独習Java
第11章　11.6~11.7

11.6 ランダムアクセスクラス

11.7 Stream Tokenizerクラス

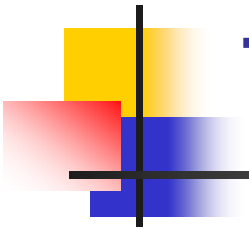
07/06/22　名雪　勲

RandomAccessFileクラス

- RandomAccessFileクラスを使用するとファイル内のどの位置にもシークし、その位置でデータを読み取り、または書き込むプログラムを作成することができる。
- 次にファイルの最後から指定した個数のバイトを表示するプログラムを紹介する。

プログラム例(1/3)

```
import java.io.*;
class Tail {
 public static void main(String args[]) {
   try{
     //RandomAccessFileオブジェクトを作成する
   RandomAccessFile raf = new RandomAccessFile(args[0], ”r”);
 //ファイルの終わりに表示するバイト数を決める
long count = Long.valueOf(args[1]).longValue();
//ファイルの長さを決める
long position = raf.length();
//現在の位置にシークする
 position -= count;
if(position < 0)
```



プログラム例(2/3)

```
position = 0;
raf.seek(position);
//バイトを読み取って表示する
while(true) {
   //バイトを読み取る
   try {
       byte b = raf.readByte();
       //文字として表示する
        System.out.print((char)b);
  }
catch (EOFException eofe) {
   break;
```

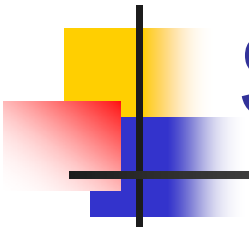
プログラム例(3/3)

```
        }
      }
    }
catch (Exception e) {
  e.printStackTrace();
    }
  }
}
```

出力結果

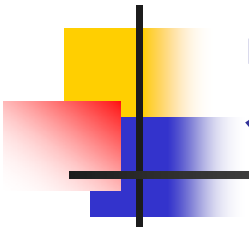
```
e.printStackTrace();
      }
    }
  }
```

出力結果はこのプログラムファイル、Tail.javaの最後の40バイトが表示されます。



StreamTokenizerクラス

- このクラスは文字入力ストリームからデータを解析して、トークンの列を生成します。
- トークンとは文章を分割した時の数字または単語を表しています。



定数

- このクラスでは4つの定数と3つの変数が定義されています。

定数or変数名	意味
TT_EOF	ファイルの終わり状態を示す
TT_EOL	行の終わり状態を示す
TT_NUMBER	数字を読み取ったことを示す
TT_WORD	単語を読み取ったことを示す
nval	現在のトークンが数字の場合に値を格納
sval	現在のトークンが文字列の場合に値を格納
ttype	数字でも文字列でもないときに値を格納

プログラム例(1/4)

- 次の例ではテキストファイルを読み取って、その内容を解析し、行数と単語を表示します。

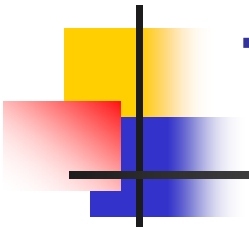
```
import java.io.*;
class StreamTokenizerDemo {
  public static void main(String args[]) {
 try {
  //FileReaderオブジェクトを作成する
  FileReader fr = new FileReader(args[0]);
 //BufferedReaderオブジェクトを作成する
  BufferedReader br = new BufferedReader(fr);
//StreamTokenizerオブジェクトを作成する
  StreamTokenizer st = new StreamTokenizer(br);
```

プログラム例(2/4)

```
//ピリオドを通常文字として定義する
st.ordinaryChar('.');
//アポストロフィーを単語文字として定義する
st.wordChars('¥'','¥'');
//トークンを処理する
while (st.nextToken() != StreamTokenizer.TT_EOF) {
   switch(st.ttype) {
     case StreamTokenizer.TT_WORD:
        System.out.println(st.lineno() + ")" + st.sval);
          break;
     case StreamTokenizer.TT_NUMBER:
         System.out.println(st.lineno() + ")" + st.nval);
            break;
```

プログラム例(3/4)

```
 default:
    System.out.println(st.lineno() + “)” + (char)st.ttype);
  }
 }
//FileReaderオブジェクトをクローズする
fr.close();
 }
 catch (Exception e) {
   System.out.println(“Exception: “ + e);
     }
   }
}
```

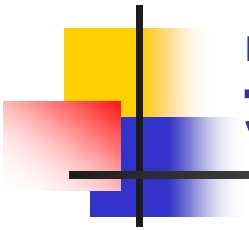


プログラム例(4/4)

- ファイルtokens.txtには次の行が入っているものとする。

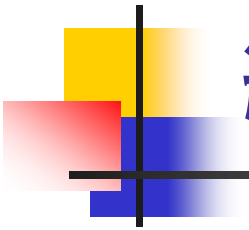
The price is $23.45.
Is that too expensive?
(I don’t think so.)

実行コマンド
java StreamTokenizerDemo tokens.txt



実行結果

1)The
1)price
1)is
1)$
1)23.45
1).
2)Is
2)that
2)too
2)expensive
2)?
3){
3)I
3)don’t
3)think
3)so
3).
3))



練習問題

- StreamTokenizerクラスを利用して、プログラム中にいくつ　{　が使われているかをカウントして合計数を表示するプログラムを作成せよ。