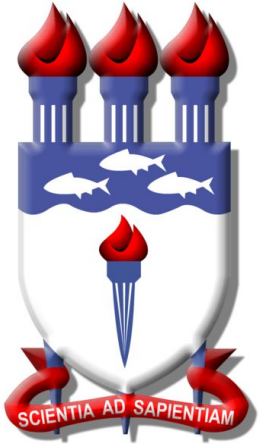
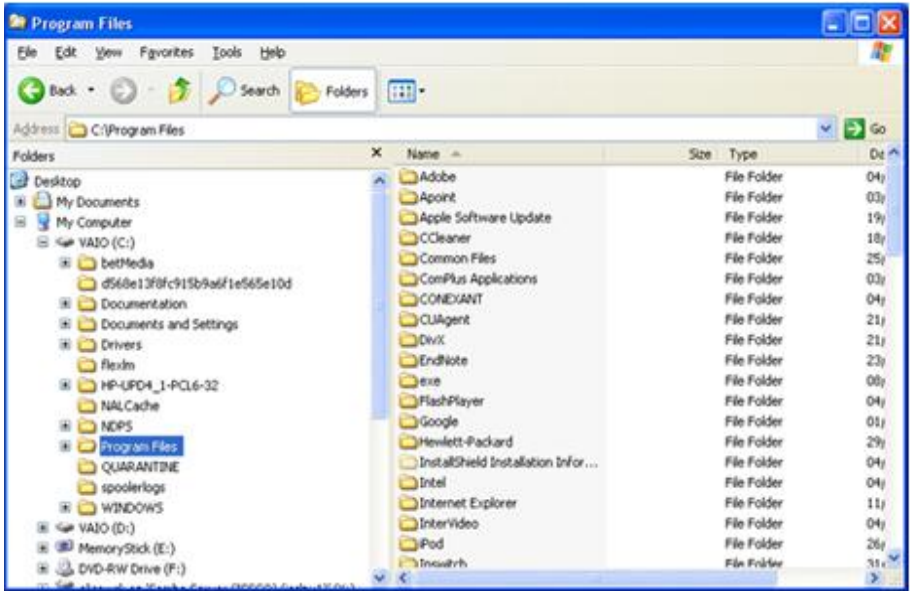


Design pattern: Composite

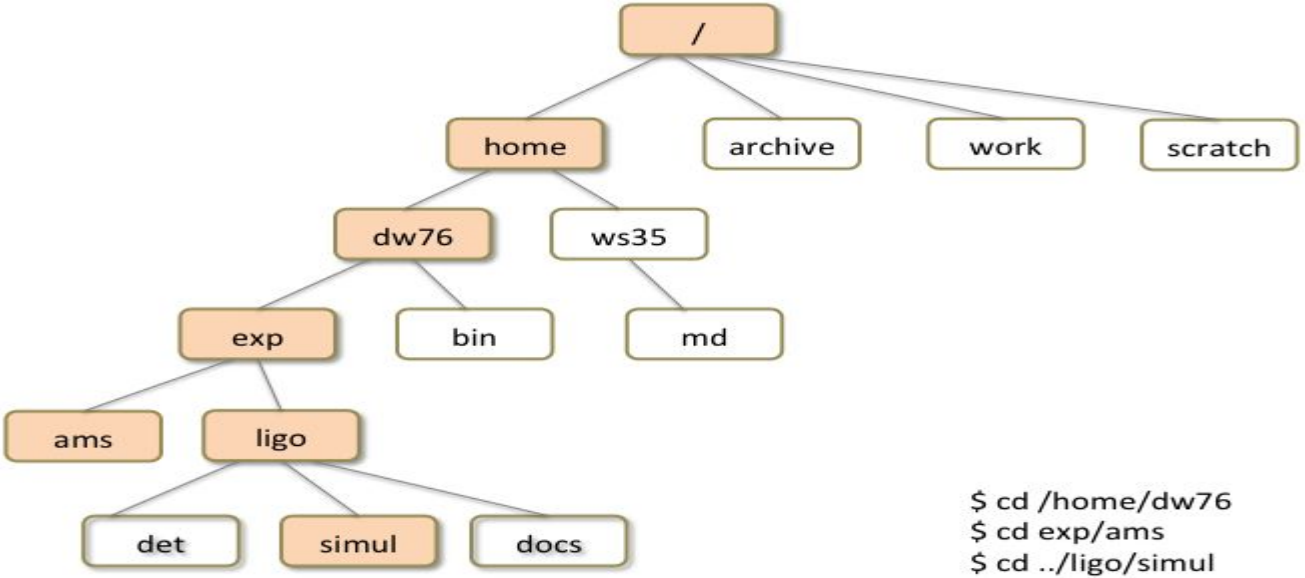
Gabriel Ferreira Cerqueira

Problem

Think about: How computer directories can be set? And what about their similarities with an ordinary program?



System directory tree



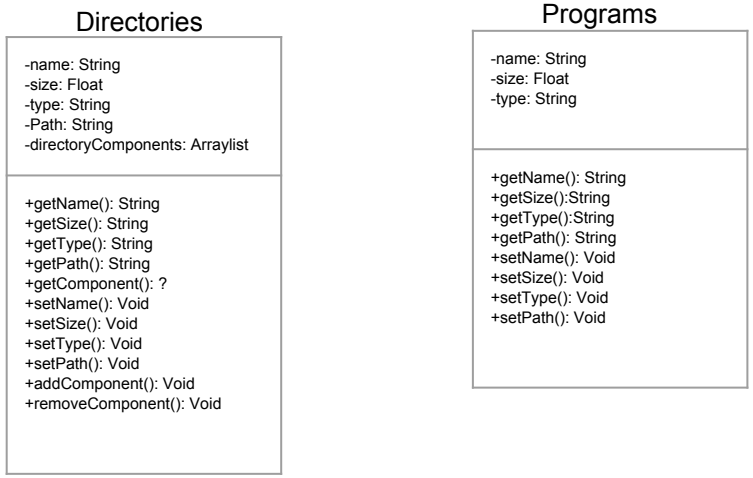

```
$ cd /home/dw76
$ cd exp/ams
$ cd ../ligo/simul
```

Motivation

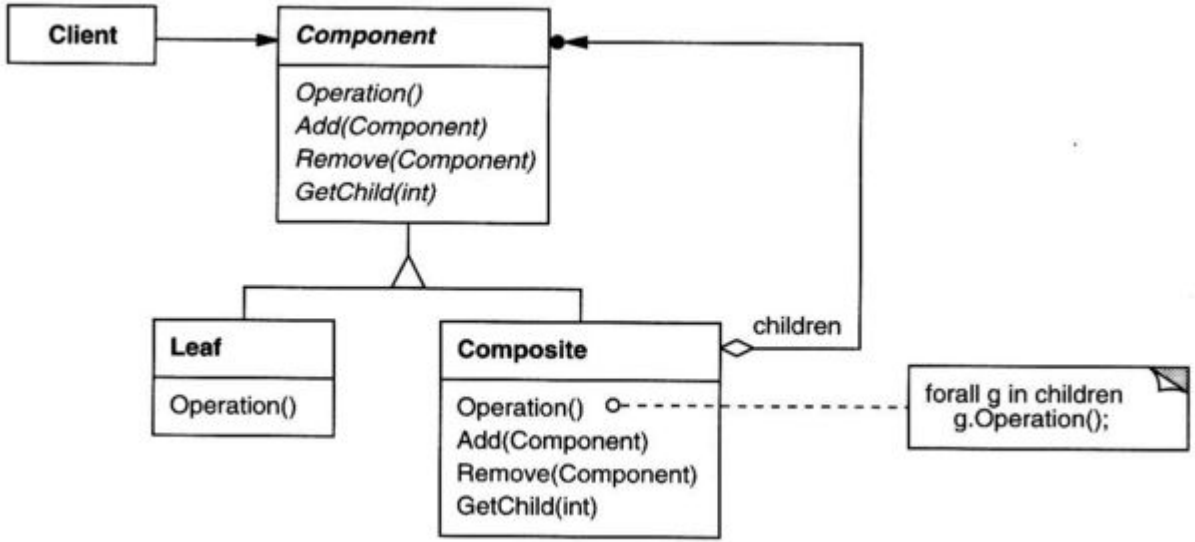
- Using composite pattern allows:
  - You to compose objects into tree hierarchical structures
  - The client to treat different types of objects uniformly.

source: Design patterns book

Different classes for different types



The Composite pattern



Applying the pattern

```
public abstract class DirectoryComponent {

        public void getName(){throw new UnsupportedOperationException();}

        public void getSize(){throw new UnsupportedOperationException();}

        public void getType(){throw new UnsupportedOperationException();}

        public void getPath(){throw new UnsupportedOperationException();}

        public void getcomponent(){throw new UnsupportedOperationException();}

        public void setSize(){throw new UnsupportedOperationException();}

        public void setType(){throw new UnsupportedOperationException();}

        public void setPath(){throw new UnsupportedOperationException();}

        public void addComponent(){throw new UnsupportedOperationException();}

        public void removeComponent(){throw new UnsupportedOperationException();}
}
```

Applying the pattern

```
public class Directory {
        private String name, type, path;
        private float size;
        private ArrayList <DirectoryComponent>;

        public void Directory(...){}

        public String getName(){// override method}

        public float getSize(){// override method}

        public String getType(){// override method}

        public String getPath(){// override method}

        public DirectoryComponent getcomponent(){// override method}

        public void setName(...){// override method}

        public void setSize(...){// override method}

        public void setType(...){// override method}

        public void setPath(...){// override method}

        public void addComponent(...){t// override method}
        public void removeComponent(...){// override method}
        }
```

Applying the pattern

```
public class Program {
        private String name, type, path;
        private float size;

        public void Program(...){}

        public String getName(){// override method}

        public float getSize(){// override method}

        public String getType(){// override method}

        public String getPath(){// override method}

        public void setName(...){// override method}

        public void setSize(...){// override method}

        public void setType(...){// override method}

        public void setPath(...){// override method}
```