

情報科学実験 B

担当：松本 真佑 先生

提出者：秋口 敬

コース：計算機科学コース

学籍番号：09B19002

提出年月日：2021 年 10 月 21 日

1 課題の目的

Pascal 風言語で記述されたプログラムからトークン列を切り出すプログラムを作成する。

2 課題達成の方針と設計

1 行ずつ読み取り、前から 1 字ずつ保存していく。

現在の位置がコメントアウトの箇所か、文字列の箇所か、それ以外化によって処理を変更する。もし特殊記号が検出されたなら、その特殊記号をトークンとして分ける。

その時点で文字列が保存されているはずなので、保存された文字列がどのトークンに類するかを調べる。

それを文字列、特殊記号の順で必要情報を出力する。

3 プログラムの実装方針

3.1 1 行ずつ読み取り、前から 1 字ずつ保存していく

pascal 風言語で書かれたファイルについて存在するかどうかを確かめる。その後、読み込みができるかどうかを確認し、読み込みができるなら次節以降の処理をする。できないならエラーを出力する。

3.2 現在の位置がコメントアウトの箇所か、文字列の箇所か、それ以外化によって処理を変更する

```
if(comout==0) {
if(mojiretu==0) {
if(c[n]=='{') {
comout=1;
n++;
}

else if(){
3 節以降の処理
}
else {

tan=tan+c[n];
n++;
}
if(n>=mojisuu) {
if(tan!="") {
出力処理
```

```

}
}
}
else {
    mojire=mojire+c[n];
    if(c[n]=='\\') {
        出力
    }
    n++;

}
}
else {
    if(c[n]=='}') {
        comout=0;
    }
    n++;
}
}

```

もし、現在見ている文字がコメントアウトをあらわす文字”{”なら、コメントアウトをあらわすフラグを立てて、その次の文字以降はコメントアウトとして扱い、無視する。コメントアウト中に”}”が来たなら、コメントアウトのフラグを折る。文字列の開始をあらわす文字”””が来たら文字列をあらわすフラグを立て、その文字以降は文字列として保存する。文字列中に”””が来たら文字列のフラグを折る。それ以外なら次節以降の処理を行う。

4 もし特殊記号が検出されたなら、その特殊記号をトークンとして分ける

ここでは例として、”>”、”>=”を検出した時の処理について説明する。

```

else if(c[n]=='>') {
    n++;
    if(n<mojisuu) {
        if(c[n]=='=') {
            n++;
            ID=28;
        }else {
            ID=29;
        }
    }else {
        ID=29;
    }
    if(tan!="") {

```

出力処理

```

}
syuturyoku=Jiku[ID]+space+Token[ID]+space+ID+space+linecount;
pw.println(syuturyoku);
tan ="";
}

```

n 文字目が”,”なら n+1 文字目も見て、”=”ならトークンは”>=”、トークン ID は 28 となる。そうでないならトークンは”>”、ID は 29 である。同様の比較、処理を特殊記号すべてについて行う。

5 その時点で文字列が保存されているはずなので、保存された文字列がどのトークンに類するかを調べる

```

public int mojiitti(String s) {
    int aide=-1;
    String slash="/";
    for(int i=0;i<=23;i++) {
        if(Objects.equals(s, Jiku[i])) {
            aide=i;
        }
    }
    if(aide==-1) {
        if(s.equals(slash)) {
            aide=5;
        }
        else if(s.matches("[+-]?\\d*(\\.\\d+)?")) {
            aide=44;
        }
        else {
            aide=43;
        }
    }
    return aide;
}

```

まず、単語が綴り記号と一致するかを調べる。次に、正規表現を用いて数字かどうかを調べる。どちらでもなければ、保存された単語は名前とする。これにより、単語のトークン、ID がわかる。

6 文字列、特殊記号の順で必要情報を出力する。

```

IDtan=mojiitti(tan);
syuturyoku=tan+space+Token[IDtan]+space+IDtan+space+linecount;
pw.println(syuturyoku);

```

```
syuturyoku=Jiku[ID]+space+Token[ID]+space+ID+space+linecount;  
pw.println(syuturyoku);  
tan ="";
```

単語について、単語、トークン名、ID、行を printwriter を用いて出力する。同様に特殊記号についても特殊記号、トークン名、ID、行を出力する。また、これとは別に文字列の処理を行っているが、その場合はその文字列、トークン名、ID、行を出力する。

7 まとめ

これにより、正しく ts ファイルに出力され、実行テストにも通った。

8 感想、謝辞

課題をいくつかのステップに分けて行うことでスムーズにプログラムを作成することができた。授業や slack での質問に答えていただいた教員の方々に感謝の意を表します。