|  |
| --- |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное  бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий

Кафедра корпоративных информационных систем

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **по лабораторной работе №2** | | |
| **по дисциплине** | | |
| **«Объектно-ориентированное программирование»**  **Тема лабораторной работы: «**Строки в Java. Применение регулярных выражений  для поиска заданных слов.**»** | | |
| Студент группы | ИКБО-07-18 | Бабух А.А. |
| Принял | ассистент кафедры | Колесников С.Н. |
|  |  |  |
| Выполнено | «\_\_\_» \_\_\_\_\_\_ 201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |  | *(подпись студента)* |
| Зачтено | «\_\_\_» \_\_\_\_\_\_ 201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |  | *(подпись преподавателя)* |

1. **Постановка задачи**
   1. **Задание №1**

Применение класса String.

* + - 1. Создать приложение - основную программу, которое должно иметь интерфейс для демонстрации работы всех трех заданий (два последних пункта будут реализовываться по мере отладки предыдущих). В каждом пункте создаются свои объекты, для решения задач задания.

1. Создать класс с именем, определяющим номер задания, для решения задач варианта.
   1. Данными класса должен быть входной текст и дополнительные строки для промежуточных вычислений.
   2. Основные операции указаны в варианте. Их надо оформить методами созданного класса. Если операция содержит отдельную задачу, то ее так же оформить отдельным методом.
   3. При разработке методов класса использовать объект и методы класса String.
2. Выполнить тестирование класса и приложения.

Вариант №2: Дан код программы на языке С и набор из нескольких ключевых слов языка.

* Сформировать строку из ключевых слов языка, которые используются в тексте. Слова должны разделяться одним пробелом.
* Первые строчные символы в словах сформированного массива заменить прописными.
* Сформировать массив, который хранит количество вхождений каждого ключевого слова в код программы.
* Сформировать строку из пользовательских идентификаторов.
  1. **Задание №2**

1. Разработать класс для решения задачи варианта.
   1. Данными класса должен быть входной текст и дополнительные строки для промежуточных вычислений.
   2. Основные операции указаны в варианте. Их надо оформить методами созданного класса. Если операция содержит отдельную задачу, то ее так же оформить отдельным методом.
   3. При разработке методов своего класса использовать объекты и методы класса StringBuffer.
2. Реализовать второй пункт консольного интерфейса приложения, созданного в задании 1, включив алгоритм демонстрации выполнения задач варианта.
3. Выполнить тестирование класса и приложения.

Вариант №2: Дан код программы на языке С и набор из нескольких ключевых слов языка.

* Сформировать строку из пользовательских идентификаторов левой части оператора присваивания.
* Преобразовать строку: оставить только ключевые слова языка, которые используются в тексте. Слова должны разделяться одним пробелом.
* Вставить новое ключевое слово, которого нет в строке, перед первым словом.
  1. **Задание №3**

Регулярные выражения — язык для построения шаблонов поиска и осуществления манипуляций с строками в тексте. Позволяет выполнять

1. Разработать класс для решения задач варианта.

2. Разработать регулярное выражение в соответствии с первой задачей варианта. Разработать метод, реализующий проверку строки на соответствие регулярному выражению и протестировать метод.

3. Разработать метод для решения второй задачи варианта. Выполнить его тестирование.

4. Реализовать третий пункт консольного интерфейса приложения, созданного в задании 1, включив алгоритм демонстрации выполнения задач варианта.

Вариант №2:

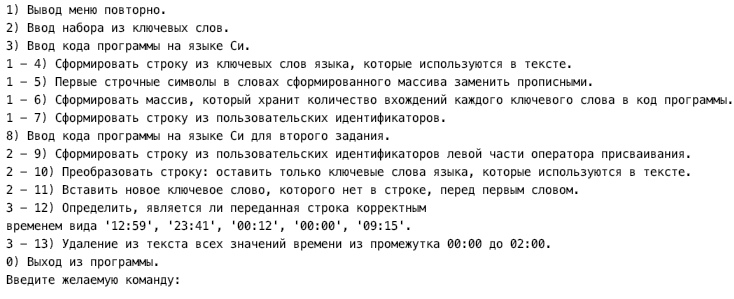
1. Определите, что переданная строка является корректным временем вида '12:59', '23:41', '00:12', '00:00', '09:15'. Время '24.00', '25.00', '12.60', '12.93', '41.93' является некорректным.

2. Удалите из текста все значения времени из промежутка 00:00 до 02:00.

1. **Описание используемых классов**

* Main.java: находится интерфейс программы;
* First.java: находятся метод формирования строки из ключевых слов, метод замены первых строчных символов прописными, метод формирования массива, хранящего количество вхождений каждого слова в код программы, и метод формирования строки из пользовательских идентификаторов.
* Second.java: находятся метод формирования строки из идентификаторов, метод преобразования строки, оставляющий только ключевые слова языка, метод вставки нового ключевого слова.
* Third.java: метод сравнения переданной строки с корректным временем и метод удаления из текста всех значений времени заданного промежутка.

1. **Пользовательский интерфейс**

Пользователь имеет возможность выполнения всех операций, предложенных программой. (Смотреть Рисунок №1)

**Рисунок 1 –** Пользовательский интерфейс

1. **Листинг программы**

**Main.java**

import java.util.HashSet;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

int comm = -1;

Scanner in = new Scanner(System.in);

outputMenu();

First first = new First();

Second second = new Second();

Third third = new Third();

while(comm != 0){

System.out.print("Введите желаемую команду: ");

comm = in.nextInt();

switch (comm){

case 1:

outputMenu();

break;

case 2:{

System.out.print("Введите набор из нескольких ключевых слов языка Си: ");

Scanner in2 = new Scanner(System.in);

String line = in2.nextLine();

HashSet<String> keys = new HashSet<String>();

for(String key: line.split(" "))

keys.add(key);

first.setKeys(keys);

second.setKeys(keys);

}

break;

case 3: {

Scanner in2 = new Scanner(System.in);

System.out.print("Введите код программы на языке Си: ");

first.setText(in2.nextLine());

}

break;

case 4:

first.showEntrys();

break;

case 5:

first.toUpper();

break;

case 6:

first.countEntrys();

break;

case 7:

first.getIdentificators();

break;

case 8: {

Scanner in2 = new Scanner(System.in);

System.out.print("Введите текст: ");

second.setText(new StringBuilder(in2.nextLine()));

}

break;

case 9:

second.identificator();

break;

case 10:

second.delIndent();

break;

case 11:

second.insertNew();

break;

case 12:{

Scanner in2 = new Scanner(System.in);

System.out.print("Введите текст: ");

third.setText(in2.nextLine());

third.checkNum();

}

break;

case 13: {

Scanner in2 = new Scanner(System.in);

System.out.print("Введите текст: ");

third.setText(in2.nextLine());

third.delNum();

}

break;

case 0:

System.out.println("Программа завершена.");

break;

default:

System.out.println("Введена неверная команда, пожалуйста, повторите ввод!");

break;

}

}

}

public static void outputMenu(){

System.out.println("1) Вывод меню повторно.");

System.out.println("2) Ввод набора из ключевых слов.");

System.out.println("3) Ввод кода программы на языке Си.");

System.out.println("1 - 4) Сформировать строку из ключевых слов языка, которые используются в тексте.");

System.out.println("1 - 5) Первые строчные символы в словах сформированного массива заменить прописными.");

System.out.println("1 - 6) Сформировать массив, который хранит количество вхождений каждого ключевого слова в код программы.");

System.out.println("1 - 7) Сформировать строку из пользовательских идентификаторов.");

System.out.println("8) Ввод кода программы на языке Си для второго задания.");

System.out.println("2 - 9) Сформировать строку из пользовательских идентификаторов левой части оператора присваивания.");

System.out.println("2 - 10) Преобразовать строку: оставить только ключевые слова языка, которые используются в тексте.");

System.out.println("2 - 11) Вставить новое ключевое слово, которого нет в строке, перед первым словом.");

System.out.println("3 - 12) Определить, является ли переданная строка корректным\n" +

"временем вида '12:59', '23:41', '00:12', '00:00', '09:15'.");

System.out.println("3 - 13) Удаление из текста всех значений времени из промежутка 00:00 до 02:00.");

System.out.println("0) Выход из программы.");

}

}

**First.java**

package com.company;

import java.util.\*;

public class First {

String text;

HashSet<String> keys;

HashSet<String> types;

HashSet<String> typeCreator;

HashSet<String> userTypes;

LinkedHashSet<String> words= new LinkedHashSet<String>();

LinkedHashMap<String, Integer> count = new LinkedHashMap<String, Integer>();

private final int shift = 'A' - 'a';

First(){

this.text = "";

this.keys = new HashSet<String>();

types = new HashSet<String>();

typeCreator = new HashSet<String>();

userTypes = new HashSet<String>();

types.add("float");

types.add("int");

types.add("double");

types.add("char");

types.add("bool");

typeCreator.add("struct");

}

First(String text, HashSet<String> keys){

this.text = text;

this.keys = keys;

types = new HashSet<String>();

typeCreator = new HashSet<String>();

userTypes = new HashSet<String>();

types.add("float");

types.add("int");

types.add("double");

types.add("char");

types.add("bool");

typeCreator.add("struct");

}

public void setText(String text) {

this.text = text;

}

public void setKeys(HashSet<String> keys) {

this.keys = keys;

}

//Сформировать строку из ключевых слов языка, которые используются в тексте.

public void showEntrys(){

if(text == ""){

System.out.println("Инициализация не произведена."); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена."); return;

}

words.clear();

for(String word:text.split("[ \n]")){

if(keys.contains(word))

words.add(word);

}

Iterator<String> it = words.iterator();

System.out.print("Ключевые слова, встречающиеся в тексте: ");

while (it.hasNext()){

System.out.print(" " + it.next());

}

System.out.println();

}

//Первые строчные символы в словах сформированного массива заменить прописными.

public void toUpper(){

if(text == ""){

System.out.println("Инициализация не произведена."); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена."); return;

}

if(words.size() == 0){

System.out.println("Массив не сформирован."); return;

}

ArrayList<String> tmp = new ArrayList<String>(0);

Iterator<String> it = words.iterator();

System.out.print("Ключевые слова, встречающиеся в тексте с заглавной буквы: ");

while (it.hasNext()){

String word = it.next();

word = (char)(word.charAt(0)+shift) + word.substring(1);

System.out.print(" " + word);

tmp.add(word);

}

words.clear();

words.addAll(tmp);

System.out.println();

}

//Сформировать массив, который хранит количество вхождений каждого ключевого слова в код программы.

public void countEntrys(){

count.clear();

if(text == ""){

System.out.println("Инициализация не произведена."); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена."); return;

}

words.clear();

for(String word:text.split("[ \n]")){

if(keys.contains(word)){

if(count.containsKey(word))

count.put(word, count.get(word)+1);

else

count.put(word, 1);

}

}

List<String> keysIn = new ArrayList<String>(count.keySet());

System.out.print("Ключевые слова, встречающиеся в тексте и количество их вхождений: ");

for(String key: keysIn)

System.out.print(" " + key + " - " + count.get(key) + ",");

System.out.println();

}

//Сформировать строку из пользовательских идентификаторов.

public void getIdentificators(){

userTypes.clear();

String[] arr = text.split("[ \n]");

String ans = "";

for(int i = 0; i < arr.length-1; i++){

if(types.contains(arr[i]))

ans += arr[i+1].replaceAll(";", "") + " ";

else

if(typeCreator.contains(arr[i])){

userTypes.add(arr[i+1]);

ans += arr[i+1].replaceAll(";", "") + " ";

i++;

}

else{

if(userTypes.contains(arr[i]))

ans += arr[i+1].replaceAll(";", "") + " ";

}

}

System.out.print("Пользовательские идентификаторы: ");

System.out.println(ans);

text = ans;

}

}

**Second.java**

package com.company;

import java.util.HashSet;

import java.util.LinkedHashSet;

public class Second {

StringBuilder text;

HashSet<String> keys;

Second(){

this.text = new StringBuilder();

this.keys = new HashSet<String>();

}

Second(StringBuilder text, HashSet<String> keys){

this.text = text;

this.keys = keys;

}

public void setKeys(HashSet<String> keys) {

this.keys = keys;

}

public void setText(StringBuilder text) {

this.text = text;

}

//Сформировать строку из пользовательских идентификаторов левой части оператора присваивания.

public void identificator(){

if(text.length() == 0){

System.out.println("Инициализация не произведена"); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена"); return;

}

StringBuilder ans = new StringBuilder();

int pos = text.indexOf("=");

while(pos != -1){

int spacePos = pos - 2;

while(spacePos >= 0 && text.charAt(spacePos) != ' ')

spacePos--;

if(!keys.contains(text.substring(spacePos+1, pos-1)))

ans.append(text.substring(spacePos+1, pos-1) + " ");

pos = text.indexOf("=", pos+1);

}

ans.deleteCharAt(ans.length()-1);

System.out.print("Идентификаторы: ");

System.out.println(ans);

}

//Преобразовать строку: оставить только ключевые слова языка, которые используются в тексте.

public void delIndent(){

if(text.length() == 0){

System.out.println("Инициализация не произведена"); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена"); return;

}

StringBuilder ans = new StringBuilder();

int pos = text.indexOf(" ");

while(pos != -1){

int spacePos = pos - 1;

while(spacePos >= 0 && text.charAt(spacePos) != ' ')

spacePos--;

String word = text.substring(spacePos+1, pos);

if(!keys.contains(word))

ans.append(text.substring(spacePos+1, pos) + " ");

pos = text.indexOf(" ", pos+1);

}

if (ans.length() > 0)

ans.deleteCharAt(ans.length()-1);

text = ans;

System.out.print("Текст без идентификаторов: ");

System.out.println(text);

}

//Вставить новое ключевое слово, которого нет в строке, перед первым словом.

public void insertNew(){

if(text.length() == 0){

System.out.println("Инициализация не произведена."); return;

}

if(keys.size() == 0){

System.out.println("Инициализация не произведена."); return;

}

HashSet<String> unused = (HashSet<String>)keys.clone();

int pos = text.indexOf(" ");

while(pos != -1){

int spacePos = pos - 1;

while(spacePos >= 0 && text.charAt(spacePos) != ' ')

spacePos--;

String word = text.substring(spacePos+1, pos);

if(unused.contains(word))

unused.remove(word);

pos = text.indexOf(" ", pos+1);

}

if (unused.size() > 0){

text.insert(0, unused.iterator().next());

}

System.out.print("Изменённый текст: ");

System.out.println(text);

}

}

**Third.java**

package com.company;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Third {

String text;

Third(){

this.text = "";

}

Third(String text){

this.text = text;

}

public void setText(String text) {

this.text = text;

}

//Определить, является ли переданная строка корректным временем вида '12:59', '23:41', '00:12', '00:00', '09:15'.

public void checkNum(){

if(text == ""){

System.out.println("Инициализация не произведена."); return;

}

Pattern pattern = Pattern.compile("[01]\\d:[0-5]\\d|2[0-3]:[0-5]\\d");

if(pattern.matcher(text).matches())

System.out.println("Строка является правильным представлением времени.");

else

System.out.println("Строка не является правильным представлением времени.");

}

//Удаление из текста всех значений времени из промежутка 00:00 до 02:00.

public void delNum(){

if(text == ""){

System.out.println("Инициализация не произведена."); return;

}

text = text.replaceAll("0[01]:[0-5]\\d\\s?|02:00\\s?", "");

if(text.length() != 0 && text.charAt(text.length()-1) == ' ')

text = text.substring(0, text.length()-1);

System.out.println("Строка без времени от 00:00 до 02:00 - " + text);

}

}