

2025년 상반기 K-디지털 트레이닝

Facade - 단순한 창구를 만든다

[KB] IT's Your Life

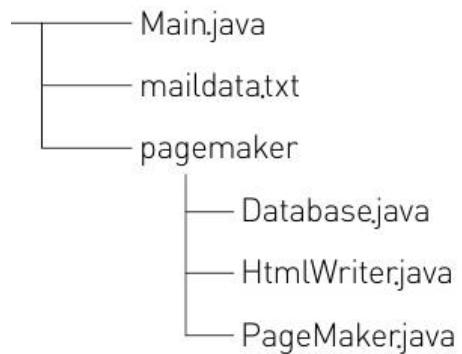
✓ Façade(파사드) 패턴

- 건물의 정면이라는 뜻
 - 뒤에 건물의 복잡한 구조가 있음
- 많은 객체들이 복잡하게 얽혀서 너저분한 세부 내용을 정리하여 높은 수준의 인터페이스(API)를 제공
- 시스템 외부에 간단한 인터페이스(API)를 보여줌
- 시스템 내부의 각 클래스의 역할과 의존 관계를 고려하여 올바른 순서로 클래스를 사용하게 함

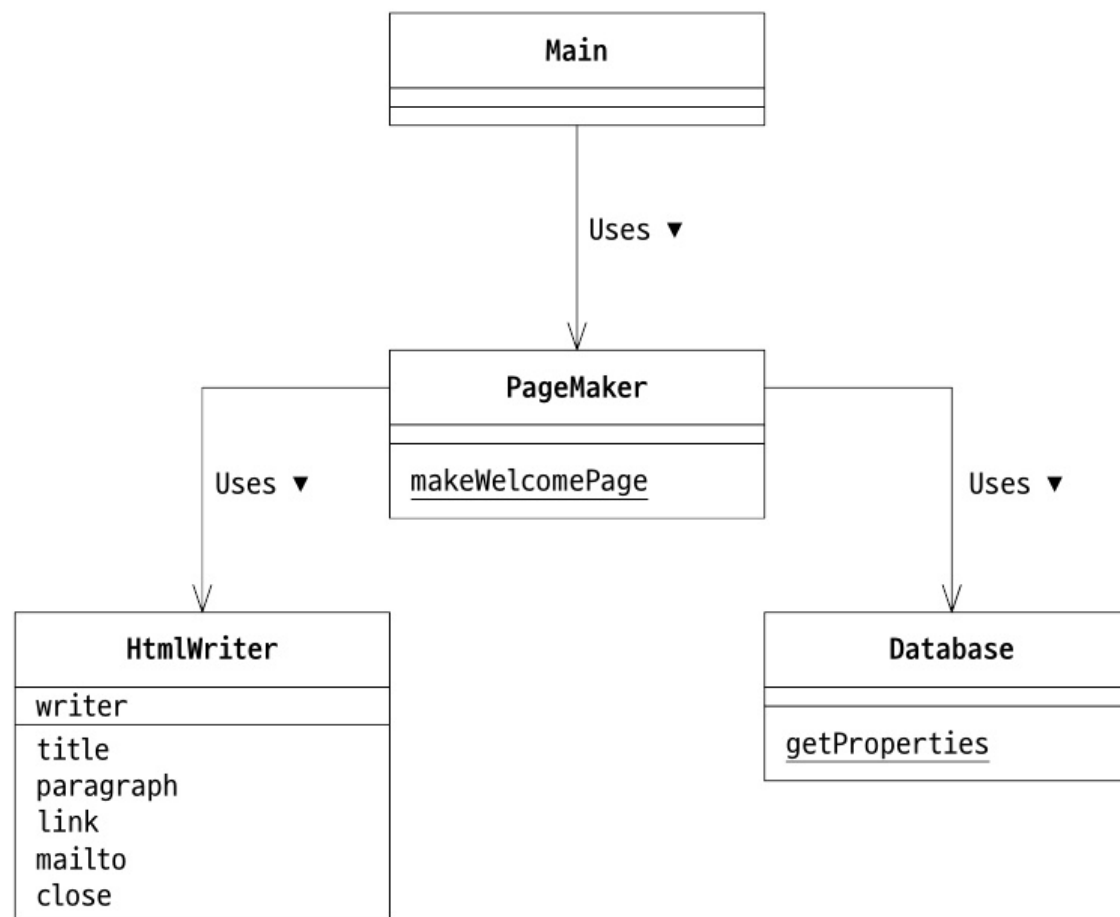
✓ 예제 프로그램

- 사용자의 웹 페이지를 작성하는 것
- 각 클래스의 역할

패키지	이름	설명
pagemaker	Database	이메일 주소에서 사용자 이름을 얻는 클래스
pagemaker	HtmlWriter	HTML 파일을 작성하는 클래스
pagemaker	PageMaker	이메일 주소로 사용자의 웹 페이지를 작성하는 클래스
이름 없음	Main	동작 테스트용 클래스



✓ 예제 프로그램의 클래스 다이어그램



maildata.txt

```
hyuki@example.com=Hiroshi Yuki  
youngjin@example.com=Kim Youngjin  
dojun@example.com=dojun  
gildong@example.com=Hong Gildong
```

pagemaker/Database.java

```
public class Database {  
    private Database() {  
    }  
  
    // 데이터베이스 이름에서 Properties를 얻는다  
    public static Properties getProperties(String dbname) throws IOException {  
        String filename = dbname + ".txt";  
        Properties prop = new Properties();  
        prop.load(new FileReader(filename));  
        return prop;  
    }  
}
```

pagemaker/HtmlWriter.java

```
public class HtmlWriter {
    private Writer writer;

    public HtmlWriter(Writer writer) {
        this.writer = writer;
    }

    // 타이틀 출력
    public void title(String title) throws IOException {
        writer.write("<!DOCTYPE html>");
        writer.write("<html>");
        writer.write("<head>");
        writer.write("<title>" + title + "</title>");
        writer.write("</head>");
        writer.write("<body>");
        writer.write("\n");
        writer.write("<h1>" + title + "</h1>");
        writer.write("\n");
    }

    // 단락 출력
    public void paragraph(String msg) throws IOException {
        writer.write("<p>" + msg + "</p>");
        writer.write("\n");
    }
}
```

pagemaker/HtmlWriter.java

```
// 링크 출력
public void link(String href, String caption) throws IOException {
    paragraph("<a href=\"" + href + "\">" + caption + "</a>");
}

// 이메일 주소 출력
public void mailto(String mailaddr, String username) throws IOException {
    link("mailto:" + mailaddr, username);
}

// HTML 닫기
public void close() throws IOException {
    writer.write("</body>");
    writer.write("</html>");
    writer.write("\n");
    writer.close();
}
}
```


pagemaker/PageMaker.java

```
public class PageMaker {
    private PageMaker() {}

    public static void makeWelcomePage(String mailaddr, String filename) {
        try {
            Properties mailprop = Database.getProperties("maildata");
            String username = mailprop.getProperty(mailaddr);
            HtmlWriter writer = new HtmlWriter(new FileWriter(filename));

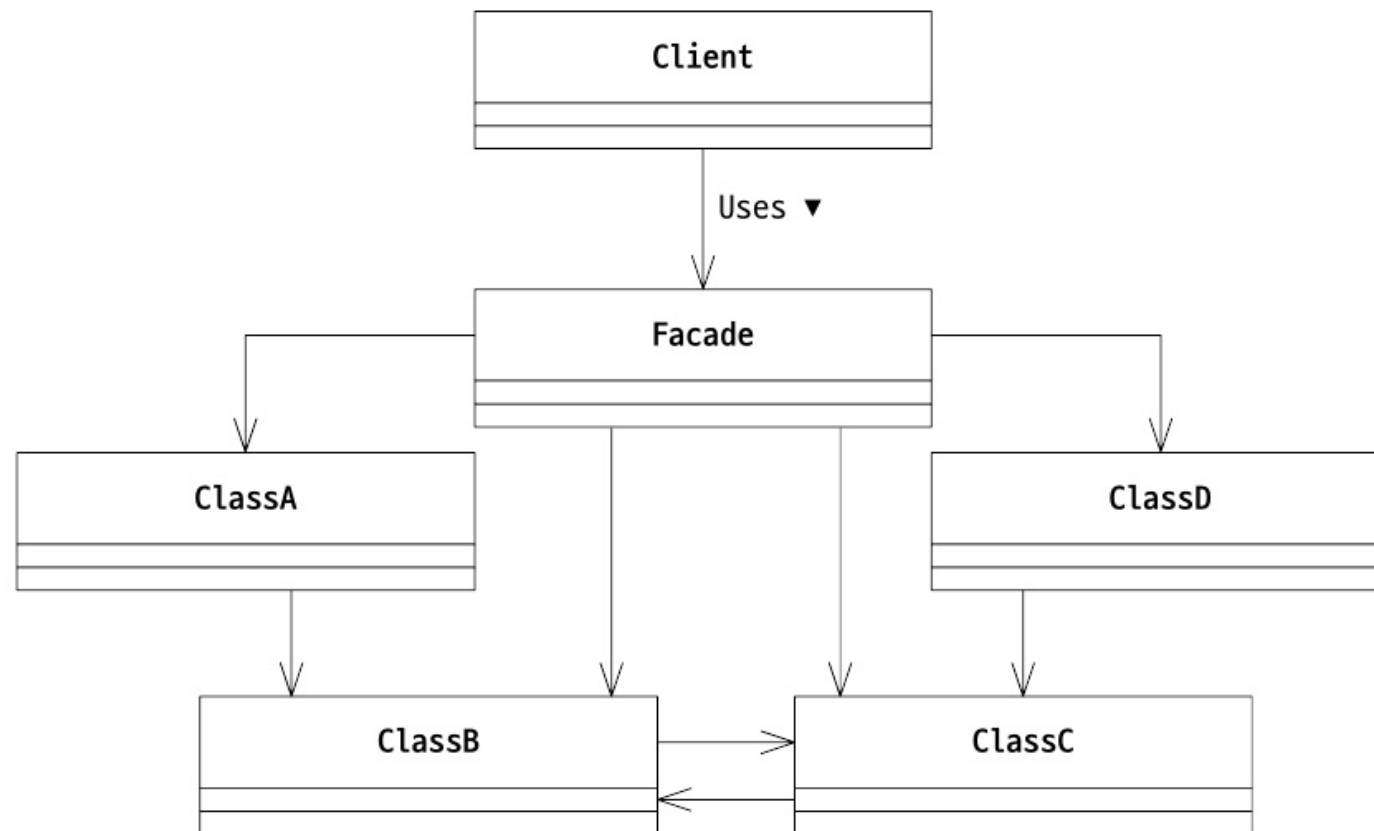
            writer.title(username + "'s web page");
            writer.paragraph("Welcome to " + username + "'s web page!");
            writer.paragraph("Nice to meet you!");
            writer.mailto(mailaddr, username);
            writer.close();
            System.out.println(filename + " is created for " + mailaddr + " (" + username + ")");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        PageMaker.makeWelcomePage("hyuki@example.com", "welcome.html");  
    }  
}
```

welcome.html is created for hyuki@example.com (Hiroshi Yuki)

✓ Façade 패턴의 클래스 다이어그램



✓ Façade 패턴의 역할

- 인터페이스(API) 수를 줄이는 것