

Visitor

```
// ComputerPart interface defining accept operation
public interface ComputerPart {
    public void accept(Visitor visitor);
}

// a computer part
public class Computer implements ComputerPart {
    ArrayList<ComputerPart> parts;
    public Computer() {
        this.parts = new ArrayList<ComputerPart>();
        this.parts.add(new Mouse());
        this.parts.add(new KeyBoard());
        this.parts.add(new Monitor());
    }
    @Override
    public void accept(Visitor visitor) {
        for(ComputerPart part : parts) {
            part.accept(visitor);
        }
        visitor.visit(this);
    }
}

public class KeyBoard implements ComputerPart {
    @Override
    public void accept(Visitor visitor) {visitor.visit(this);}
}

public class Monitor implements ComputerPart {
    @Override
    public void accept(Visitor visitor) {visitor.visit(this);}
}

public class Mouse implements ComputerPart {
    @Override
    public void accept(Visitor visitor) {visitor.visit(this);}
}

public interface Visitor {
    public void visit(Computer computer);
    public void visit(KeyBoard keyBoard);
    public void visit(Mouse mouse);
    public void visit(Monitor monitor);
}

public class ConcreteVisitor implements Visitor {
    @Override
    public void visit(Computer computer) {
        System.out.println("Displaying Computer");
    }
    @Override
    public void visit(KeyBoard keyBoard) {
        System.out.println("Displaying KeyBoard");
    }
    @Override
    public void visit(Mouse mouse) {
        System.out.println("Displaying Mouse");
    }
    @Override
    public void visit(Monitor monitor) {
        System.out.println("Displaying Monitor");
    }
}
```