

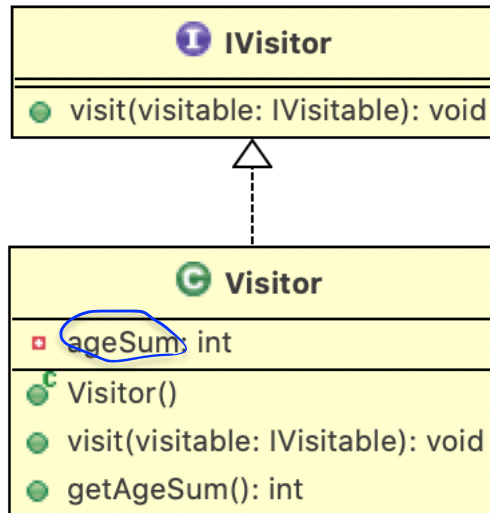
# 방문자 패턴

# 방문자 패턴이란?

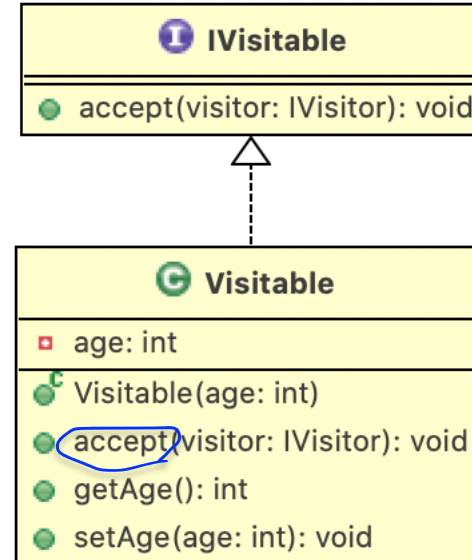
---

실제 로직을 가지고 있는 객체(Visitor)가 로직을 적용할 객체(Element)를 방문하면서 실행하는 **패턴**

# 방문자 패턴 Diagram



실제 로직(ageSum)을  
가지고 있는 객체



로직을 적용할 객체(age)

# 방문자 패턴의 목적

---

구조를 수정하지 않고 방문자를 추가하여 새로운 로직을 적용할 수 있다.

```
public interface IVisitor {  
    public void visit(IVisitable visitable);  
}
```

```
public interface IVisitable {  
    public void accept(IVisitor visitor) ;  
}
```

```
public class Visitable implements IVisitable {  
  
    private int age;  
  
    public Visitable(int age) {  
        this.age = age;  
    }  
    @Override    visitor를 받아들임  
    public void accept(IVisitor visitor) {  
        visitor.visit(this);  
    }    visitable 클래스 객체 전체를 넘겨줌  
    public int getAge() {  
        return age;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
}
```

```
public class Visitor implements IVisitor {  
  
    private int ageSum;  
  
    public Visitor() {  
        ageSum = 0;  
    }  
  
    @Override    가져온 this (visitable의 객체) -부모클래스로 묵시적 형변환  
    public void visit(IVisitable visitable) {  
        if(visitable instanceof Visitable) {  
            visitable이 여러개일경우를 위해 instanceof로 체크  
            ageSum += ((Visitable)visitable).getAge();  
            자식클래스로 다시 명시적 형변환  
        }  
    }  
  
    public int getAgeSum() {  
        return ageSum;  
    }  
}
```

```
public class VisitorMain {  
  
    public static void main(String[] args) {  
  
        ArrayList<IVisitable> visitables = new ArrayList<>();  
        visitables.add(new Visitable(1));  
        visitables.add(new Visitable(2));  
        visitables.add(new Visitable(3));  
        visitables.add(new Visitable(4));  
        visitables.add(new Visitable(5));  
  
        IVisitor visitor = new Visitor();  
  
        for(IVisitable visitable : visitables) {  
            visitable.accept(visitor);  
        }    visitor가 모든 visitable을 방문하며 계산 수행  
  
        System.out.println(((Visitor) visitor).getAgeSum());  
    }  
}
```



감사합니다