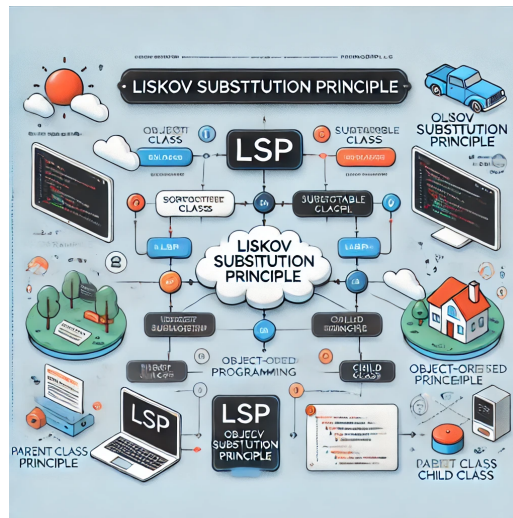


LSP(Liskov Substitution Principle)



자식 클래스는 부모 클래스의 역할을 대체할 수 있어야 함

1. LSP

- 설명 : 올바른 상속을 위해, 자식 객체의 확장이 부모 객체의 방향을 온전히 따르도록 권고하는 원칙
- 중요성 :
 - 유지보수성 : 기존 코드를 변경하지 않아도 새로운 기능을 추가하거나 기존 기능을 개선 할 수 있음
 - 재사용성 : 부모 클래스의 동작을 그대로 이어 받으면 코드 재사용성이 증가함
 - 확장성 : 시스템을 확장하기가 더 쉬워짐

2. 코드 예시

- LSP 위반 코드

```
public class Rectangle{
    protected int width;
    protected int height;
    public int getWidth(){
        return width;
    }
    public int getHeight(){
        return height;
    }
    public void setWidth(int width){
        this.width = width;
    }
    public void setHeight(int height){
        this.height = height;
    }
    public int getArea(){
        return width * height;
    }
}
```

```
public class Square extends Rectangle{
    public void setWidth(int width){
        super.setWidth(width);
        super.setHeight(getWidth());
    }
    public void setHeight(int height){
        super.setHeight(height);
        super.setWidth(getHeight());
    }
}
```

```

    }
}

```

```

public class Main{
    public static void main(String[] args){
        Rectangle rectangle = new Rectangle();
        rectangle.setWidth(10);
        rectangle.setHeight(5);
        Rectangle square = new Square();
        square.setWidth(10);
        square.setHeight(5);
        System.out.println(rectangle.getArea()); //50
        System.out.println(square.getArea()); //25
    }
}

```

- LSP 준수 코드

```

public class Shape{
    protected int width;
    protected int height;
    public Shape(int width, int height){
        this.width = width;
        this.height = height;
    }
    public int getWidth(){
        return width;
    }
    public int getHeight(){
        return height;
    }
    public int getArea(){
        return width * height;
    }
    public void setHeight(int height) {
        this.height = height;
    }
    public void setWidth(int width) {
        this.width = width;
    }
}

```

```

//직사각형 클래스
public class Rectangle extends Shape{
    public Rectangle(int width, int height){
        super(width, height);
    }
}

```

```

//정사각형 클래스
public class Square extends Shape{
    public Square(int length){
        super(length, length);
    }
}

```

```

public class Main{
    public static void main(String[] args){
        Shape rectangle = new Rectangle(10, 5);
        Shape square = new Square(5);
        System.out.println(rectangle.getArea());
        System.out.println(square.getArea());
    }
}

```