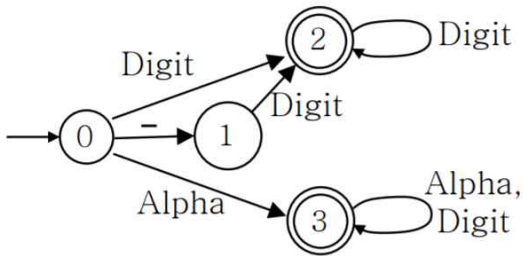




과목명	프로그래밍 언어 개론
담당교수님	조은선 교수님
이름	정성욱
학번	201404376

## 문제 해결 방법

이 문제에서는 읽어들이는 문자에 따라 상태가 나뉩니다.  
다음의 문자들에 대해 데이터 처리를 해야 됩니다.  
A-z, 1-9, '-'  
아래는 상태도입니다.



읽어 들어오는 문자들에 따라 상태가 달라집니다.A-z는 문자(3), 1-9는 숫자(2), 초기상태(0), (1)은 - 상태입니다. 이것을 표현한 코드가

```
private void initTM() {
    transM = new int[4][128];
    // 정의된 상태 이외의 상태는 전부 알수없기 때문에 -1로 마스킹해줌
    for(int i=0; i<transM.length; i++){
        for(int j=0; j<transM[i].length; j++){
            transM[i][j] = -1;
        }
    }
    //0-9는 숫자 A-z는 문자 - 는 1의 상태 초기상태 0을 정의 해줍니다.
    //아스키코드표값 참고
    for(int dec =48; dec<=57; dec++) {
        transM[3][(char)dec] = 3;
        transM[2][(char)dec] = 2;
        transM[1][(char)dec] = 2;
        transM[0][(char)dec] = 2;
    }

    transM[0]['-'] = 1;
    for(int dec =65; dec<123; dec++) {
        transM[3][(char)dec] = 3;
        //transM[2][(char)dec] = 3;
        //transM[1][(char)dec] = 3;
        transM[0][(char)dec] = 3;
    }
}

st = new StringTokenizer(this.source, " ", true);
```

이부분입니다. 로직을 보면, -를 받아서 숫자가 나오면 digit으로 처음부터 숫자면 digit으로 문자(A-z)를 받으면 Alpha로 이동합니다. 이외의 상태는 전부 -1로 갑니다. Scanner 자체를 재정의 한 것이 이 과제의 핵심입니다. Token 정적 클래스를 정의 하여, 상태를 정의해두도록 한 구조를 이용합니다. 흐름을 보면 스캐너로 문자열을 읽어서 transM의 배열을 만든후 transM을 초기화하는 함수를 실행합니다. initTM 함수내에서 오토마타 배열들의 상태정의가 이루어지고, String 토큰라이저 객체를 이용하여 “ ”로 토큰으로 쪼갭니다. 이제 List<Token> tokens = s.tokenize()가 실행이되는데, 이 tokenize()는 최종 결과를 출력할 리스트를 만듭니다 그리고 현재 스캐너 클래스가 가지고 있는 토큰을 가져오기 위해 nextToken을 이용합니다. 그럼 이제 nextToken을 보겠습니다.Token을 리턴하는데 시작하면 일단 stateId에 0을 넣어둬서 시작하면 바로 초기 상태를 명시적으로 지정합니다. 그리고 위에서 정의한 st객체가 토큰을 가지고없다면 null을 리턴합니다. 그렇지 않다면 temp에 토큰하나를 받아오는데 혹시 이것이 공백과 같다면 공백이 아닌 것을 읽을 때까지 계속 한 문자씩 읽어들이는 겁니다. 그리고 공백이 아닌 문자를 읽게 되었을 때 최종적으로 리턴할 값인result를 정의하고 상태를 체크하는 루프를 실행합니다. 위에서 정의한 transM이 가지고있는 인덱스 현재값을 넣어 그 값들이 가르키고있는 상태를 확인하고 어떻게할건지 판별합니다 예를 들어 [0][a] a는 97이고 위에선 (char)dec로 둬, 배열값의 의미는 초기상태 - 문자 -이므로 문자로 가는 것입니다. 정의되지않은 값이면 에러 문자 출력하고 바로null을 리턴합니다. 매루프가 끝날때마다 현재상태를 옛날상태로 두고 다시 올라가서 새것을 받아서 진행합니다. 모든 문자를 다 받았다면 이제 마지막으로 상태 정의를 할 차례입니다. 현재까지 받은 토큰들의 값들을 하나씩 받아와서 비교하고 마지막 상태의 값과 같다면 result에 id와 현재 읽은 문자열을 매개변수로 Token 객체를 만들고 return으로 result를 리턴합니다. 이로써 하나의 문자열에 대한 처리가 끝나는 것입니다.

## 느낀 점

오토마타에 익숙하지 않아 조금 헷갈렸습니다.

## 구현 결과

입력 : banana -33 aaaa12 ss-123

출력 :

unknown state! rejected

[[ID: banana], [INT: -33], [ID: aaaa12]]