Федеральное агентство связи

Ордена Трудового Красного знамени

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

«Московский технический университет связи и информатики

Кафедра «Математической кибернетики и информационных технологий»

Отчёт по лабораторной работе №2

Проектирование лексического анализатора

Выполнил:

Студент группы БФИ 1701

Сизов А.Д

Проверила:

Ассистент кафедры МКиИТ

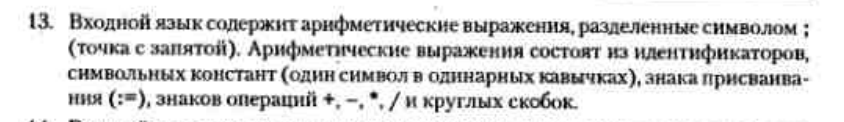
Алексанян Диана Ашотовна

https://github.com/AragonRyzen/SPO\_Labr1.git

Вариант 13

Москва 2020г.

**Задание**



Код класса Main.java

|  |
| --- |
| import java.io.IOException; import java.io.File; import java.nio.file.Files; import java.nio.file.Paths; import java.util.regex.\*; import java.io.\*;   public class Main {    public static void main(String[] args) {    String fileread = "t.txt";  String txt = "";  String filewrite = "rez.txt";   File file= new File("rez.txt");  BufferedWriter writer = null;   try {  txt = *readUsingFiles*(fileread);  writer = new BufferedWriter(new FileWriter(file));  } catch (IOException e) {  e.printStackTrace();  }   String txt2 = "";  txt2 += txt.replaceAll("\\r\\n", "");  System.*out*.println(txt2);  txt = txt2.replaceAll(" ", "");  System.*out*.println(txt);  txt2 = txt;   String result = "dfdf";    while (txt2.equals("")==false ) {   if (txt2.matches("^\\+(.\*)") == true) {  result = "+ - Знак сложения \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^-(.\*)") == true) {  result = "- - Знак вычитания\n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^:=(.\*)") == true) {  result = ":= - Знак присваивания \n";  txt2 = txt2.substring(2);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^\\\*(.\*)") == true) {  result = "\* - Знак умножения \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^/(.\*)") == true) {  result = "/ - Знак деления \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^;(.\*)") == true) {  result = "; - Точка с запятой \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^\\((.\*)") == true) {  result = "( - Открывающая скобка \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  if (txt2.matches("^\\)(.\*)") == true) {  result = ") - Закрывающая скобка \n";  txt2 = txt2.substring(1);  *Writein*(result,writer);  result = "";  continue;  }  //СТАТИЧЕСКИЕ  if (txt2.matches("^'[a**-**zA**-**Z]+'(.\*)") == true) {  Pattern p = Pattern.*compile*("^'[a**-**zA**-**Z]+'");  Matcher m = p.matcher(txt2);  while (m.find()){  result = txt2.substring(m.start(), m.end()) + " - Константное значение \n";  txt2 = txt2.substring(m.end());  }    *Writein*(result,writer);  result = "";   continue;  }   //ИДЕНТИФИКАТОРЫ  if (txt2.matches("[a**-**zA**-**Z]+(.\*)") == true) {  Pattern p = Pattern.*compile*("^[a**-**zA**-**Z]+");  Matcher m = p.matcher(txt2);  while (m.find()){  result = txt2.substring(m.start(), m.end()) + " - Идентификатор \n";  txt2 = txt2.substring(m.end());  }    *Writein*(result,writer);  result = "";   continue;  }    }  try {  writer.close();}  catch (Exception e){System.*out*.println("Не удалось закрыть файл");}  }   private static void Writein (String str, BufferedWriter fw)  {  try {  fw.write(str);  fw.flush();   } catch (Exception e) {  System.*out*.println("Ошибка записи");  }   }    private static String readUsingFiles (String filename) throws IOException {  return new String(Files.*readAllBytes*(Paths.*get*(filename)));  }    } |

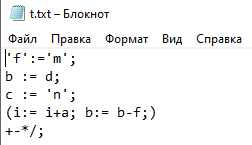


Рисунок – Входные данные

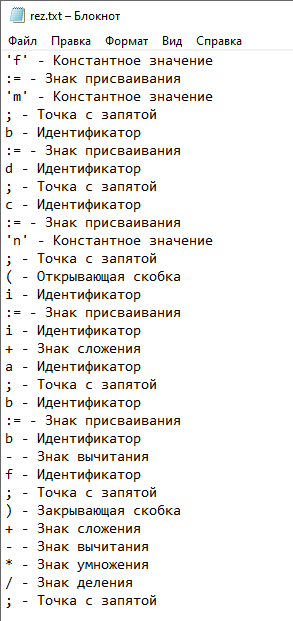


Рисунок – Результат