

情報科学演習 D

第 1 回レポート課題

提出年月日：平成 30 年 10 月 18 日

電子メール：u839996e@ecs.osaka-u.ac.jp

学籍番号：09B16084

氏名：山野廣大

1 課題 1 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ことを本課題の目的として設定する。

第 1 引数で解析する pas ファイルが、第 2 引数で出力する ts ファイルが指定される。正常に処理が終了した場合は標準出力に”OK”を出力する。入力ファイルが見つからない場合は標準エラーに”File not found”と出力し、終了する。

2 プログラムの動作原理

2.1 プログラムの流れ

1. 入出力用のファイルを作成する
2. 入力ファイルを読み込みモードでオープンする ※
3. 入力ファイルを 1 行読み込み、行数カウンタを 1 増やす
4. 読み込んだ行に文字が含まれていたらその行の文字列と行数カウンタを引数として、トークンを作成しトークンリストに追加するメソッドを呼び出す
5. 入力ファイルの最後の行が読み込まれるまで 3,4 を繰り返す
6. 作成したトークンリストを利用して出力ファイルに指定された形式で書き込む
7. 入出力用のファイルを閉じる
8. 標準出力に”OK”と出力する

※入力ファイルが見つからない場合は標準エラーに”File not found”と出力し、終了する

3 プログラムの実装方法

この章では上記のプログラムの流れの 4 のトークンを作成する部分とトークンリスト、6 の出力ファイルに書き込む部分についての説明を行う。

3.1 トークンの作成

トークンの作成では引数として渡された 1 行分の文字列をまず 1 文字ずつに分解して String 型の配列に格納する。そしてその 1 文字ずつを判定していく。判定する際の基準は以下の 5 種類である。

- 指導書で指定されたトークンであるか
- 数字であるか
- アルファベットであるか

- ‘であるか
- { であるか
- 以上のどれにも当てはまらないか

以上の判定を行い、StringBuilder のクラスを用いて 1 つの塊を作る。そして次に、その塊についてどのトークンに対応しているのか、もしくは注釈であるのかを判定する。

次に、文字判定の基準と、塊の作り方、作成すべきトークンの判定について詳しく説明する。

3.1.1 指導書で指定されたトークンであるかの判定

指定されたトークンの判定は指導書 4.2 トークンの種類の ID24 から 42 をそのままの順番で、その後ろの要素に"/" を String 型のリストに格納し、ArrayList クラスの contains メソッドを用いて判定を行う。1 行を 1 文字ずつに分解した際の保存先の型を char ではなく String にしたのは ArrayList の型が String 型であるためだ。contains メソッドで true が帰ってきた場合はもう 1 文字先まで読み込み 2 文字で 1 つのトークンとなっているかを再び判定する。2 文字で 1 つのトークンとなっていた場合は StringBuilder を用いてその 2 文字を 1 つの塊とする。そうして作成した塊に対して新しいトークンを作成する。String 型の塊を引数とし、対応するトークンの ID、トークン名を判定するメソッドを読み込む。ID の取得は先ほど説明した ArrayList クラスの indexOf メソッドを用いる。指導書では初めに出てくるトークンである"=" の ID が 24 であるため、indexOf メソッドで取得した値に 24 を加えて返り値とする。また、"/" のみ ID が 5 であるため、引数として"/" が与えられた場合のみ条件分岐して 5 を返すように設計している。次にトークン名の取得は String 型のトークン名を格納するリストを用いる。このリストには ID 番号順に指導書で指定されたトークン名が格納されている。先ほど取得した ID を引数として渡し、ID 番目の要素を返り値としている。このようにして得られた ID、トークン名に加えソースコード中に出てきた文字列、行番号を Token クラスのインスタンス変数に代入することでトークンの作成としている。

3.1.2 数字であるかの判定

数字の判定も 0 から 9 までの数字を String 型のリストに格納し、contains メソッドを用いて判定する。数字であると判定した場合は数字以外の文字が読み込まれるまで 1 文字ずつ読み込んで判定していく。数字以外の文字が出現した場合はその文字の 1 つ前までを塊としトークンを作成する。数字トークンの作成は ID に 44 を代入し、これは必ず必要な仕様ではないが判定した数字を int 型の数値に変形してトークンの変数に代入している。

3.1.3 アルファベットであるかの判定

アルファベットの判定は上記の指定されたトークン、数字どちらにも含まれない文字列でアルファベット以外を排除したものとして判定する。1 文字がアルファベットと判定されたらアルファベットまたは数字以外を読み取るまで連続で判定し、アルファベットまたは数字以外を読み込んだ文字の直前までを塊にする。次にその塊がキーワードの文字列と一致するかを判定する。子の判定も String 型のリストの contains メソッドを用いて行う。true が返ってきた場合はキーワードとなり、indexOf メソッドを用いてトークン ID を取得し、ID を元にトークン名を取得してトークンを

作成する。キーワード判定で false が帰ってきた場合は名前として ID43 の名前としてトークンを作成する。

3.1.4 ‘であるかの判定

入力として与えられる pas ファイルのプログラムは’ が出現したら次に’ が出現するまでの間の文字列を”文字列”として扱う必要がある。そのため、’ から’ までの間の要素に関してはトークンの判定を行わずにそのすべてをまとめて 1 つの塊にしている。そしてできた塊を ID45 の文字列としてトークン化する。

3.1.5 { であるかの判定

入力として与えられる pas ファイルのプログラムは”{“が出現したら”}” が出現するまでの文字列を注釈として扱う必要がある。そのため”{“であると判定したら”}” が出現するまでその間の文字列に関してはトークンを作らず index の値をインクリメントしている。

3.1.6 その他の判定

以上のどれにも当てはまらない場合（空白文字やタブ文字を想定している）に関しては index をインクリメントし、トークン化はせず読み飛ばす処理を行うよう設計している。

3.2 トークンリストへの追加

上記で作られた塊がどのトークンに対応するか判定した後、それをトークンとしてトークンリストに追加した。その方法としてトークン名、ID、行数番号、ソースコード中に出てきた本来の文字列(数値)を格納するインスタンス変数を持つ Token クラスの set メソッドを使ってそれぞれの値を格納した後そのトークンをリストへ add することにより実装した。

3.3 出力ファイルへの書き込み

ts ファイルへの書き込みは作成したトークンリストの各要素について書き込み用の形式に変形してそれを書き込むという方法をとった。書き込み用の形式への変形は StringBuilder のクラス、Token クラスの get メソッドを用いて行った。

4 まとめ

課題 1 では 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した。トークン列を切り出すうえで入力プログラムから 1 行ずつ読み取り、その 1 文字ずつ判定していくという方法を選んだ。そこで判定したトークンについて、判定が終わった順（プログラムに出てきた順）にトークンリストに追加していき入力ファイルをすべて読み終えたらトークンリストをもとに出力形式に沿う形で出力ファイルに書き込みを行った。以上が私の作成したプログラムのまとめであり、指導書記載の外部使用は全て満たすことができた。

5 感想

私はトークンの作成の段階で初めは1文字ずつ分けるのではなく空白文字またはタブ文字で分解してからそれをさらに分解しようとしていた。しかしそれでは文字列の中の空白文字またはタブ文字も分解してしまい、使用を満たす動作が難しいと判断したため1文字ずつ分解し判定する仕様に変更した。

また、1文字ずつ判定していく際、1文字目が指定されたトークンと一致した場合に次の文字を読み込んで2文字で1つのトークンとなるもの（今回では"<>","<=",">=",":=","..")か判定する必要があるのだが、それは1文字目を見れば2文字でトークンとなりうるものか予想できるが今回は全てのトークンに対して2文字目を読み込んでその2文字でトークンと一致するか判定する仕様とした。ここは改善できる余地があると思う。

また、トークン作成の部分で1文字ずつ判定していく仕様にしたのだが、読むべき文字数を超えて参照しようとして、NullPointerExceptionのエラーが出たり、最後の文字が読めていなくてテストに失敗したりという部分でやや苦労した。

謝辞

課題1を行うにあたり、基本的な質問をTAや教員の方々にしてしまい申し訳ないという気持ちとともに感謝の気持ちでいっぱいです。また、演習時間外にもメールで質問対応していただき大変参考になりました。ありがとうございました。引き続きご指導宜しくお願い致します。

参考文献

[1] コンパイラの入り口、「字句解析」のための文字列操作

<http://www.atmarkit.co.jp/ait/articles/0705/15/news135.html>