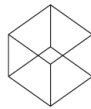
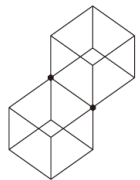


Builder 패턴



JAVA 개체 지향 디자인 패턴

UML과 GoF 디자인 패턴 핵심 10가지로 배우는



빌더 패턴

- ❖ 객체 생성 패턴
- ❖ 생성자의 인자가 많은 경우
- ❖ 생성자의 인자들 중에 필수적 인자와 선택적 인자가 혼합되어 있는 경우
- ❖ Immutable 객체(변경할 수 없는 객체)를 생성하고 싶은 경우

예제

```
public class Book {  
    private Long id; //필수  
    private String isbn; //필수  
    private String title;  
    private String author;  
    private int pages;  
    private String category;  
}
```

Telescoping Constructor패턴

[점층적 생성자 패턴]

- ❖ 필수 인자를 받는 생성자를 정의한 후에 선택적 인자를 추가로 받는 생성자를 계속해서 정의

```
public class Book {  
    private Long id; // 필수  
    private String isbn; // 필수  
    private String title;  
    private String author;  
    private int pages;  
    private String category;  
}
```

필수적 인자 생성자

```
public Book(Long id, String isbn) {  
    this.id = id;  
    this.isbn = isbn;  
}
```

선택적 인자를 받는 생성자

```
public Book(Long id, String isbn, String title) {  
    this.id = id;  
    this.isbn = isbn;  
    this.title = title;  
}
```

```
public Book(Long id, String isbn, String title, String author) {  
    this.id = id;  
    this.isbn = isbn;  
    this.title = title;  
    this.author = author;  
}
```

선택적 인자를 받는 생성자

```
public Book(Long id, String isbn, String title, String author,
            int pages) {
    this.id = id;
    this.isbn = isbn;
    this.title = title;
    this.author = author;
    this.pages = pages;
}
```

```
public Book(Long id, String isbn, String title, String author,
            int pages, String category) {
    this.id = id;
    this.isbn = isbn;
    this.title = title;
    this.author = author;
    this.pages = pages;
    this.category = category;
}
```

점층적 생성자 패턴의 문제점

- ❖ 인자가 많을 수록 생성자 개수도 많아진다.
- ❖ 생성자의 인자가 어떤 의미인지 파악하기 힘들다.
- ❖ 동일한 타입 인자인 경우
 - `Book book = new Book(1L, "isbn1234", "Design Pattern", "Insang Chung", 360, "CE");`
 - `Book book = new Book(1L, "isbn1234", "Insang Chung", "Design Pattern", 360, "CE");`
- ❖ 가독성이 떨어진다.

JavaBeans 패턴

❖ Setter 메소드로 각 속성의 값을 설정하는 방법

```
public Book() { }  
public void setId(Long id) { this.id = id; }  
public void setIsbn(String isbn) { this.isbn = isbn; }  
public void setTitle(String title) { this.title = title; }  
public void setAuthor(String author) { Author = author; }  
public void setPages(int pages) { this.pages = pages; }  
public void setCategory(String category) { this.category = category; }
```

❖ 가독성은 향상되나 immutable object를 만들 수 없음

Builder 패턴

```
public class Book {  
    private Long id; //필수  
    private String isbn; //필수  
    private String title;  
    private String author;  
    private int pages;  
    private String category;  
  
    public static class BookBuilder {  
        ...  
    }  
}
```

Builder 패턴

```
public static class BookBuilder {
    private Long id; //필수
    private String isbn; //필수
    private String title;
    private String author;
    private int pages;
    private String category;

    public BookBuilder(Long id, String isbn) {
        this.id = id;
        this.isbn = isbn;
    }

    public BookBuilder title(String title) { this.title = title; return this; }
    public BookBuilder author(String author) { this.author = author; return this; }
    public BookBuilder pages(int pages) { this.pages = pages; return this; }
    public BookBuilder category(String category) { this.category = category; return this; }
    public Book build() {
        Book book = new Book();
        book.id = this.id;
        book.isbn = this.isbn;
        book.author = this.author;
        book.title = this.title;
        book.pages = this.pages;
        book.category = this.category;
        return book;
    }
}
```

Builder 패턴

- ❖ 가독성 개선
- ❖ 메소드 체인닝
- ❖ Immutable 객체 생성

```
Book book = new Book.BookBuilder(1L, "isbn1234")  
    .author("insangchung")  
    .pages(360).category("CE")  
    .title("Design Pattern")  
    .build();
```

Lombok @Builder 어노테이션 사용

❖ 롬복 다운로드 <https://projectlombok.org/download>

❖ Getter/Setter/Builder 자동생성

@Getter

@Builder

```
public class LombokBook {  
    private Long id; //필수  
    private String isbn; //필수  
    private String title;  
    private String author;  
    private int pages;  
    private String category;  
}
```

Builder 패턴

```
LombokBook book = new LombokBook.LombokBookBuilder()  
    .id(1L)  
    .isbn("isbn1234")  
    .author("insang chung")  
    .pages(360)  
    .category("CE")  
    .title("Design Pattern")  
    .build();
```

Mandatory 필드

```
@Getter
@Builder
public class LombokBook {
    @NonNull
    private Long id; //필수
    @NonNull
    private String isbn; //필수
    private String title;
    private String author;
    private int pages;
    private String category;
}
```