

Cấu trúc File and Folder

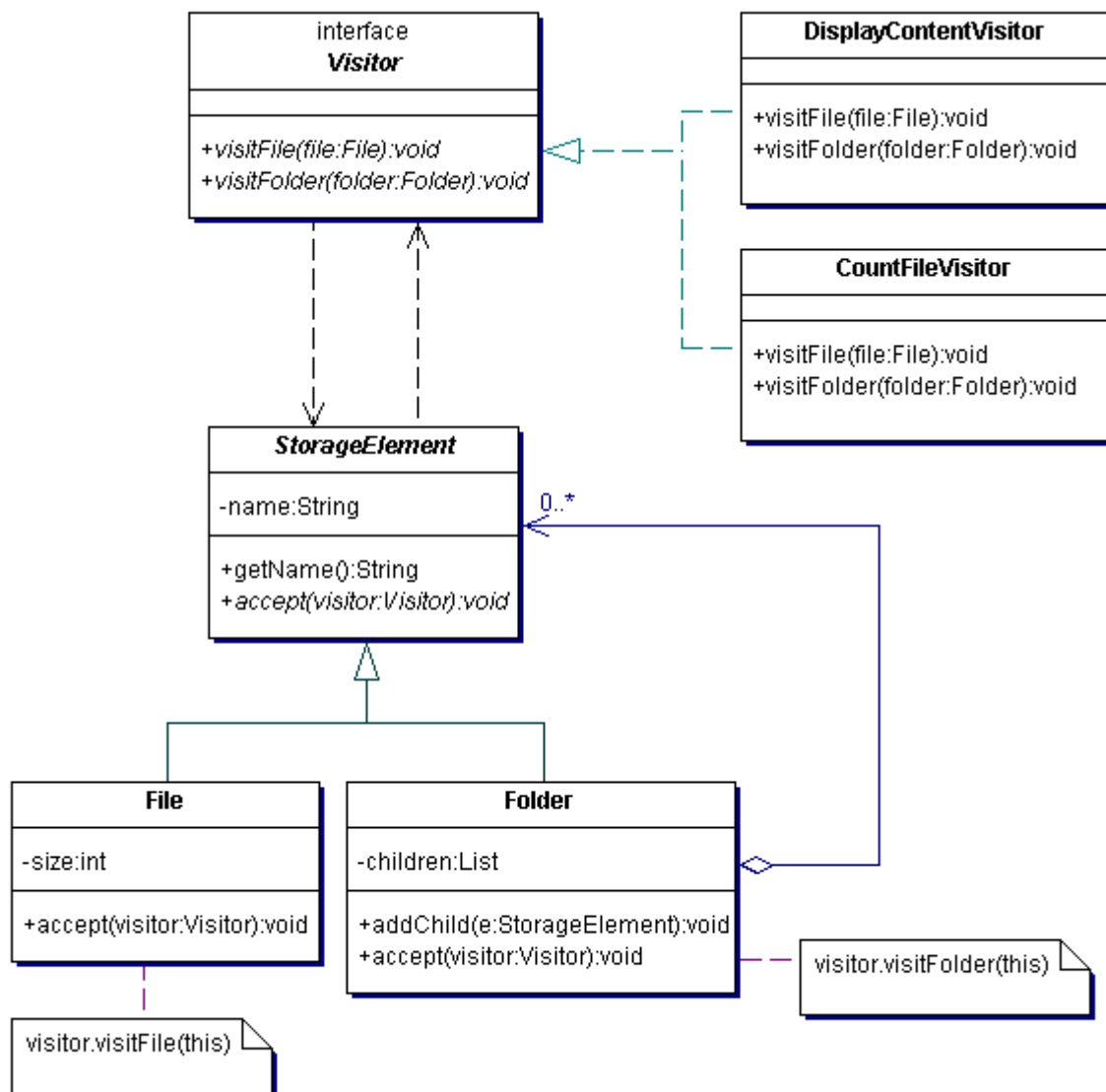
File và folder là các thành phần của một hệ thống quản lý tập tin.

- File biểu diễn dữ liệu cần lưu trữ.
- Folder là ngăn chứa các file. Trong mỗi folder có thể chứa các folder con khác.

Thiết kế lớp để biểu diễn cấu trúc File và Folder như trên

Thiết kế các thao tác xử lý sau trên cấu trúc file và folder, thiết kế sao cho có thể thêm các thao tác khác một cách dễ dàng

1. Tính kích thước file / folder
2. Đếm số file trong một folder
3. In nội dung file / folder
4. Tìm một file / folder
5. Xóa một file / folder trong một folder
6. Chép một file / folder từ một folder này sang một folder khác.



StorageElement

```
package fileSystem;

abstract public class StorageElement {
    private String name;

    public StorageElement(String name){
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public abstract Object execute(Visitor algo, Object... inp);
}
```

File

```
package fileSystem;

public class File extends StorageElement {
    private int size;

    public File(String name, int size){
        super(name);
        this.size = size;
    }

    public int getSize() {
        return size;
    }

    public Object execute(Visitor algo, Object... inp) {
        return algo.visitFile(this, inp);
    }
}
```

Folder

```
package fileSystem;

import java.util.ArrayList;
import java.util.List;

public class Folder extends StorageElement {
    private List<StorageElement> children = new ArrayList<StorageElement>();

    public Folder(String name){
        super(name);
    }

    public void addChild(StorageElement e) {
        children.add(e);
    }

    public List<StorageElement> getChildren(){
        return children;
    }

    public Object execute(Visitor algo, Object... inp) {
        return algo.visitFolder(this, inp);
    }
}
```

Visitor

```
package fileSystem;

public interface Visitor {
    Object visitFile(File host, Object... inp);
    Object visitFolder(Folder host, Object... inp);
}
```

CountFileVisitor

```
package fileSystem;

public class CountFileVisitor implements Visitor {

    public Object visitFile(File host, Object... inp) {
        return 1;
    }

    public Object visitFolder(Folder host, Object... inp) {
        int num = 0;
        for (StorageElement s : host.getChildren()){
            num += (Integer) s.execute(this);
        }
        return num;
    }
}
```

DisplayContentVisitor

```
package fileSystem;

public class DisplayContentVisitor implements Visitor {

    public Object visitFile(File host, Object... inp) {
        int level = (Integer)inp[0];
        StringBuffer s = new StringBuffer();
        for (int i = 0; i < level; i++){
            s.append("  ");
        }
        s.append(host.getName() + " " + host.getSize() + "\n");
        return s.toString();
    }

    public Object visitFolder(Folder host, Object... inp) {
        int level = (Integer)inp[0];
        StringBuffer s = new StringBuffer();
        for (int i = 0; i < level; i++){
            s.append("  ");
        }
        s.append(host.getName()+ "\n");

        for (StorageElement e : host.getChildren()){
            s.append(e.execute(this, level + 1));
        }
        return s.toString();
    }
}
```