Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №1  
з дисципліни «Технології Java конструювання програмного забезпечення»

«**РОБОТА З РЯДКАМИ І РЕГУЛЯРНИМИ ВИРАЗАМИ**»

Варіант 17

Виконав: студент групи ТР-23

Ровний Г.О.

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Завдання 1

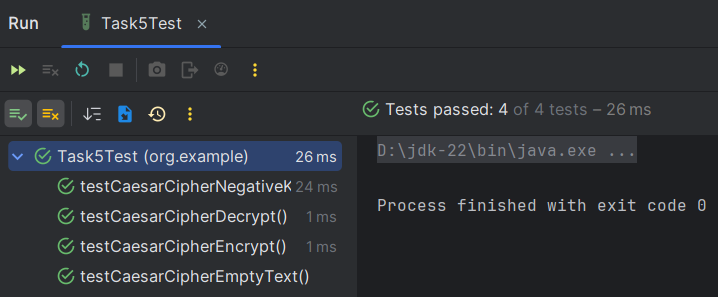
Код:

package org.example;  
  
import java.util.Scanner;  
  
public class Task5 {  
  
 public static String caesarCipher(String text, int key) {  
 StringBuilder result = new StringBuilder();  
 for (char character : text.toCharArray()) {  
 char shiftedChar = (char) (character + key);  
 result.append(shiftedChar);  
 }  
  
 return result.toString();  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть текст: ");  
 String text = scanner.nextLine();  
 System.*out*.print("Введіть ключ (ціле число): ");  
 int key = scanner.nextInt();  
 System.*out*.print("Оберіть 1 - шифрувати, 2 - розшифрувати): ");  
 int choice = scanner.nextInt();  
  
 if (choice == 1) {  
 String encryptedText = *caesarCipher*(text, key);  
 System.*out*.println("Зашифрований текст: " + encryptedText);  
 } else if (choice == 2) {  
 // Розшифруємо текст  
 String decryptedText = *caesarCipher*(text, -key);  
 System.*out*.println("Розшифрований текст: " + decryptedText);  
 } else {  
 System.*out*.println("Невірний вибір дії.");  
 }  
  
 scanner.close();  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task5Test {  
  
 @Test  
 void testCaesarCipherEncrypt() {  
 String text = "ABCD";  
 int key = 3;  
 String expected = "DEFG"; // Зашифрований текст  
 String result = Task5.*caesarCipher*(text, key);  
 *assertEquals*(expected, result);  
 }  
  
 @Test  
 void testCaesarCipherDecrypt() {  
 String text = "DEFG";  
 int key = 3;  
 String expected = "ABCD";  
 String result = Task5.*caesarCipher*(text, -key);  
 *assertEquals*(expected, result);  
 }  
  
 @Test  
 void testCaesarCipherNegativeKey() {  
 String text = "DEFG";  
 int key = -3;  
 String expected = "ABCD" ;  
 String result = Task5.*caesarCipher*(text, key);  
 *assertEquals*(expected, result);  
 }  
  
 @Test  
 void testCaesarCipherEmptyText() {  
 String text = "";  
 int key = 5;  
 String expected = "";  
 String result = Task5.*caesarCipher*(text, key);  
 *assertEquals*(expected, result);  
 }  
}

Скріншот:



# Завдання 2

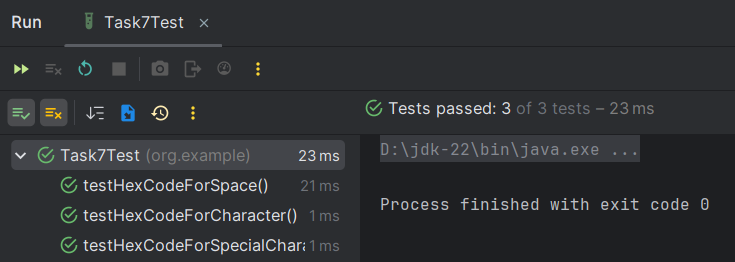
Код:

package org.example;  
  
import java.util.Scanner;  
  
public class Task7 {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 StringBuilder hexCodes = new StringBuilder();  
 System.*out*.println("Введіть символи. Для завершення введіть пробіл.");  
 while (true) {  
 String input = scanner.nextLine();  
 if (input.equals(" ")) {  
 break;  
 }  
 for (char inputChar : input.toCharArray()) {  
 String hexCode = String.*format*("\\u%04X", (int) inputChar);  
 hexCodes.append(hexCode).append(" ");  
 }  
 }  
 System.*out*.println("Коди символів у шістнадцятковому представленні: " + hexCodes.toString().trim());  
 scanner.close();  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task7Test {  
  
 @Test  
 void testHexCodeForCharacter() {  
 char inputChar = 'a';  
 String expectedHexCode = "\\u0061";  
 String result = String.*format*("\\u%04X", (int) inputChar);  
 *assertEquals*(expectedHexCode, result);  
 }  
  
 @Test  
 void testHexCodeForSpace() {  
 char inputChar = ' ';  
 String expectedHexCode = "\\u0020";  
 String result = String.*format*("\\u%04X", (int) inputChar);  
 *assertEquals*(expectedHexCode, result);  
 }  
  
 @Test  
 void testHexCodeForSpecialCharacter() {  
 char inputChar = '@';  
 String expectedHexCode = "\\u0040";  
 String result = String.*format*("\\u%04X", (int) inputChar);  
 *assertEquals*(expectedHexCode, result);  
 }  
}

Скріншот:



# Завдання 3

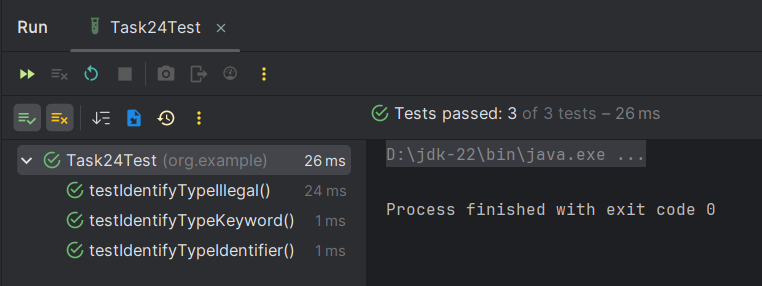
Код:

package org.example;  
  
import java.util.HashSet;  
import java.util.Set;  
  
public class Task24 {  
 private static final Set<String> *JAVA\_KEYWORDS* = new HashSet<>();  
  
 static {  
 *JAVA\_KEYWORDS*.add("if");  
 *JAVA\_KEYWORDS*.add("for");  
 *JAVA\_KEYWORDS*.add("while");  
 *JAVA\_KEYWORDS*.add("do");  
 *JAVA\_KEYWORDS*.add("else");  
 // Додайте інші ключові слова за потреби  
 }  
  
 public static String identifyType(String arg) {  
 if (*JAVA\_KEYWORDS*.contains(arg)) {  
 return "Keyword";  
 } else if (*isValidIdentifier*(arg)) {  
 return "Identifier";  
 } else {  
 return "Illegal";  
 }  
 }  
  
 public static boolean isValidIdentifier(String arg) {  
 // Перевірка, чи є аргумент валідним ідентифікатором  
 if (arg.isEmpty() || !Character.*isJavaIdentifierStart*(arg.charAt(0))) {  
 return false;  
 }  
  
 for (char ch : arg.toCharArray()) {  
 if (!Character.*isJavaIdentifierPart*(ch)) {  
 return false;  
 }  
 }  
  
 return true;  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("Кількість аргументів: " + args.length);  
  
 for (String arg : args) {  
 String type = *identifyType*(arg);  
 System.*out*.println("Аргумент: \"" + arg + "\" - Тип: " + type);  
 }  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task24Test {  
  
 @Test  
 void testIdentifyTypeKeyword() {  
 *assertEquals*("Keyword", Task24.*identifyType*("if"));  
 *assertEquals*("Keyword", Task24.*identifyType*("for"));  
 *assertEquals*("Keyword", Task24.*identifyType*("while"));  
 }  
  
 @Test  
 void testIdentifyTypeIdentifier() {  
 *assertEquals*("Identifier", Task24.*identifyType*("variable"));  
 *assertEquals*("Identifier", Task24.*identifyType*("\_myVar"));  
 *assertEquals*("Identifier", Task24.*identifyType*("$myVar"));  
 *assertEquals*("Identifier", Task24.*identifyType*("var123"));  
 }  
  
 @Test  
 void testIdentifyTypeIllegal() {  
 *assertEquals*("Illegal", Task24.*identifyType*("123abc"));  
 *assertEquals*("Illegal", Task24.*identifyType*("!invalid"));  
 *assertEquals*("Illegal", Task24.*identifyType*("var!able"));  
 *assertEquals*("Illegal", Task24.*identifyType*(""));  
 }  
}

Скріншот:



# Завдання 4

Код:

package org.example;  
  
public class Task32 {  
  
 public static boolean isInteger(String arg) {  
 if (arg.isEmpty()) {  
 return false;  
 }  
 int startIndex = 0;  
 if (arg.charAt(0) == '+' || arg.charAt(0) == '-') {  
 startIndex = 1;  
 }  
 for (int i = startIndex; i < arg.length(); i++) {  
 if (!Character.*isDigit*(arg.charAt(i))) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 public static void main(String[] args) {  
 int count = args.length;  
 int numberCount = 0;  
 Integer maxNumber = null;  
  
 System.*out*.println("Кількість аргументів: " + count);  
  
 for (String arg : args) {  
 System.*out*.println("Аргумент: \"" + arg + "\"");  
  
 if (*isInteger*(arg)) {  
 numberCount++;  
 int number = Integer.*parseInt*(arg);  
 if (maxNumber == null || number > maxNumber) {  
 maxNumber = number;  
 }  
 }  
 }  
 System.*out*.println("Кількість аргументів-чисел: " + numberCount);  
 if (maxNumber != null) {  
 System.*out*.println("Максимальне число: " + maxNumber);  
 } else {  
 System.*out*.println("Не знайдено жодного аргументу-числа.");  
 }  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task32Test {  
  
 @Test  
 void testIsInteger() {  
 // Правильні цілі числа  
 *assertTrue*(Task32.*isInteger*("0"), "0 is a valid integer.");  
 *assertTrue*(Task32.*isInteger*("123"), "123 is a valid integer.");  
 *assertTrue*(Task32.*isInteger*("-123"), "-123 is a valid integer.");  
 *assertTrue*(Task32.*isInteger*("+123"), "+123 is a valid integer.");  
  
 //Неправильні цілі числа  
 *assertFalse*(Task32.*isInteger*("12.3"), "12.3 is not a valid integer.");  
 *assertFalse*(Task32.*isInteger*("abc"), "abc is not a valid integer.");  
 *assertFalse*(Task32.*isInteger*("12a3"), "12a3 is not a valid integer.");  
 *assertFalse*(Task32.*isInteger*(""), "Empty string is not a valid integer.");  
  
 //Неправильний формат  
 *assertFalse*(Task32.*isInteger*(" "), "Space is not a valid integer.");  
 }  
}

Скріншот:

