

# 応用プログラミング [Q2] レポート

C0115114 菅野路哉

2016 年 5 月 7 月

## 1 第 3 回 課題 2

リスト 1 のようなファイルから Iterator パターンを使ってデータを 1 つ 1 つ読み取り、出力するプログラムを作成する。

ファイルから学籍番号を取り出すクラスは StudentDirectory, そのイテレータに相当するクラスは StudentDirectoryIterator という名前にする。

リスト 1 入力データが記録されたファイル

```
C0116001
C0116002
C0116003
C0116004
C0116005
C0116006
C0116007
C0116008
C0116009
C0116010
```

## 2 ソースコード

Aggregate 実装

リスト 2 StudentDirectory.java

```
1 import java.io.BufferedReader;
2 import java.io.FileReader;
3 import java.io.IOException;
4
5 import java.util.List;
6 import java.util.ArrayList;
7 import java.util.Iterator;
8
9 public class StudentDirectory implements Iterable{
10     private List<String> students = new ArrayList<String>();
11     private int length;
12     public StudentDirectory(String filename){
13         try{
14             BufferedReader br = new BufferedReader(new FileReader(filename));
15             String line = null;
16             while((line = br.readLine()) != null){
17                 this.students.add(line);
18             }
19             this.length = this.students.size();
20         }catch(IOException e){
21         }
22     }
23
24     public int getLength(){
```

```

25     return this.length;
26 }
27
28 public String getStudentAt(int index){
29     return this.students.get(index);
30 }
31
32 @Override
33 public Iterator iterator(){
34     return new StudentDirectoryIterator(this);
35 }
36 }

```

## Iterator 実装

リスト 3 StudentDirectoryIterator.java

```

1 import java.util.Iterator;
2
3 public class StudentDirectoryIterator implements Iterator{
4     private StudentDirectory studentDirectory;
5     private int index;
6     public StudentDirectoryIterator(StudentDirectory studentDirectory){
7         this.studentDirectory = studentDirectory;
8     }
9
10    @Override
11    public boolean hasNext(){
12        return this.index < this.studentDirectory.getLength();
13    }
14
15    @Override
16    public Object next(){
17        return this.studentDirectory.getStudentAt(this.index++);
18    }
19 }

```

## 実行ファイル

リスト 4 Main.java

```

1 public class Main{
2     public static void main(String[] args){
3         for(Object o: new StudentDirectory("./data.txt")){
4             System.out.println(o);
5         }
6     }
7 }

```

## 3 実行結果

リスト 5 実行結果

```

C0116001
C0116002
C0116003
C0116004
C0116005
C0116006
C0116007
C0116008
C0116009
C0116010

```

## 4 解説

リスト 2 では、`Iterable` interface に対して実装を行った。`Iterable` は `Aggregate` に相当する。コンストラクターでファイル名を受け取り、そのファイルを一行ずつ読み込む。読み込んだデータを、`List` である `students` に記録していく。データ件数は `length` に記録する。

リスト 3 では、`Iterator` interface に対して実装を行った。`hasNext` では、データ件数である `StudentDirectory` のデータ件数と現在のインデックスを比較し、データが残っているかを判別する。`next` では、`StudentDirectory` から `getStudentAt` を使い、引数番目のデータを受け取る。