  
 }  
}

Вивід :

D:\Objects\JavaLab\lab3\target\classes>java task31 Task said to order by first characters

There are 7 arguments :

Task

said

to

order

by

first

characters

Sorted :

There are 7 arguments :

by

characters

first

order

said

Task

To

**Unit тести**

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class taskTest {  
  
 @Test  
 public void task5test(){  
 task5 tester = new task5() ;  
 char [] actuall = tester.*Caesar*( -13 , "HellOo there, I'm 18 !".toCharArray() , true) ;  
 char [] expected = "UryyBb gurer, V'z 96 !".toCharArray();  
 *assertArrayEquals*( expected , actuall );  
 }  
  
 @Test  
 public void task13test(){  
 task13 tester = new task13();  
 char[] actuall = tester.*changer* ("I'm going to feel".toCharArray() , "abcdefghijklmnopqrstuvwxyz" , "zyxcbawvutsrqponmlkjihgfed" );  
 char[] expected = "U'q woupw jo abbr".toCharArray() ;  
 *assertArrayEquals*( expected , actuall );  
 }  
  
 @Test  
 public void task25test(){  
 task25 tester = new task25();  
 String [] three = {"Second", "section" , "Tesec" } ;  
 String actuall = tester.*findSubString*( three );  
 String expected = "Sec" ;  
 *assertEquals*( expected , actuall );  
 }  
  
 @Test  
 public void task31test(){  
 task31 tester = new task31();  
 String [] three = {"wrestler" , "Big", "My" , "Fellow " } ;  
 String actuall = tester.*Sort*( three );  
 String expected = "Big" ;  
 *assertEquals*( expected , actuall );  
 }  
  
}

Вивід :

