

시스템 분석및 설계			
작성자	20141640 이석원	제출일	2019.04.28
담당 교수님	권기태	제출기한	2019.4.30

클래스의 종류와 하는일			
visitor			
visitor (파일과 디렉토리를 방문하는 방문자를 나타내는 추상 클래스)	메소드	void visit(File)	파일을 방문할 때의 처리를 하위클래스에 강제한다.
		void visit(Directory)	디렉토리를 방문할 때의 처리를 하위클래스에 강제한다.
package visitor; public abstract class Visitor { public abstract void visit(File file); public abstract void visit(Directory directory); }			
FileNameFindVisitor (visitor의 하위클래스, 파일의 이름에 따라 검색하는 역할을 하는 visitor다.)	필드	String name	파일을 검색할 검색어를 저장한다.
		ArrayList found	검색한 파일들을 저장한다.
	메소드	void visit(File)	name을 이용해서 File.getName()과 비교한(contains메소드)후 name을 포함한다면 그 객체를 found에 저장한다.
		void visit(Directory)	Directory의 iterator를 받아서 Directory의 내용물에 다시 accept요청을 합니다.
package visitor; import java.util.Iterator; import java.util.ArrayList; public class FileNameFindVisitor extends Visitor { private String name; private ArrayList found = new ArrayList(); public FileNameFindVisitor(String name) { this.name = name; } public Iterator getFoundFiles() {			

```

        return found.iterator();
    }
    public void visit(File file) {
        if (file.getName().contains(name)) {
            found.add(file);
        }
    }
    public void visit(Directory directory) {
        Iterator it = directory.iterator();
        while (it.hasNext()) {
            Entry entry = (Entry)it.next();
            entry.accept(this);
        }
    }
}

```

Element			
Element (인터페이스입니다. 이 인터페이스를 구현하는 객체에 visitor를 적용합니다.)	메소드	<i>void accept(visitor)</i>	visitor 객체에 대한 처리를 하위 클래스에게 구현하도록 강제합니다.

```

package visitor;

public interface Element {
    public abstract void accept(Visitor v);
}

```

Entry (인터페이스 Element를 구현하는 추상클래스입니다.	메소드	<i>getName</i>	name 변수의 getter메소드 구현을 강제합니다.
		<i>getSize</i>	size 변수의 getter메소드 구현을 강제합니다.
		Entry add(Entry)	추상메소드(Entry)의 add를 호출하면 에러를 발생시키도록 합니다.
		Iterator iterator()	추상메소드의(Entry)의 iterator를 호출하면 에러를 발생시키도록 합니다.
		String toString()	“파일이름 (파일크기)”형식의 문자열을 반환합니다.

```

package visitor;

```

```

import java.util.Iterator;

public abstract class Entry implements Element {
    public abstract String getName();
    public abstract int getSize();
    public Entry add(Entry entry) throws FileTreatmentException {
        throw new FileTreatmentException();
    }
    public Iterator iterator() throws FileTreatmentException {
        throw new FileTreatmentException();
    }
    public String toString() {
        return getName() + " (" + getSize() + ")";
    }
}

```

File (Entry를 상속하여 Entry의 추상 메소드를 구현합니다. iterator와 add를 재정의 하지않아서 File이 이 메소드들을 호출하면 예러가 호출됩니다.)	필드	String name	name 저장
		int size	size 저장
	메소드	String getName()	name의 getter
		int getSize()	size의 getter
		void accept(visitor)	현재 객체에 visitor를 적용합니다.

```

package visitor;

public class File extends Entry {
    private String name;
    private int size;
    public File(String name, int size) {
        this.name = name;
        this.size = size;
    }
    public String getName() {
        return name;
    }
    public int getSize() {
        return size;
    }
    public void accept(Visitor v) {
        v.visit(this);
    }
}

```

Directory	필드	<i>name</i>	name 저장
-----------	----	-------------	---------

(Entry를 상속하여 추상 메소드를 구현하며, iterator, add를 재정의 하여 이 메소드들을 호출해도 에러가 나지 않습니다.)		ArrayList dir	컨테이너로서 하위 항목들을 저장합니다.
	메소드	String getName()	name의 getter
		int getSize()	size의 getter 하위 항목들을 iterator를 이용하여 각각의 size를 모두 합한후 그값을 반환합니다.
		void accept(visitor)	현재 객체에 visitor를 적용합니다.
		Entry add(Entry)	dir ArrayList에 Entry를 저장합니다.
		Iterator iterator()	dir ArrayList의 iterator를 반환합니다.
Main			
<div>- root<ul style="list-style-type: none">- bin<ul style="list-style-type: none">- vi 10000- latex 20000- tmp- usr<ul style="list-style-type: none">- Kim<ul style="list-style-type: none">- diary.html 100- composit.java 200- hyejaKim.txt 150- Lee<ul style="list-style-type: none">- memo.tex 300- index.html 350- Leehyeja.txt 325- Park<ul style="list-style-type: none">- game.doc 400- junk.mail 500- Parkhyeja.txt 450</div>			
<pre>package visitor; import java.io.IOException; import java.util.Iterator; import visitor.*; public class Main { public static void main(String[] args) { // TODO Auto-generated method stub try { Directory rootdir = new Directory("root"); Directory bindir = new Directory("bin");</pre>			

