

2025년 상반기 K-디지털 트레이닝

Proxy - 필요해지면 만든다

[KB] IT's Your Life

✓ Proxy

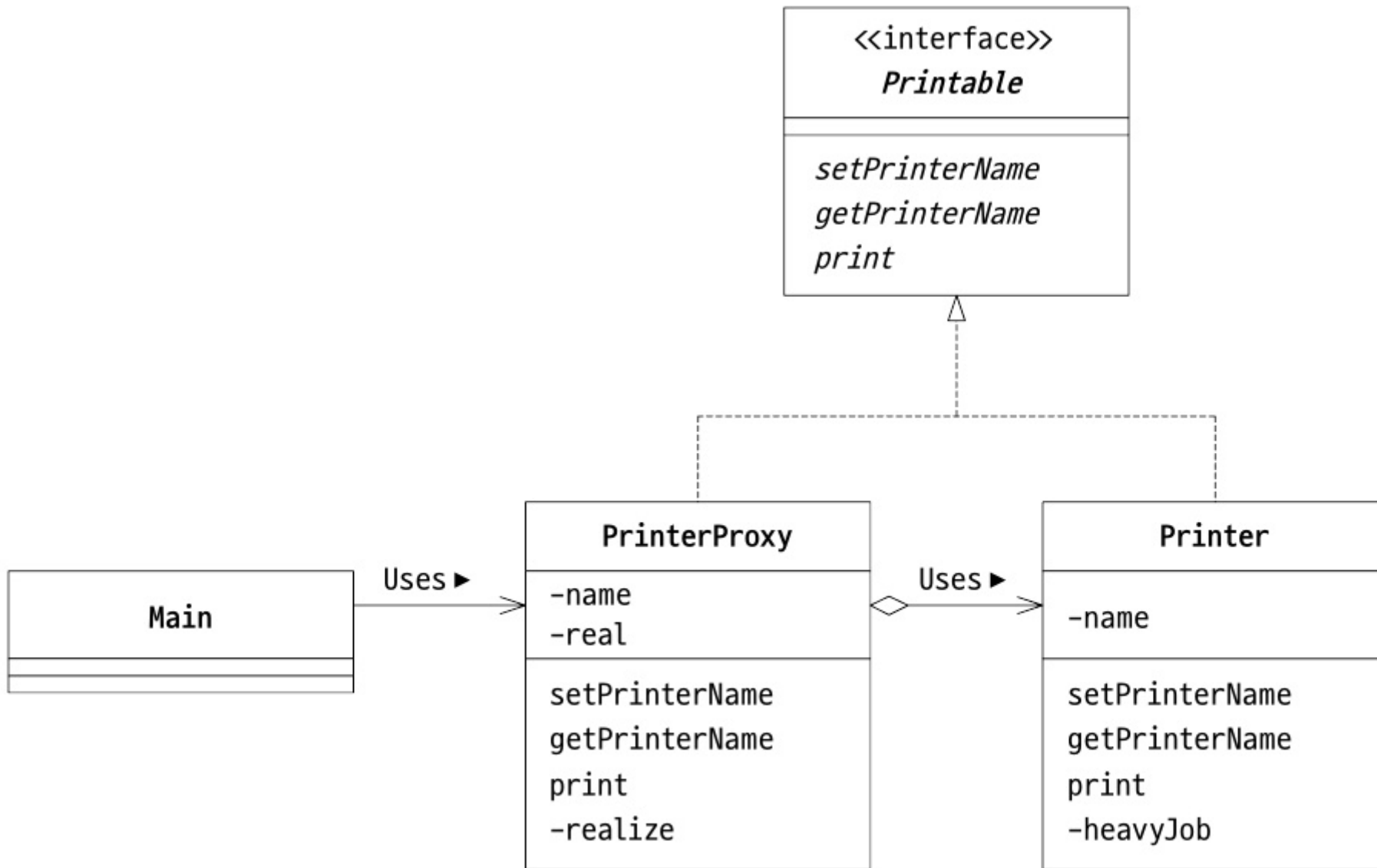
○ 대리인

- 일을 해야 할 본인을 대신하는 사람

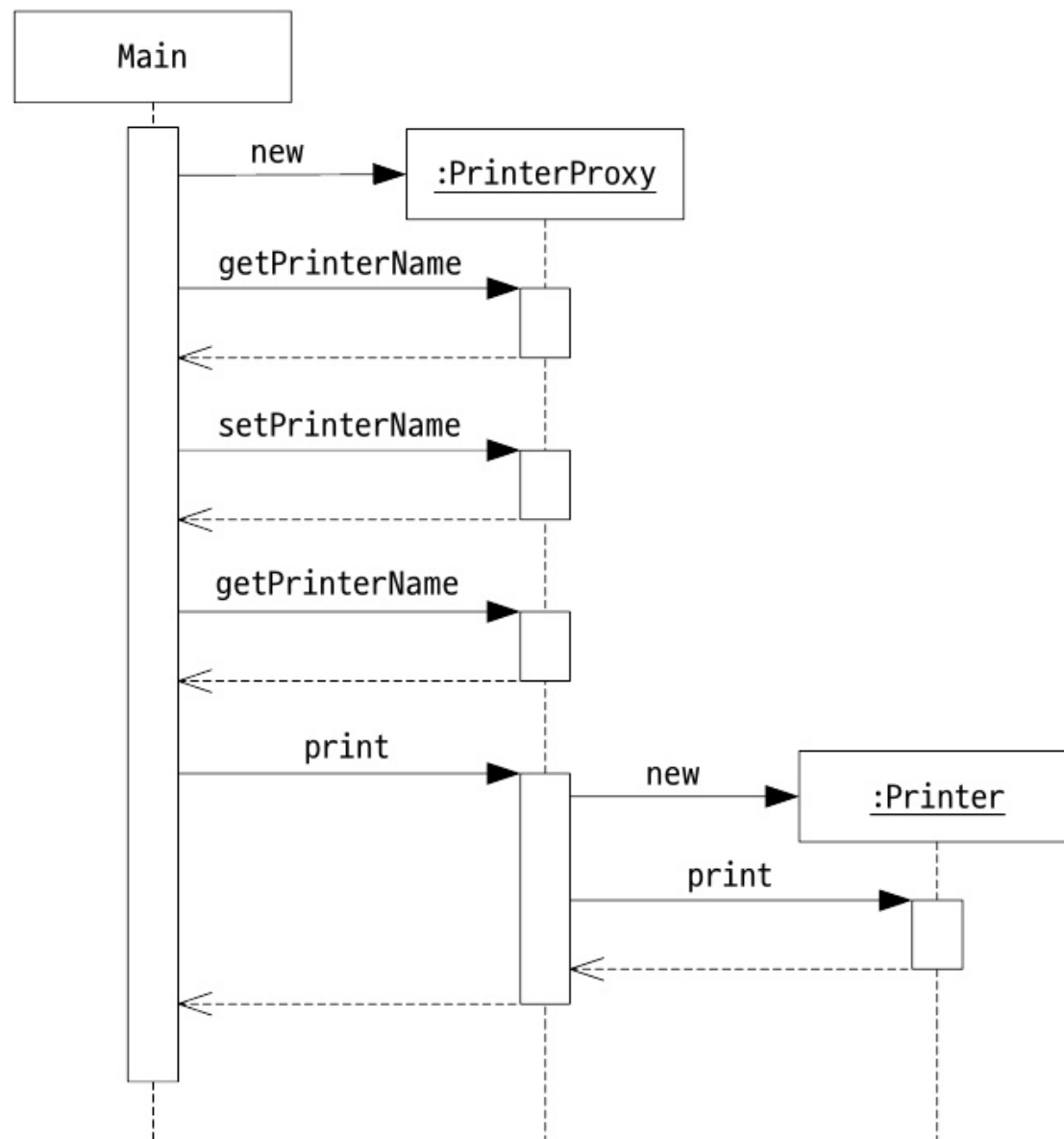
- ✓ 예제 프로그램
 - 이름 붙인 프린터

이름	해설
Printer	이름 붙인 프린터를 나타내는 클래스(본인)
Printable	Printer와 PrinterProxy의 공통 인터페이스
PrinterProxy	이름 붙인 프린터를 나타내는 클래스(대리인)
Main	동작 테스트용 클래스

✓ 예제 프로그램 클래스 다이어그램



시퀀스 다이어그램



Printable.java

```
public interface Printable {  
    void setPrinterName(String name);    // 이름 설정  
    String getPrinterName();             // 이름 취득  
    void print(String string);           // 문자열 표시(프린트 아웃)  
}
```

Printer.java

```
public class Printer implements Printable {
    private String name;    // 이름

    public Printer() {
        heavyJob("Printer 인스턴스 생성 중");
    }

    public Printer(String name) {
        this.name = name;
        heavyJob("Printer 인스턴스(" + name + ") 생성 중");
    }

    private void heavyJob(String msg) {
        System.out.print(msg);
        for(int i=0; i<5; i++) {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {

            }
            System.out.print(".");
        }
        System.out.println("완료");
    }
}
```

Printer.java

```
@Override
public void setPrinterName(String name) {
    this.name = name;
}

@Override
public String getPrinterName() {
    return name;
}

@Override
public void print(String string) {
    System.out.println("==" + name + "===");
    System.out.println(string);
}
}
```


PrintProxy.java

```
public class PrintProxy implements Printable{
    private String name ;    // 이름
    private Printer real;    // 실체

    public PrintProxy() {
        this.name = "No Name";
        this.real = null;
    }

    public PrintProxy(String name) {
        this.name = name;
        this.real = null;
    }

    @Override
    public void setPrinterName(String name) {
        if(real != null) {
            real.setPrinterName(name);
        }
        this.name = name;
    }
}
```

PrintProxy.java

```
@Override
public String getPrinterName() {
    return name;
}

@Override
public void print(String string) {
    realize();
    real.print(string);
}

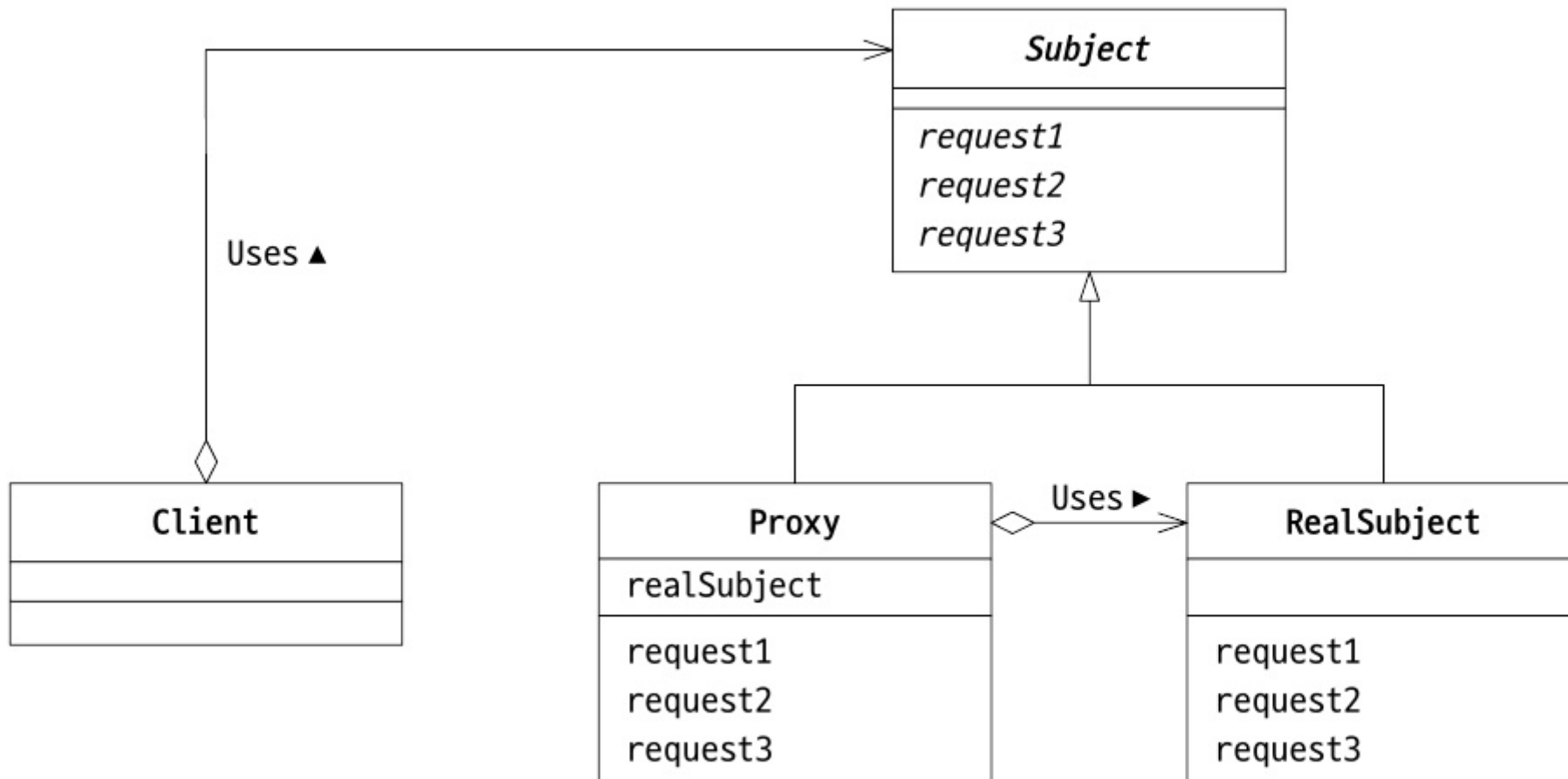
private void realize() {
    if(real == null) {
        real = new Printer(name);
    }
}
}
```

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Printable p = new PrintProxy("Alice");  
        System.out.println("이름은 현재 " + p.getPrinterName() + "입니다.");  
        p.setPrinterName("Bob");  
        p.print("Hello, world");  
    }  
}
```

```
이름은 현재 Alice입니다.  
Printer 인스턴스(Bob) 생성 중.....완료  
===Bob===  
Hello, world
```

✓ Proxy 패턴의 클래스 다이어그램



✓ Proxy 패턴

○ 대리인을 사용해 속도 올리기

- 실체가 다른 컴퓨터(다른 네트워크)에 있는 경우, 실체의 내용을 캐싱하여 바로 리턴
- HTTP 프록시

○ 대리자와 위임

- Proxy가 처리할 수 있는 일은 직접 수행(대리),
- Proxy가 처리못 하는 일은 본인이 수행(위임)

○ 투과적이란?

- 본인의 모습 그대로를 proxy가 가지는 것