Regular expression in Java (8 points)

Due date: In the labs of Jan 19/22.

How to submit

Submit to our marking program at our submission site .

Purpose

Warm-up with Java programming. Get familiar with regular expression. Learn to read grammar of a language. Understand the wide application of regular expression.

Assignment specification

Your job is to count the number of identifiers in programs written in our Tiny language. Click on the link for identifiers for the explanation for identifier.
You will write a method that pick out the identifiers from a text file. Here are the sample input . Your method should return a set of identifers that consists of {f2, x,y,z,F1}. Please note that in this sample program the following are not counted as identifiers:

- A41, input, output: they are quoted hence they are not treated as identifiers;
- INT, READ etc.: They are keywords used in our Tiny language hence they should not be picked up.

Here are the test cases for the assignment: case 1, case 2, case 3, case 4, case 5, case 6. (ID counts: 5 4 6 7 8 9).
In this assignment you can suppose that there are no comments in the input programsi writtin in TINY language.

You will write two different programs to do this:

1. Program A11.java is not supposed to use regular expressions, not regex package, not the methods involvoing regular expression in String class or other classes. Your program should use the most primitive method, i.e. look at characters one by one, and write a loop to check whether they are quoted strings, identifiers, etc.
2. Program A12.java will use java.util.regex. One useful link to start with is a tutorial for Java regex.

Your programs should be able to run by typing:

```
%javac A11.java
%java A11  A1.tiny
%javac A12.java
%java A12 A1.tiny
```

The starter code for A11 is

```
import java.io.FileReader;
import java.io.BufferedReader;
import java.util.Set;
import java.util.HashSet;

public class A11 {
        static boolean isLetter(int character) {
                return (character >= 'a' && character <= 'z') || (character >= 'A' && character <= 'Z');
        }

        static boolean isLetterOrDigit(int character) {
                return isLetter(character) || (character >= '0' && character <= '9');
        }

        public static Set<String> getIdentifiers(String filename) throws Exception{
                String[] keywordsArray = { "IF", "WRITE", "READ", "RETURN", "BEGIN",
                                "END", "MAIN", "INT", "REAL" };
                Set<String> keywords = new HashSet();
                Set<String> identifiers = new HashSet();
                for (String s : keywordsArray) {
                        keywords.add(s);
                }
                FileReader reader = new FileReader(filename);
                BufferedReader br = new BufferedReader(reader);
                String line;
                while ((line = br.readLine()) != null) {
                        int i=0;
```

```
                        while (i < line.length()) {
                                if (line.charAt(i)=='\"'){
                                // throw away quoted strings
                                if (isLetter(line.charAt(i))){
                                // get the identifier
                                }

                                return identifiers;
                        }
                }
        }

        public static void main(String[] args) throws Exception{
                Set<String> ids=getIdentifiers("A1.tiny");
                for (String id :ids)
                        System.out.println(id);
        }
}
```

The starter code for A12 is

```
import java.io.*;
import java.util.HashSet;
import java.util.Set;
import java.util.regex.*;
public class A12 {
        public static Set<String> getIdRegex(String filename) throws Exception{
                String[] keywordsArray = { "IF", "WRITE", "READ", "RETURN", "BEGIN","END", "MAIN", "INT", "REAL" };
                Set<String> keywords = new HashSet();
                Set<String> identifiers = new HashSet();
                for (String s : keywordsArray)
                        keywords.add(s);

                FileReader reader = new FileReader(filename);
                BufferedReader br = new BufferedReader(reader);
                String line;
                //Pattern idPattern = ......;
                //Pattern quotedStringPattern = .....;
                while ((line = br.readLine()) != null) {
                Matcher m_quotedString = quotedStringPattern.matcher(line);
                                String lineWithoutQuotedStrings = m_quotedString.replaceAll("");
                                Matcher m = idPattern.matcher(lineWithoutQuotedStrings);
                                while (m.find()) {
                                        String id = line.substring(m.start(), m.end());
                                        if (!keywords.contains(id))
                                                identifiers.add(id);
                                }
                }
                return identifiers;
     }
     public static void main(String[] args) throws Exception{
        Set<String> ids=getIdRegex("A1.tiny");
        for (String id :ids)
                System.out.println(id);
     }
}
```

Hints to work on your program

Try to run the following code first. Then modified it into A12.

```
import java.util.regex.*;
public class RegexTest {
public static void main(String args[]) {
```

```
        String pattern = "\\d{4}-(0?[1-9]|1[012])-\\d{2}";
        String text = "final exam 2008-04-22, or 2008-4-22, but not     2008-22-04";
        Pattern p = Pattern.compile(pattern);
        Matcher m = p.matcher(text);
        while (m.find()) {
                System.out.println("valid date:"+text.substring(m.start(),  m.end()));
        }
        }
}
```

Marking Scheme

```
yourMark=0;
if (A11.java, A12.java are not sent properly) return;
for (each of A11, A12)
     if (it is compiled correctly)  yourMark+=1;
for (each of A11, A12){
         if (your java program reads A1.tiny && generates corrent results)
                    for (each of the 6 tests cases)
                          if (it is correct)   yourMark+=0.5;
         if (youCode.length() among the top 6 students) yourMark+=0.5;

 }
 for (each day of your late submission)  yourMark=yourMark*0.8;
```

Bonus Mark

If your code is among the top six in terms of length, you will receice 0.5 point for A11 or A12.