

DI





# 교재(1 ~ 3, 10)



# Design Pattern

---

- ❖ Template Method Pattern
- ❖ Factory Method Pattern
- ❖ Strategy Pattern
- ❖ Template/Callback Pattern



# IoC(DI)

---

- ❖ Inversion Of Control
- ❖ Dependency Injection
- ❖ Dependency Lookup

# 요구사항

---

- ❖ 이미 저장되어있는 사용자 정보를 ID 를주면 이름과 패스워드를 가지고 오는 프로그램 만들어줘.



# 요구사항

---

- ❖ DB에 직접 사용자 등록하니까 불편해.
- ❖ 사용자 추가기능 만들어줘

# 요구사항

---

- ❖ 현재는 D사에서 사용하고있는데..
- ❖ N사에다가 납품해야한데..
- ❖ 거기는 DB 가 달라..
- ❖ 어떻게하지?



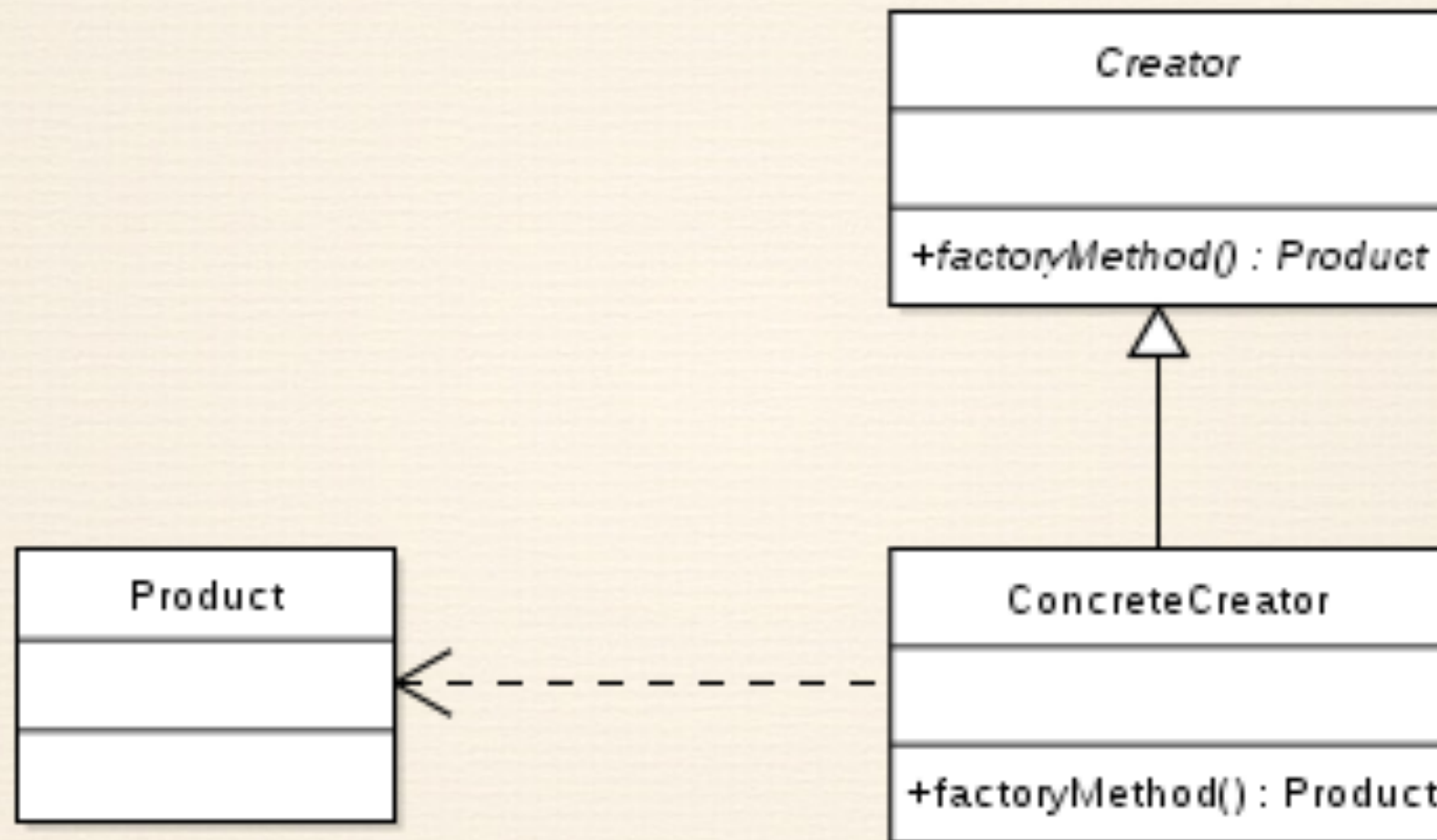
# Template Method Pattern

---

```
public abstract class Super {  
  
    public void templateMethod() {  
        //기본 알고리즘 코드  
        hookMethod();  
        abstractMethod;  
        ...  
    }  
  
    protected void hookMethod() { }  
    public abstract void abstractMethod();  
}  
  
public class Sub1 extends Super {  
    protected void hookMethod() {  
        ...  
    }  
    public void abstractMethod() {  
        ...  
    }  
}
```



# Factory Method Pattern



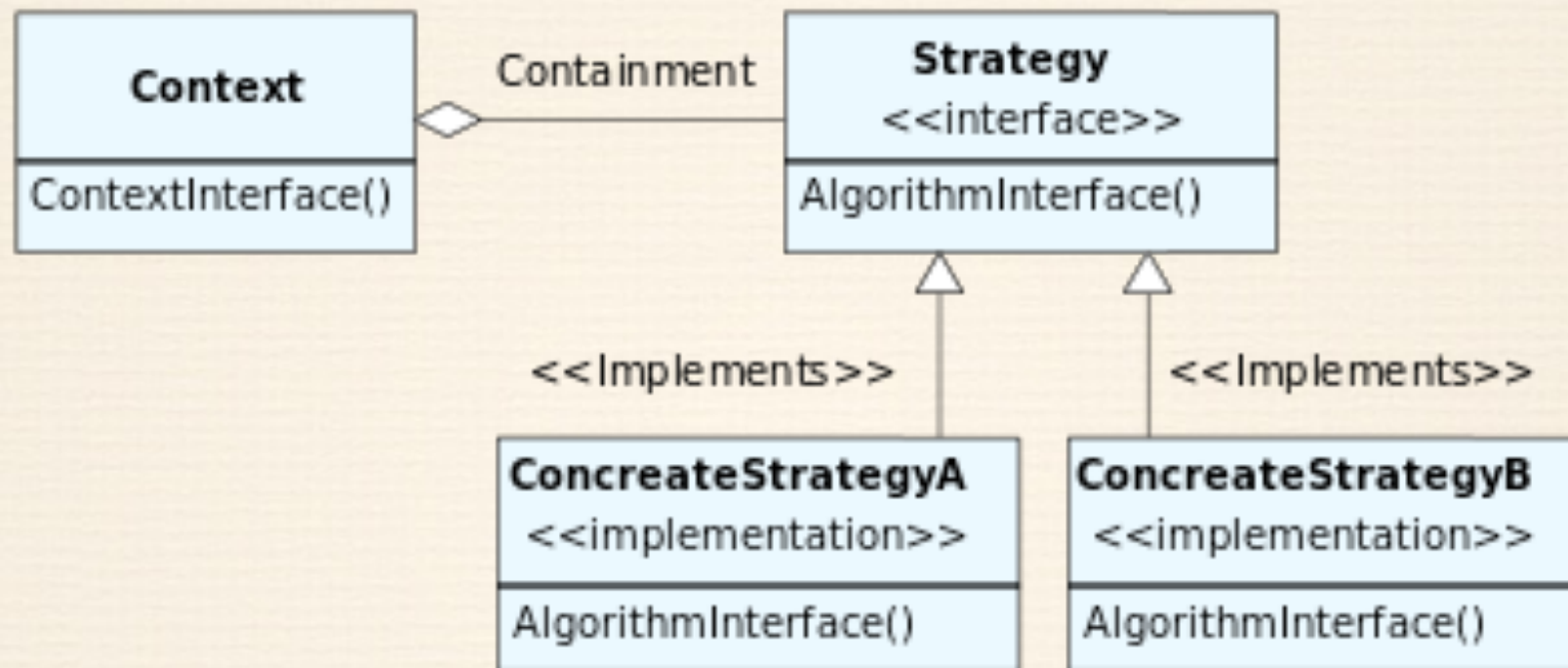


[illegible]

❖ 상품을 조회,등록했으면 좋겠어.



# Strategy Pattern





# Spring Container, Bean

---

- ❖ DaoFactory => Spring Container (Application Context)
- ❖ Bean : Spring Container 가 관리하는 java object (UserDao)



# 장애

---

- ❖ dao 를 사용하는데, 가끔 서버가 죽어 어떻게 된거야?



# 요구사항

---

- ❖ 사용자 삭제하는거 만들어줘
- ❖ 사용자 수정하는거 만들어줘



# JdbcTemplate

---

- ❖ `jdbcTemplate.update()`
- ❖ `jdbcTemplate.queryForObject()`
- ❖ ...



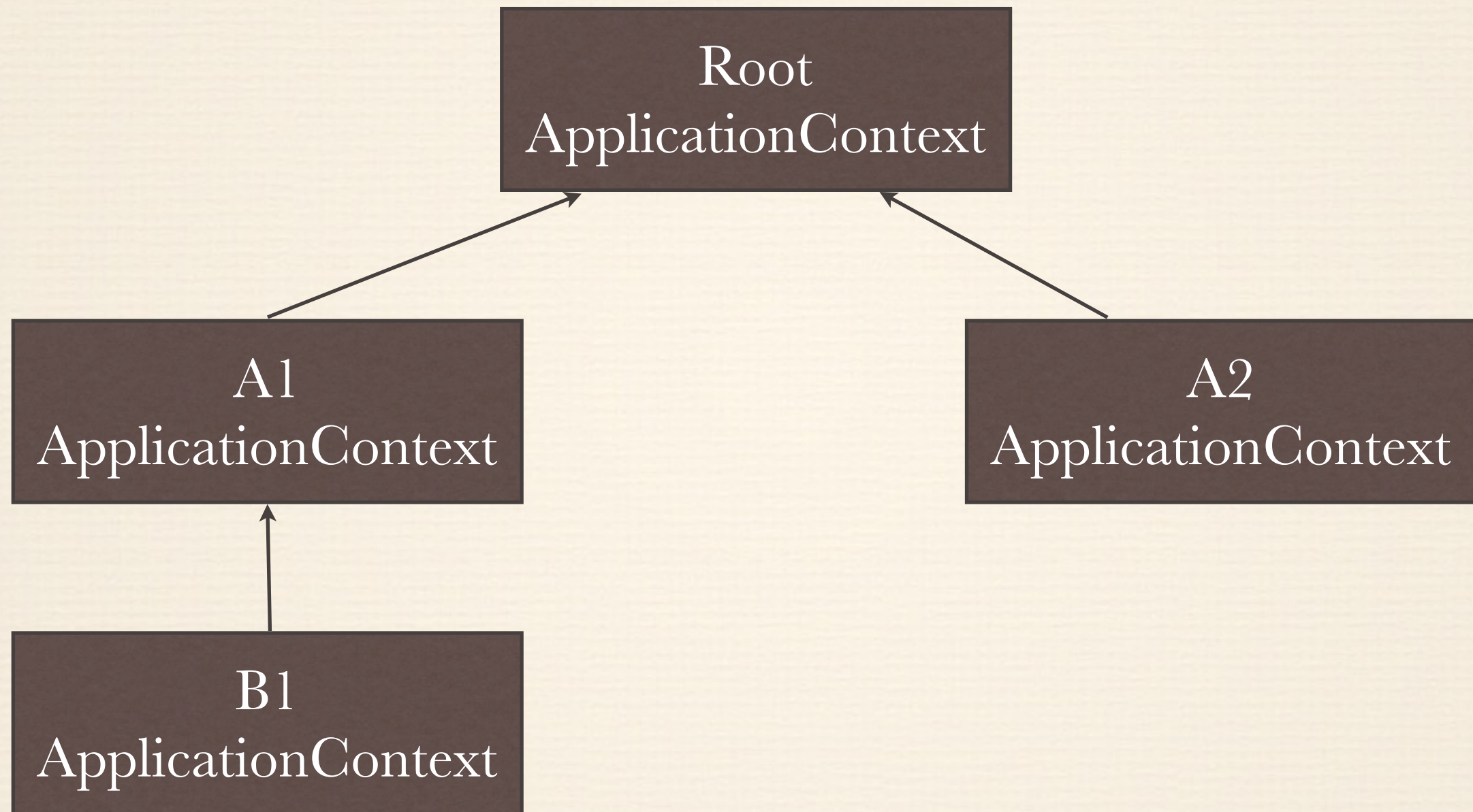
# ApplicationContext

---

- ❖ StaticApplicationContext
- ❖ GenericXmlApplicationContext
- ❖ AnnotationConfigApplicationContext
- ❖ 각각의 WebApplicationContext
  - ❖ StaticWebApplicationContext
  - ❖ XmlWebApplicationContext
  - ❖ AnnotationConfigWebApplicationContext



# ApplicationContext 계층구조





# Bean 등록

---

- ❖ `<bean> ... </bean>`
- ❖ `@Configuration ... @Bean ... @Bean`
- ❖ `@Component @Controller @Service @Repository`



# DI 설정

---

- ❖ `<property> ... </property>`
- ❖ `<constructor-arg> ... </constructor-arg>`
- ❖ `@Resource(name="jdbcTemplate")`
- ❖ `@Autowired`



# Value 설정

---

- ❖ `<property name="name" value="value" />`
- ❖ `@Value`
  - ❖ `@Value("value") String str;`
  - ❖ `@Value("1.2") double rate;`
  - ❖ `@Value("1,2,3,4") int[] intarr;`



# Value 설정

---

- ❖ `<property name="names">`
  - ❖ `<list>`
    - ❖ `<value> a </value>`
    - ❖ `<value> b </value>`
  - ❖ `</list>`
- ❖ `</property>`
- ❖ `<property name="ages">`
  - ❖ `<map>`
    - ❖ `<entry key="Kim" value="30" />`
    - ❖ `<entry key="Lee" value="35" />`
  - ❖ `</map>`
- ❖ `</property>`



# Value 설정

~~~~~

- ❖ `<property name="settings">`
  - ❖ `<props>`
    - ❖ `<prop key="username">jeju</prop>`
    - ❖ `<prop key="password">jejupw</prop>`
  - ❖ `</props>`
- ❖ `</property>`
- ❖ `<property name="beans">`
  - ❖ `<list>`
    - ❖ `<ref bean="beanA" />`
    - ❖ `<ref bean="beanB" />`
  - ❖ `</list>`
- ❖ `</property>`



# Bean Scope

---

- ❖ singleton (Default)
- ❖ prototype
- ❖ request
- ❖ session
- ❖ globalSession
- ❖ application