

PL Assignment #2 : Recognizing Tokens

과제물 부과일 : 2015-03-16(월)

Program Upload 마감일 : 2015-03-22(일) 23:59:59

문제

다양한 형태의 identifier, integer number(음수 포함) 들로 이루어진 text file을 입력 받아, 각 요소를 인식하여 출력하는 program을 작성하시오. Input file name은 as02.txt이다.

예를 들어 as02.txt file의 내용이 아래와 같다면,

```
banana 267 h cat -3789 7 y2010
```

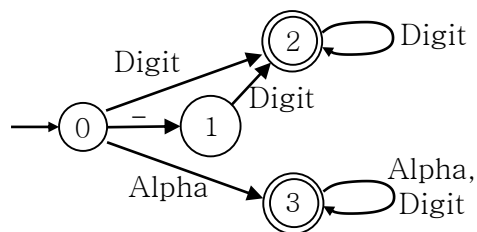
출력은 아래와 같아야 한다.

```
id: banana
int: 267
id: h
id: cat
int: -3789
int: 7
id: y2010
```

Regular Expression

```
id:      Alpha[Alpha|Digit]*
int:     Digit+ | "-" Digit+
Alpha:   [A-Z] | [a-z]
Digit:   [0-9]
```

mDFA



Programming

Token 표현하기

```
public enum TokenType{
    ID(3), INT(2);

    private final int finalState;

    TokenType(int finalState) {
        this.finalState = finalState;
    }
}
```

Data Type

```
public static class Token {
    public final TokenType type;
    public final String lexme;

    Token(TokenType type, String lexme) {
        this.type = type;
        this.lexme = lexme;
    }

    @Override
    public String toString() {
        return String.format("[%s: %s]", type.toString(), lexme);
    }
}
```

Programming

```
public class Scanner {
```

```
    private String source;
    private StringTokenizer st;

    public Scanner(String source) {
        this.source = source == null ? "" : source;
        // 문자열을 토큰화 해주는 클래스, " "을 delimiter로 함
        this.st = new StringTokenizer(this.source, " ");
    }

    private Token nextToken() {
        int state = 0;
        boolean errorState = false;
        // 토큰이 더 있는지 검사
        if (!st.hasMoreTokens())
            return null;
        // 그 다음 토큰을 받음
        String temp = st.nextToken();
        Token result = null;

        for (int i = 0; i < temp.length() && !errorState; i++) {
            switch (state) {
                case 0:
```

```

        if (Character.isDigit(temp.charAt(i))) state = 2;
        else if (temp.charAt(i) == '-') state = 1;
        else if (Character.isLetter(temp.charAt(i))) state = 3;
        else errorState = true;
        break;
    case 1:
        //상태에 따라 state를 변경하도록 아래를 채우시오.
    case 2:
        //상태에 따라 state를 변경하도록 아래를 채우시오.
    case 3:
        //상태에 따라 state를 변경하도록 아래를 채우시오.
    default:
        System.out.println("Case error: " + temp);
        return result;
    }
}

if (errorState) {
    System.out.println("acceptState error: " + temp);
    return result;
}

switch (state) {
    case 2:
        //해당 상태에 맞는 토큰을 생성하여 반환
    case 3:
        //해당 상태에 맞는 토큰을 생성하여 반환
    }
    return result;
}

public List<Token> tokenize() {
    List<Token> tokens = new ArrayList<Token>();
    Token t = null;

    //토큰 List를 반환하도록 작성

    return tokens;
}

public static void main(String[] args) {
    //txt file to String
    String source = ...
    Scanner s = new Scanner(source);
    List<Token> tokens = s.tokenize();
    //print
}

```

}유의 사항

- 입력 data는 프로그램을 제대로 검증할 수 있는 data로 구성되어야 한다.

수정: 2015-03-16