



소프트웨어공학

강의노트

서비스 지향(SOA),관점지향(AOP) 개발 방법론

❖ 학습안내

이번 시간의 학습내용과 학습목표를 확인해보세요.

■ 학습내용

- 서비스 지향(SOA) 개발 방법론
- 관점 지향(AOP) 개발 방법론
- 정보보호 고려사항

■ 학습목표

- 서비스 지향(SOA) 개발방법론에 정의와 사용사례에 대하여 설명할 수 있다.
- 관점 지향(AOP) 개발 방법론에 정의와 사용사례에 대하여 설명할 수 있다.
- 정보보호 고려사항에 대하여 설명할 수 있다.



❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

1. 웹서비스의 개념

- ◆ 서비스 지향(Service Oriented Architecture)
 - 필요한 소프트웨어나 정보시스템을 해당 서비스를 어떤 체계에 의하여 제공 받아서 사용하는 개념
 - 여러 부분에서 공통으로 제공될 서비스를 서비스공급자가 먼저 개발하거나, 그 부분을 제공받는다고 생각하고 전체 시스템을 개발·이용하는 웹 서비스의 개념을 사용함
 - 소프트웨어 생명주기 마지막 단계의 프로그램이나 문서 등을 이용,초기 단계의 생성물에 해당하는 정보나 문서를 만들어 내는 일
 - 설계부터 순차적으로 이루어지는 순 공학에 상대되는 개념
- ◆ 웹서비스
 - 현재의 인터넷(World Wide Web)은 단순한 웹문서(html)의 전달체계가 아니라 응용프로그램 수준의 서비스를 제공함
 - 사용자의 요청에 대해 프로그래밍 언어 및 운영체제의 제약 없이 시스템간의 동적연동을 통해 최적의 응답을 웹으로 제공하는 서비스
 - 전 세계적으로 구축되어 있는 Web 기반을 가지고 기업, 단체, 개인들 상호간의 서로의 소프트웨어 시스템 서비스, 콘텐츠 등을 주고 받고 재사용하고자 하는 서비스
 - 현재 웹 서비스를 수행하고자 하는 표준 기술로 SOAP, UDDI, WSDL의 웹 서비스의 표준 요소기술을 사용

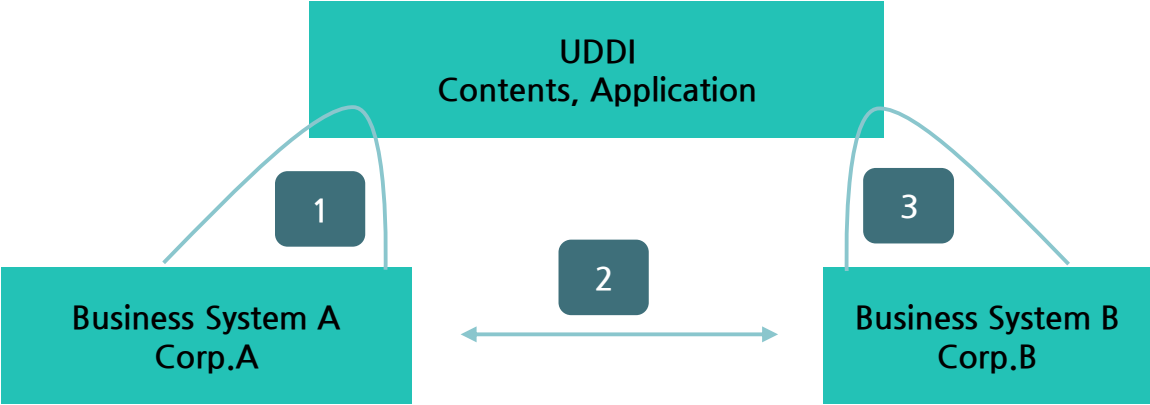
SOAP (Simple Object Access Protocol)	웹 서비스 체계에서 서로의 통신을 주고 받게 해주는 XML체계
UDDI (Universal Description, Discovery and Integration)	원하는 서비스가 어디에 있는지 알려주는 XML체계
WSDL (Web Service Description Language)	원하는 서비스를 어떤 방식으로 사용할 수 있는지 설명해 놓은 XML체계

❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

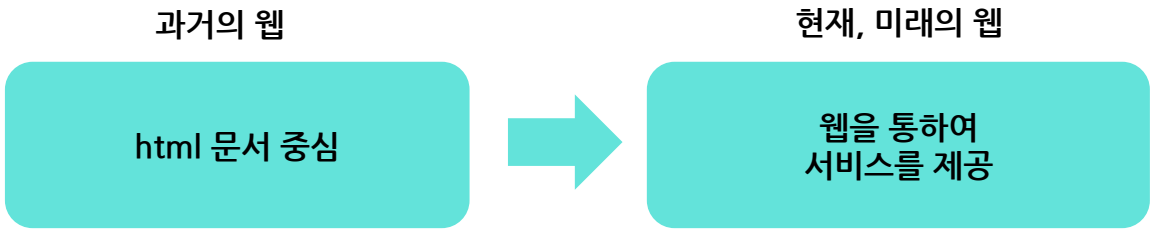
1. 웹서비스의 개념(계속)

◆ 웹서비스 흐름도



- 1 기업 A 또는 비즈니스 시스템 A는 UDDI를 통하여 서비스 X가 비즈니스 시스템 B에 있음을 **WSDL형태로** 전송 받음
- 2 기업 A는 기업 B와 **SOAP**를 이용한 **객체통신**을 요구함
- 3 기업 B는 UDDI를 통해 **기업 A에 대한 정보 습득**으로 신뢰를 가지고 서비스 X를 제공함

◆ 웹서비스 흐름



- 지금까지의 웹은 html 문서 중심적인 웹이었으나, 현재, 미래는 **웹을 통하여 서비스를 제공하는 개념으로 진화**됨
- Html문서가 결과라면 어떤 기능들을 제공하는 서비스 체계를 웹으로 제공할 것

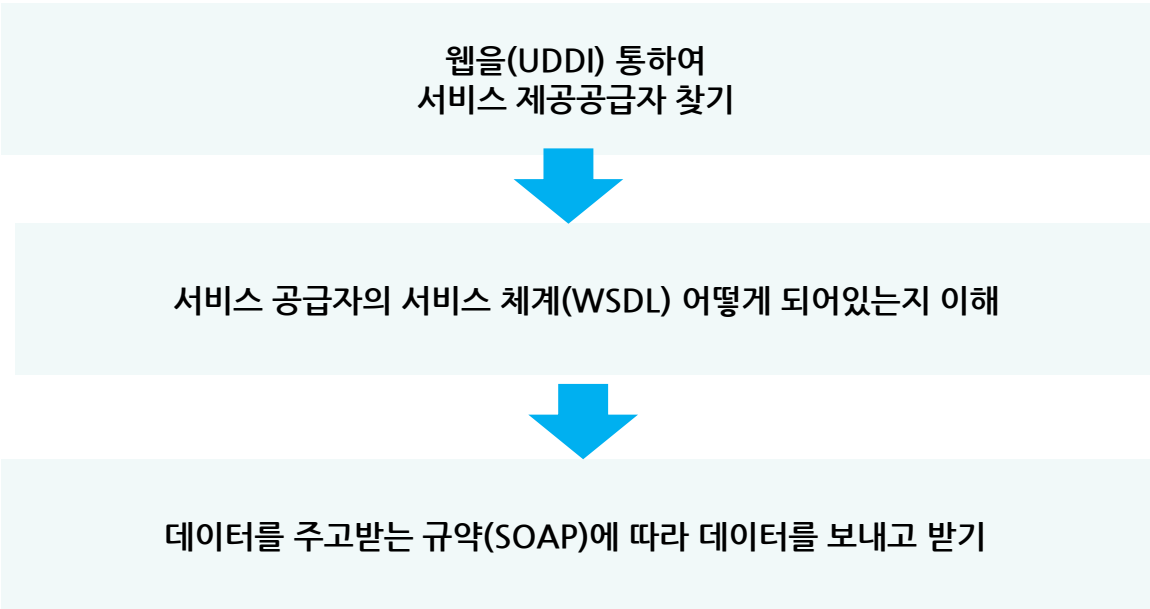
❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

1. 웹서비스의 개념(계속)

◆ 웹서비스 흐름(계속)

- [예] 회사 내 인사평가모듈이 필요할 때 웹서비스의 흐름



- [예] 버스 도착시간이나 날씨정보 같은 공공 정보를 웹 서비스로 부터 제공 받아 **스마트폰 어플**을 만드는 사례

2. 서비스 도출 방법

◆ 서비스 공학

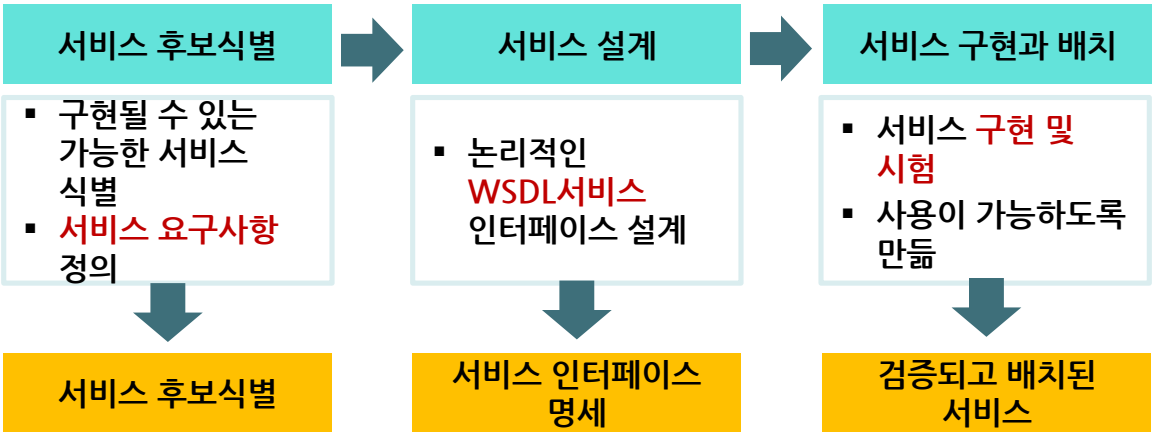
- SOA체계에서 **서비스를 개발**하는 방법임
- 웹 서비스 체계는 서비스를 개발하는 프로세스와 관련이 많음
- 웹 서비스를 구현하는 경우도 일반적인 서비스 공학에서의 **서비스 개발 절차와 동일**하게 진행됨

❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

2. 서비스 도출 방법(계속)

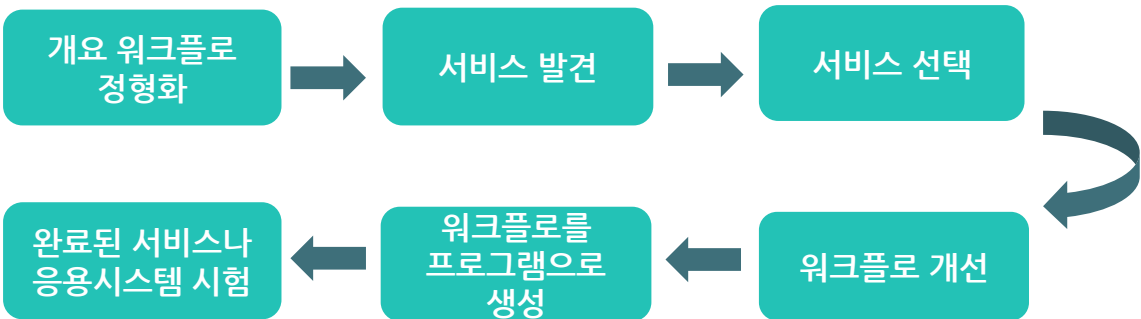
◆ SOA에서 서비스 개발 방법



3. 서비스 소프트웨어 구축 절차

◆ 소프트웨어 구축 절차

- 서비스는 시스템이나 기기 관점보다는 실제 업무나 일이 벌어지는 **처리절차 (Workflow)의 분석**을 통해 도출됨

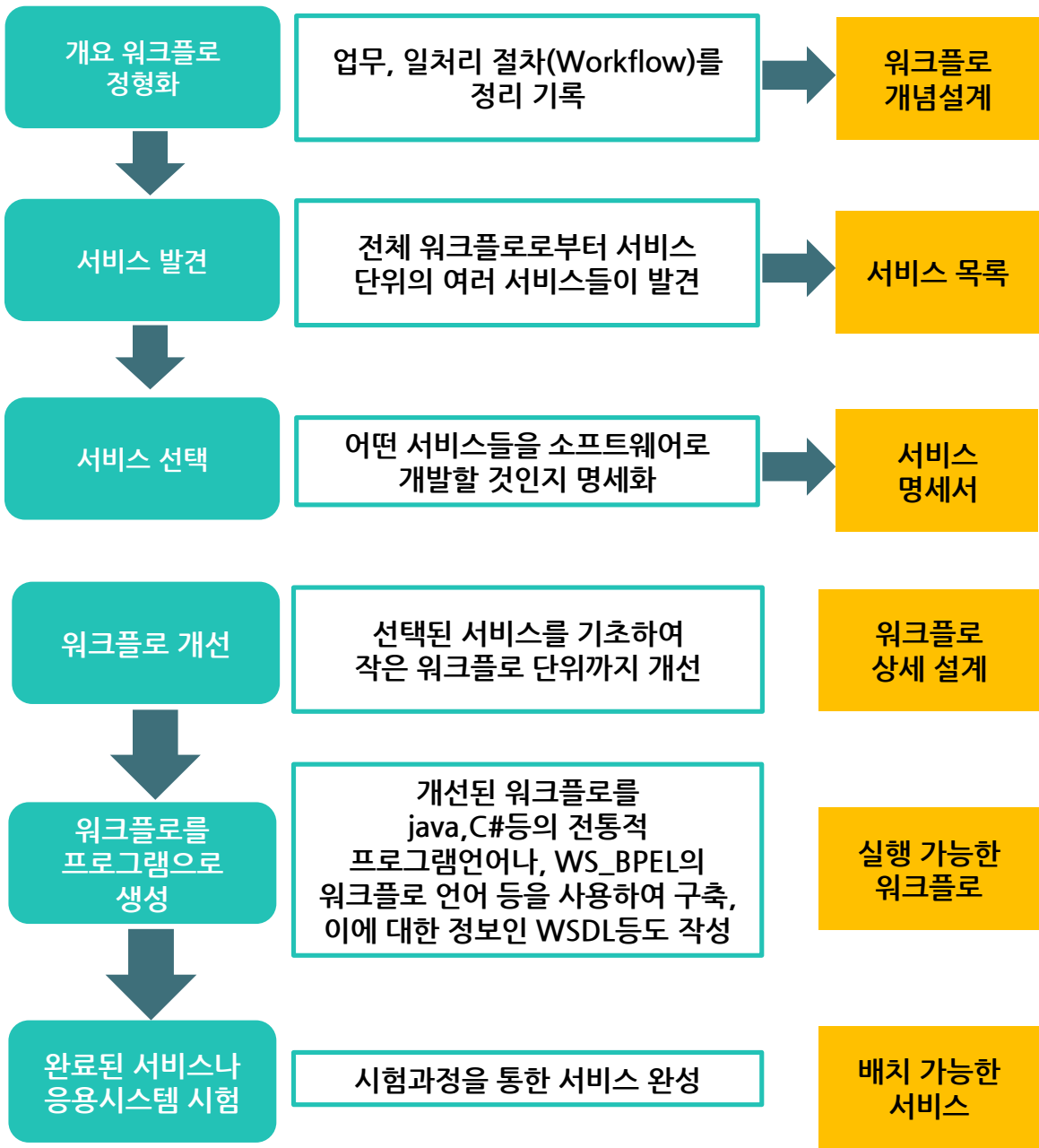


❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

3. 서비스 소프트웨어 구축 절차(계속)

◆ 소프트웨어 구축 절차(계속)



❖ 학습내용

[1] 서비스 지향(SOA) 개발 방법론

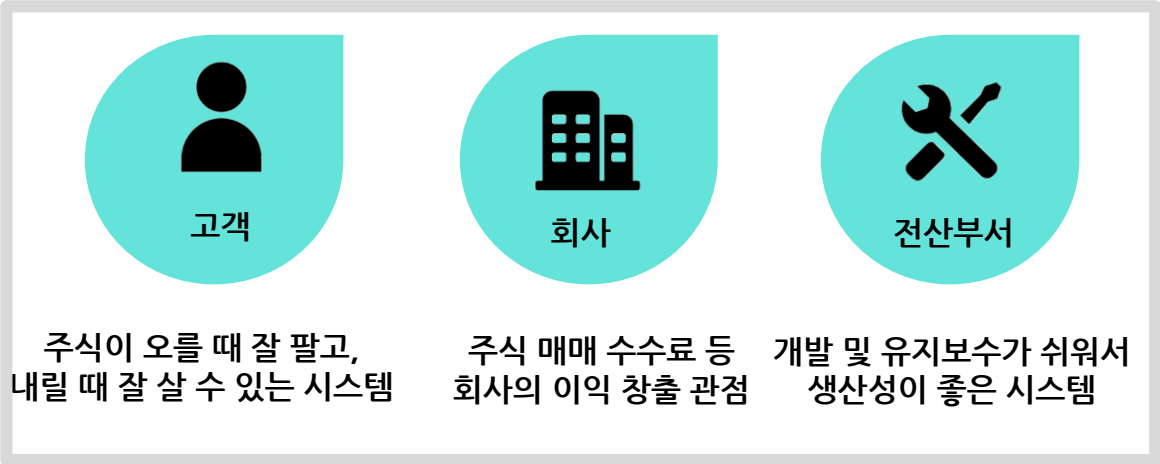
3. 서비스 소프트웨어 구축 절차(계속)

- ◆ 소프트웨어 재사용(Software Reuse)
 - SOA는 비슷하지만 다른 개념
 - 이미 개발된 소프트웨어를 일부 혹은 전체를 다시 사용하여 기존의 소프트웨어 **품질과 생산성 및 신뢰도**를 높이고 **개발 기간과 비용**을 낮추는 방법

[2] 관점지향(AOP) 개발 방법론

1. 관심사의 분리

- 관점지향 개발 방법론의 개요
 - 관점지향 소프트웨어(Aspect Oriented Software Development)
 - 중심적인 내용을 **추상화**하여 프로그램을 개발하는 기법
 - 다양한 **관점**을 프로그래밍에 녹여 넣는 기법
- ◆ 관점지향 개발 방법론의 개요
 - [예] 증권 매매 시스템: 이해 당사자마다 **시스템의 목적**에 대한 **시각(관점)**이 각각 다름

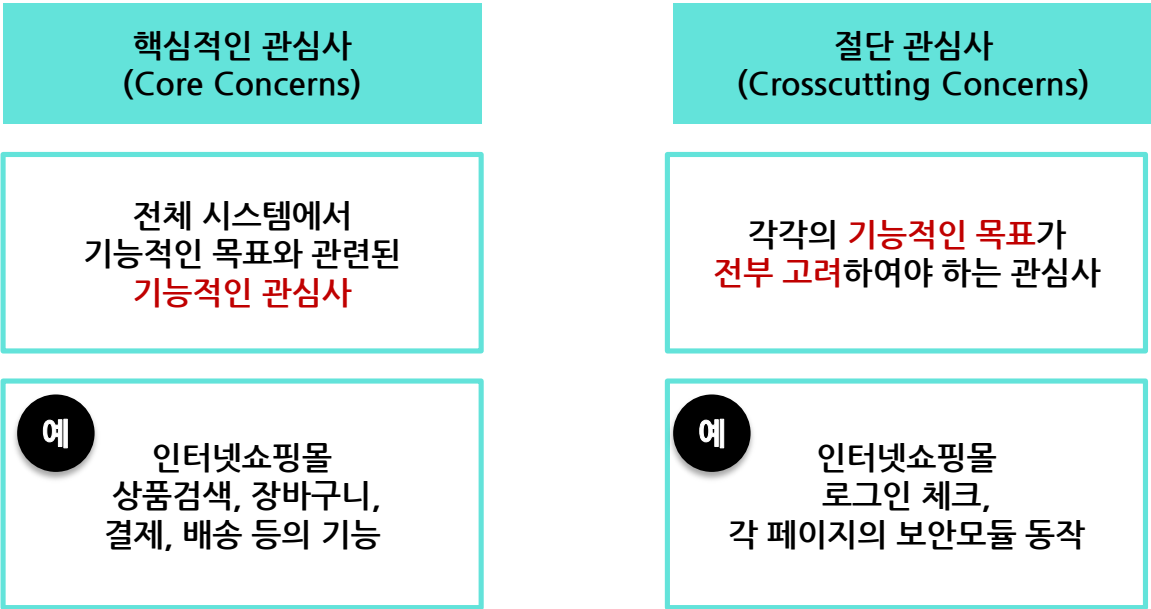


❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

1. 관심사의 분리(계속)

◆ 시스템 구축에 있어서 관심사



- ◆ 관점지향 프로그래밍 이론에서 사용하는 용어
- AOP를 직접 구현하는 프로그래밍 언어(AspectJ)에서 사용되는 용어
 - **관점지향 프로그래밍 이론**의 세부적 구현 방법을 이해하는데 중요

◆ 관점지향 프로그래밍 이론에서 사용하는 용어

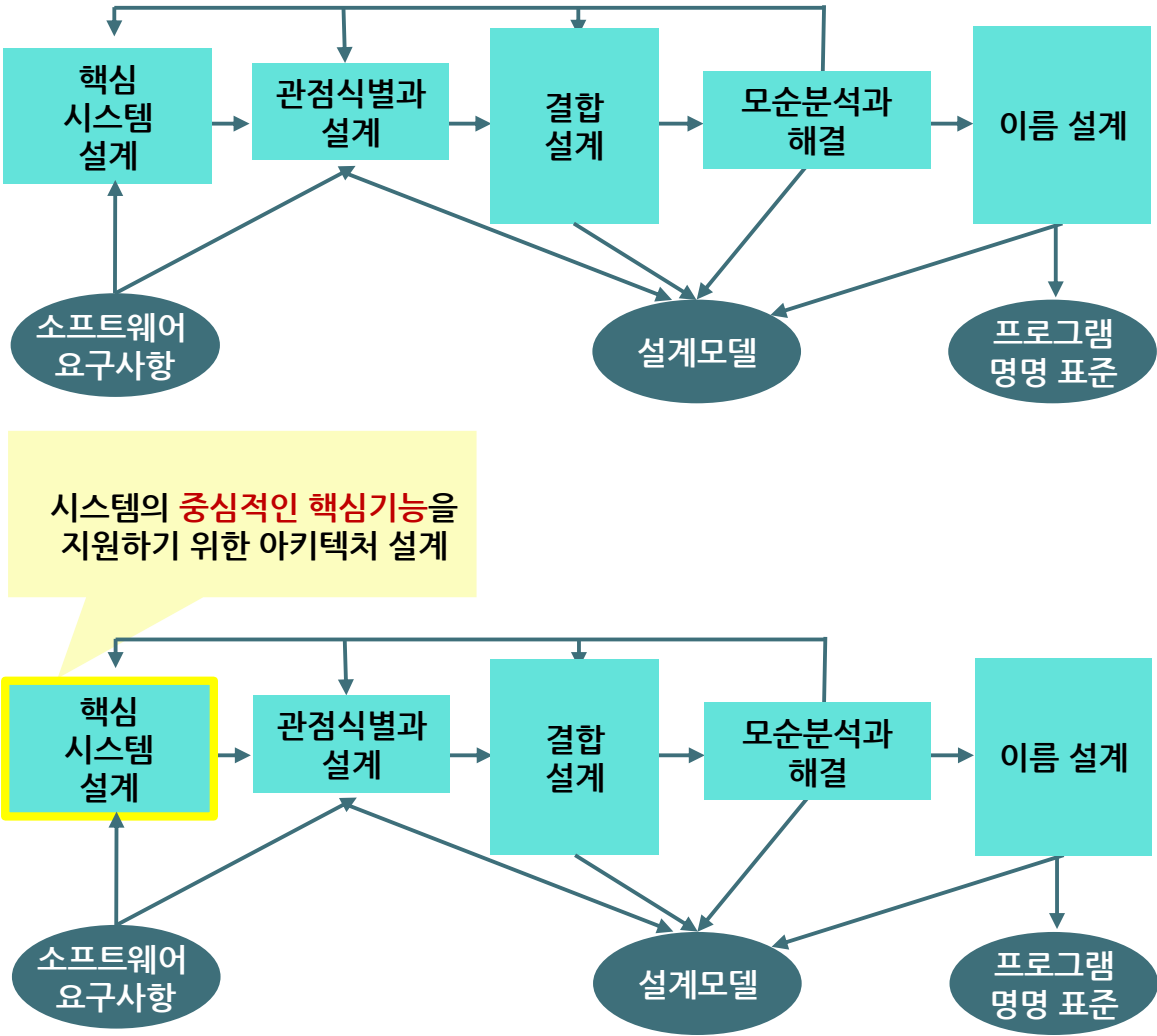
구분	내용
관점(Aspect)	관심사를 추상적으로 정의한 프로그램 코드 모음
어드바이스(Advice)	관심사를 추상적으로 정의한 프로그램 코드
포인트 컷 (Point Cut)	관점에서 포함되는 문장으로, 연관된 어드바이스들이 실행되는 결합지점 을 정의함
결합지점 (Joint Point)	관점이나 관점의 부분코드(어드바이스)가 실행될 수 있는 실행 프로그램에 있는 이벤트
결합 (Weaving)	보다 많은 관점을 조합하기 위해 결합지점에 어드바이스 코드를 통합 하는 일

❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

2. 관점을 이용한 소프트웨어 개발

- ◆ 관점을 이용한 소프트웨어 개발 절차
 - 관점을 이용한 소프트웨어 시스템 구현 시 설계단계는 일반적인 과정과 조금 다르며, 다음의 관점을 고려하여 설계한 후 진행

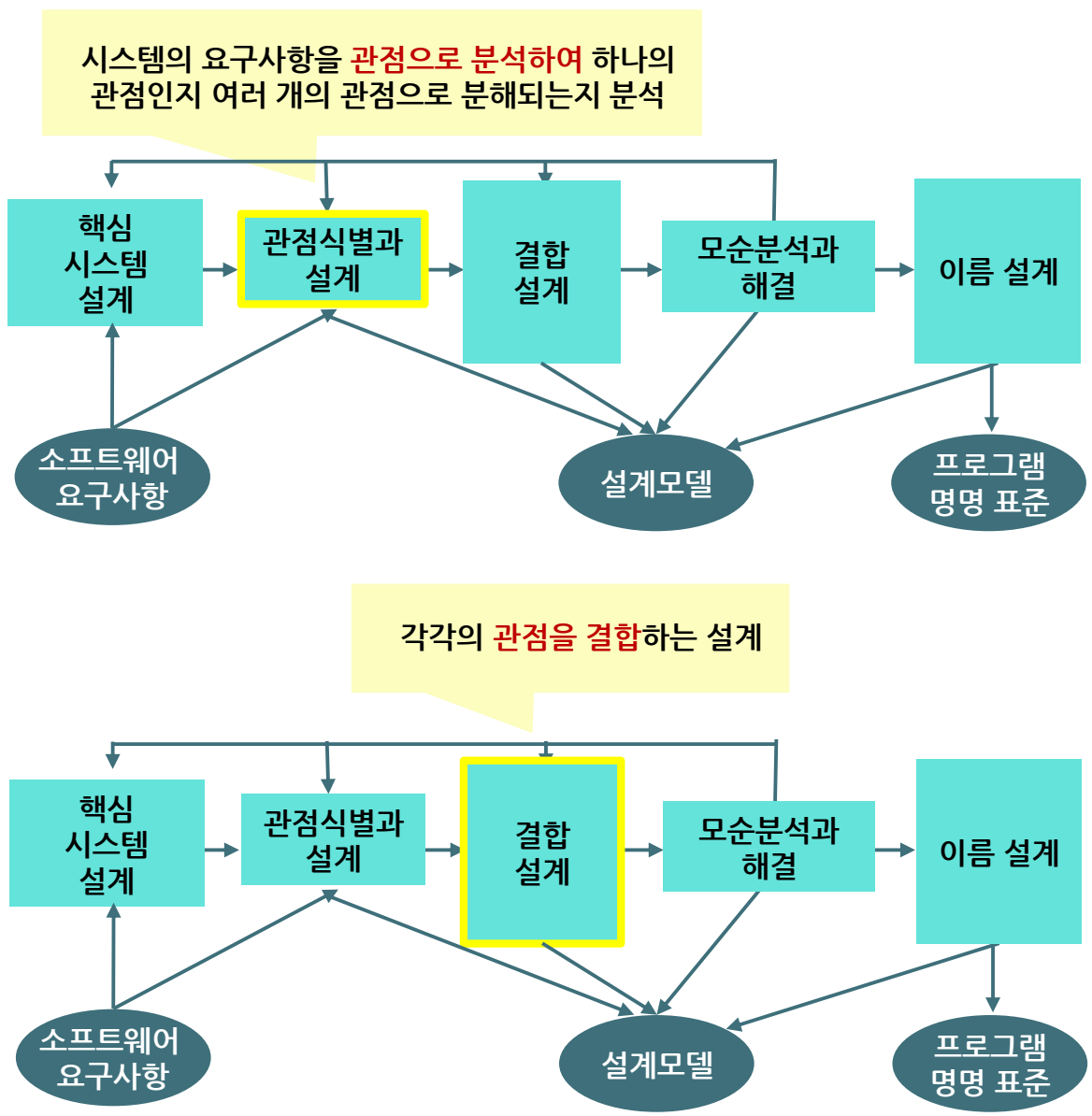


❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

2. 관점을 이용한 소프트웨어 개발(계속)

◆ 관점을 이용한 소프트웨어 개발 절차(계속)

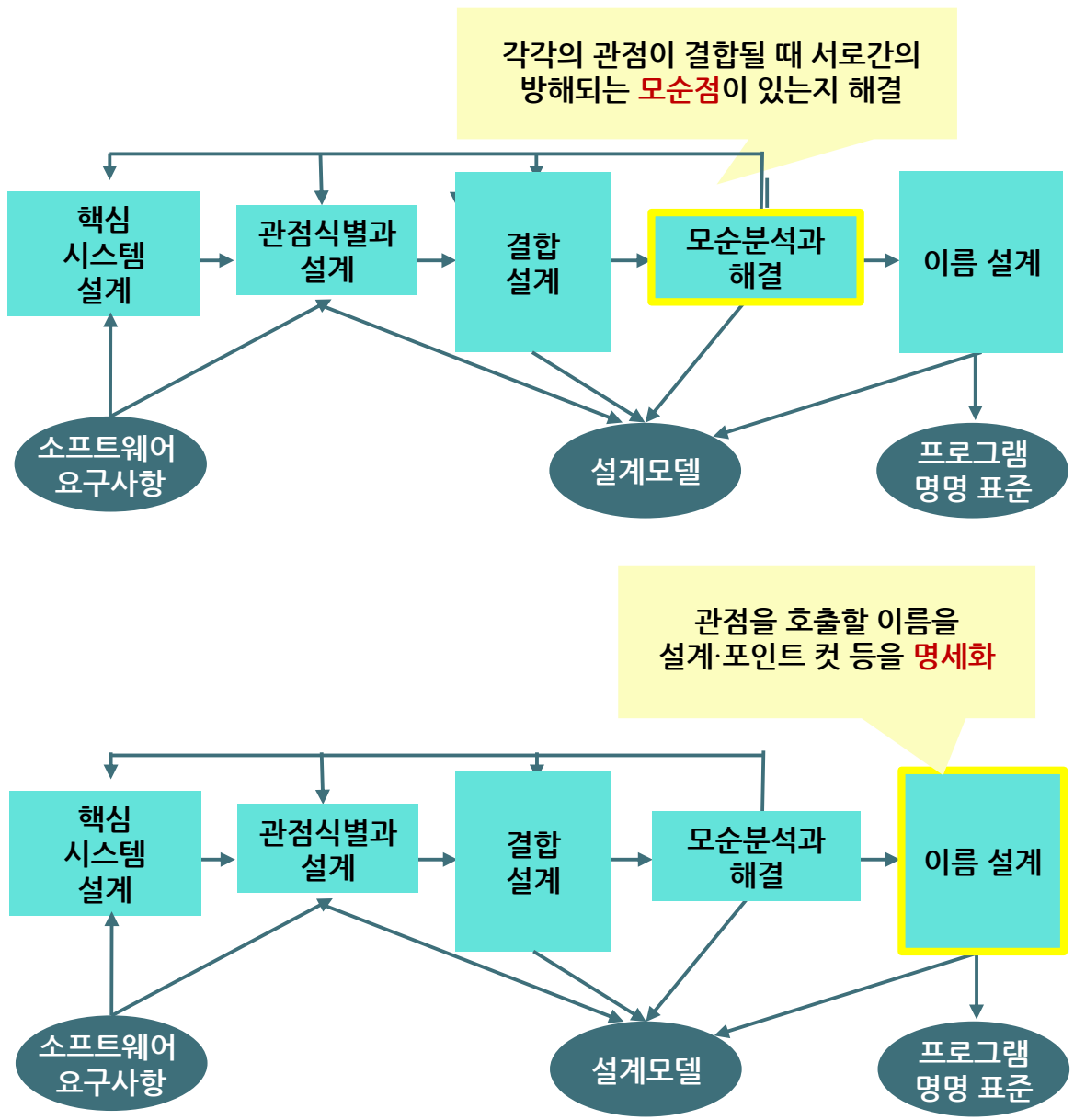


❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

2. 관점을 이용한 소프트웨어 개발(계속)

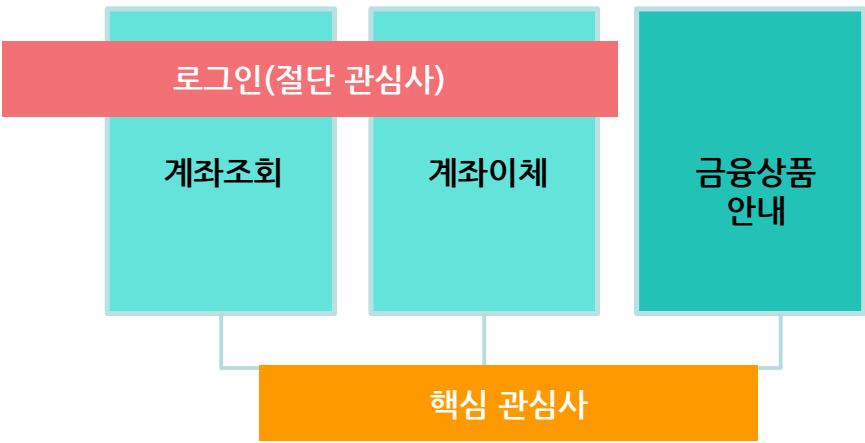
◆ 관점을 이용한 소프트웨어 개발 절차(계속)



❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

3. 관점지향 프로그래밍 사례 - 인터넷 뱅킹 시스템 구축



- 계좌조회, 계좌이체는 로그인 과정이 필요
- 금융상품안내는 로그인 없이 홈페이지 접속자에게 제시

◆ OOP에서 절단 관심사의 표현

- 인터넷 뱅킹 시스템 구축을 객체지향방식(OOP/CBD)으로 프로그래밍하는 경우 핵심 관심사 단위의 모듈 중심으로 프로그램이 구현
- 계좌조회, 계좌이체, 금융상품 안내 별 모듈(클래스 등)으로 구현함
- 하지만 로그인 체크나 로깅등의 절단 관심사에 대하여 각 모듈에 삽입(Weaving) 하여야 하기 때문에 코드가 복잡

로깅(Logging): 각 모듈의 실행시간 분석을 위해
모듈 시작과 끝에 시간을 기록, 간격 사이의 실행시간을 기록

- ◆ OOP에서 절단 관심사를 표현하기가 어려우며, 또한 일반적 구조적 프로그래밍에서 절단 관심사를 표현하기가 어렵고 복잡함
- ◆ 프로그램의 변경, 재 사용시 절단 관심사는 프로그래밍을 더욱 복잡하게 해주는 요소임

❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

3. 관점지향 프로그래밍 사례 - 인터넷 뱅킹 시스템 구축(계속)

◆ OOP에서 절단 관심사의 표현(계속)

```
Class account_inq() //계좌조회
{
  ...
}

class account_trans() //계좌이체
{
  ...
}

class financial_inst() //금융상품안내
{
  ...
}
```



절단 관심사
추가

```
class account_inq() //계좌조회
{
  if(! login_ok()) return login();
  //로그인 체크가 안 되었으면 로그인 수행
  ...
}

class account_trans() //계좌이체
{
  if(! login_ok()) return login();
  //로그인 체크가 안되 있으면 로그인 수행
  ...
}

class financial_inst() //금융상품안내
{
  ...
}
```

```
function login_ok()
{
  ...
}

function login()
{
  ...
}
```

❖ 학습내용

[2] 관점지향(AOP) 개발 방법론

3. 관점지향 프로그래밍 사례 - 인터넷 뱅킹 시스템 구축(계속)

- ◆ AOP에서 절단 관심사의 표현
 - AOP에서는 절단 관심사, 결합지점 등을 정의하여 삽입하는 방식으로 구현

```
class account_inq() //계좌조회
{
    ...
}

class account_trans() //계좌이체
{
    ...
}

class financial_inst() //금융상품안내
{
    ...
}
```



```
@aspect login_ok() //
{
    @pointcut before(execute accout_inq,account_trans)
    //
    // (join-point)
    //
```

로그인 체크모듈이
관점(절단 관심사)임을 선언

이 절단 관심사의 결합지점은(Point Cut)
accout_inq와 account_trans라는 핵심 관심사가
핵실행되기 전(Before)임을 나타냄

이렇게 단순히 절단 관심사를 정의하는 것만으로 해당
프로그램이 컴파일 또는 실행 될 때 핵심 관심사에
절단 관심사가 삽입(Weaving)되어 구성됨

❖ 학습내용

[3] 정보보호 고려사항

1. 정보보호 개념

- ◆ 정보보호
 - 정보보호(보안)는 시스템과 데이터의 기밀성, 무결성, 가용성에 대한 위협에 대하여 대응하는 모든 활동

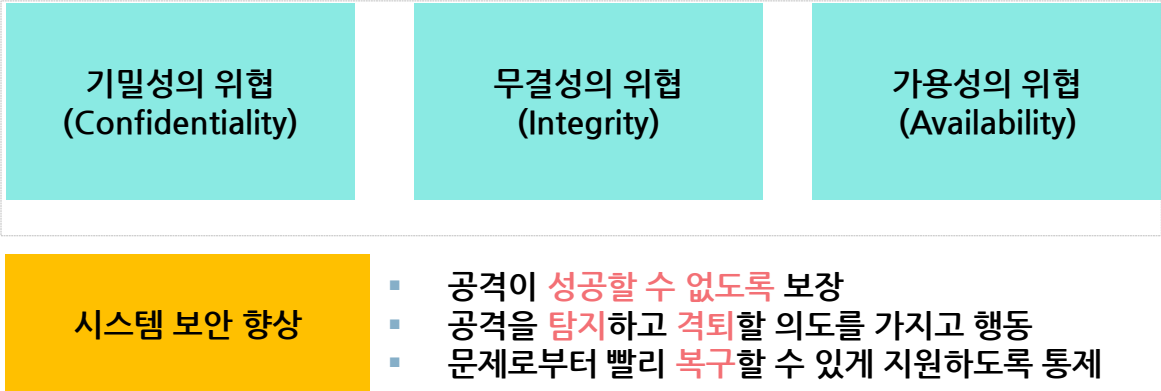


❖ 학습내용

[3] 정보보호 고려사항

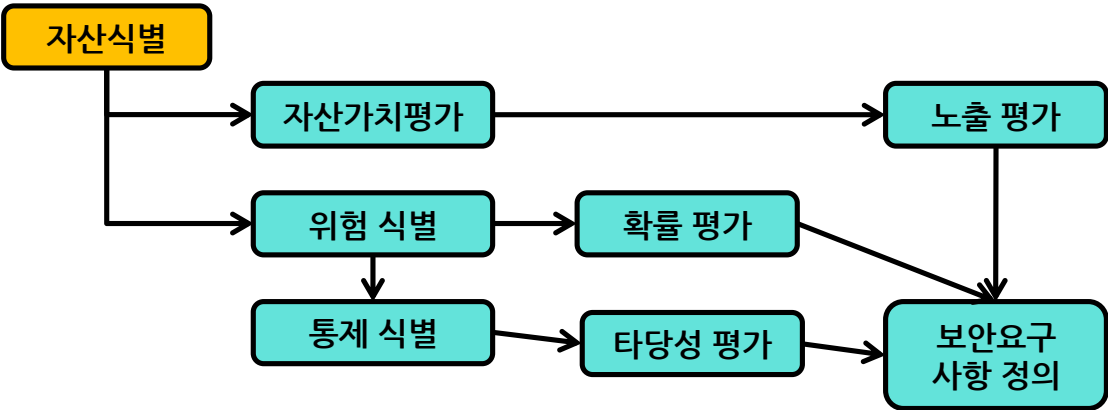
1. 정보보호 개념(계속)

- ◆ 정보보호(계속)
 - 정보보호(보안)는 시스템과 데이터의 기밀성, 무결성, 가용성에 대한 위협에 대하여 대응하는 모든 활동(계속)



2. 정보보호 관리

- ◆ 사전의 위험을 평가 하는 방법



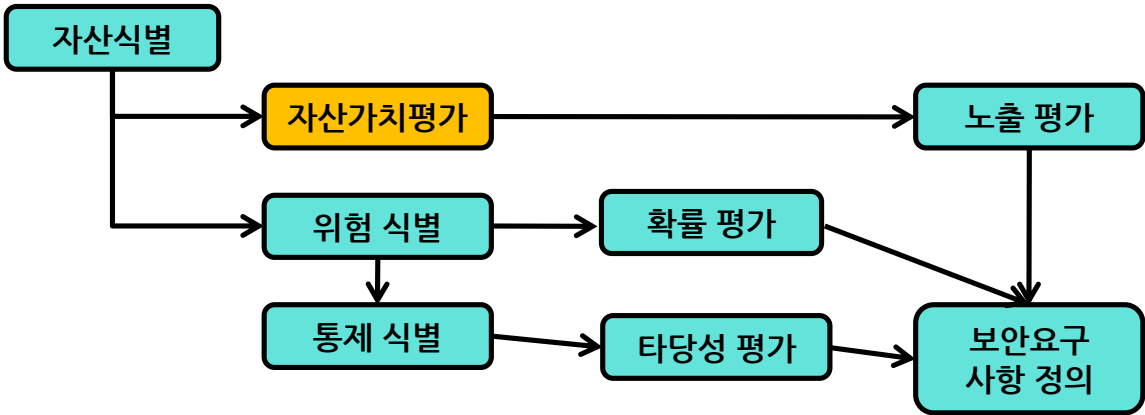
- 보호할 필요성이 있는 시스템 자산을 식별하는 단계
- 시스템에 관련된 데이터, 시스템 자체, 특정 시스템 기능도 자산으로 인식

❖ 학습내용

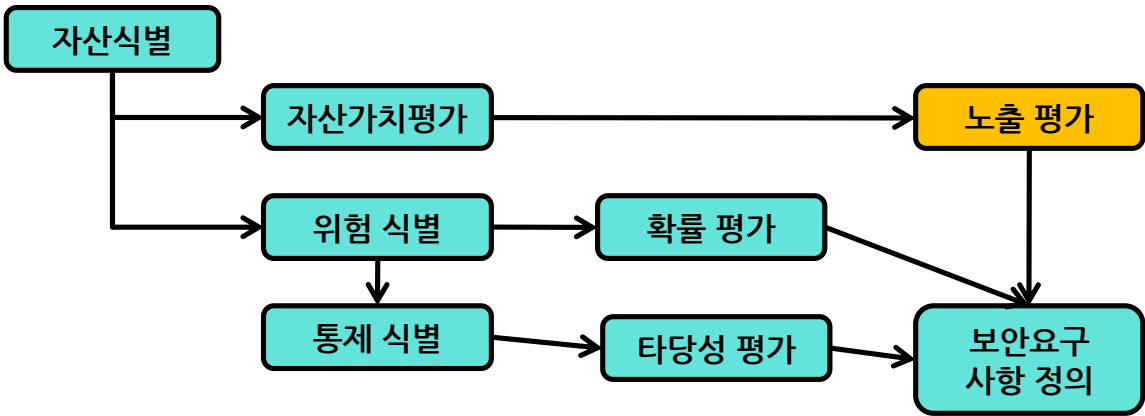
[3] 정보보호 고려사항

2. 정보보호 관리(계속)

◆ 사전의 위험을 평가 하는 방법(계속)



- 식별된 자산의 가치를 추정하는 단계



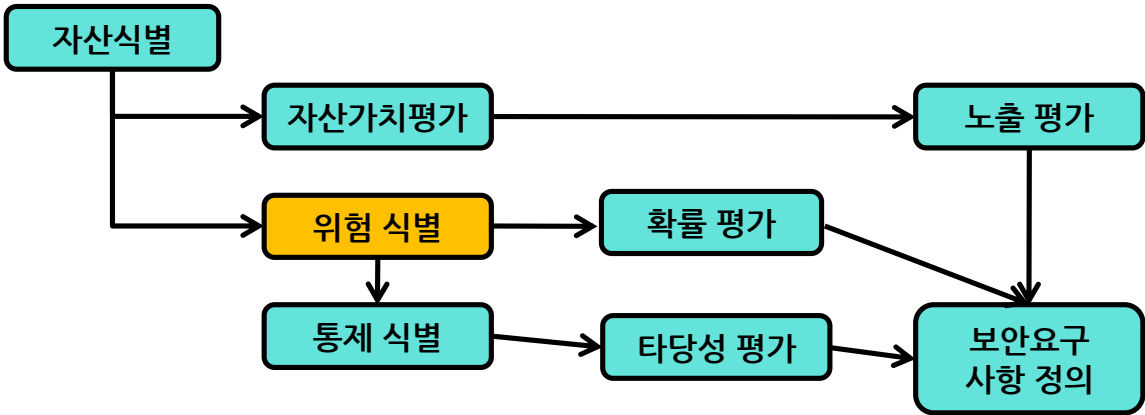
- 각 자산에 관련된 잠재적인 손실을 평가하는 단계

❖ 학습내용

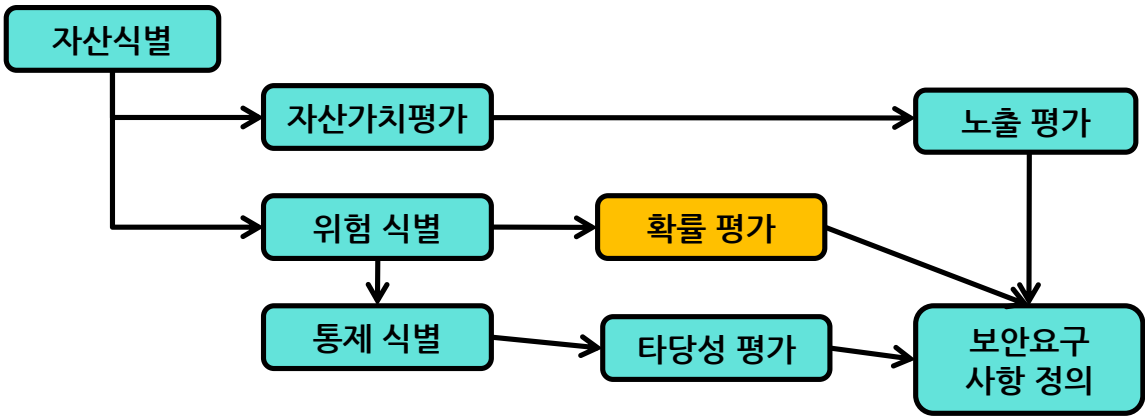
[3] 정보보호 고려사항

2. 정보보호 관리(계속)

◆ 사전의 위험을 평가 하는 방법(계속)



- 시스템 자산에 대한 **위험을 인식**하는 단계



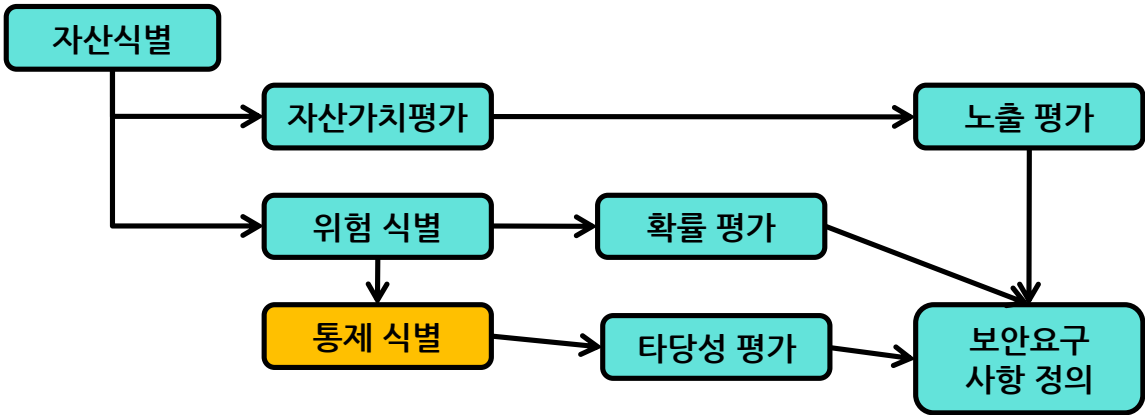
- 각 위협의 확률을 **추정**하는 단계

❖ 학습내용

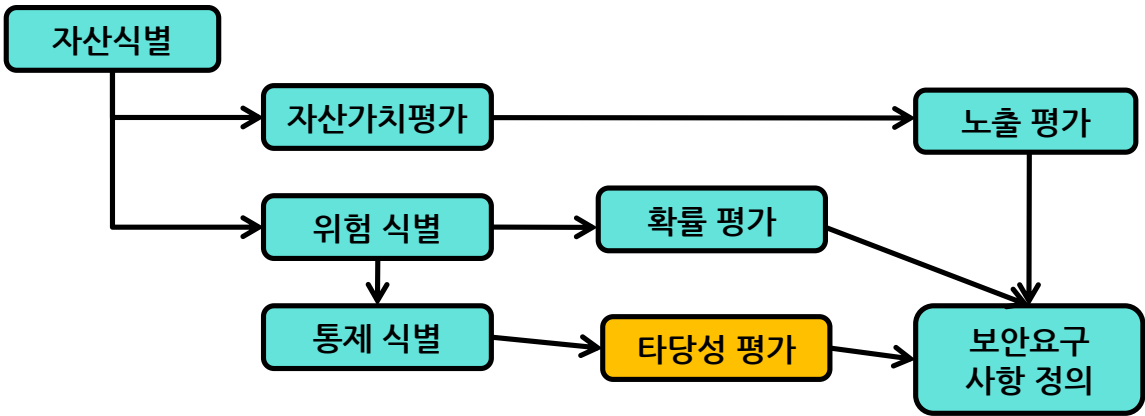
[3] 정보보호 고려사항

2. 정보보호 관리(계속)

◆ 사전의 위험을 평가 하는 방법(계속)



- 자산을 보호하기 위해 적절하게 시행되는 **통제**를 **제안**하는 단계



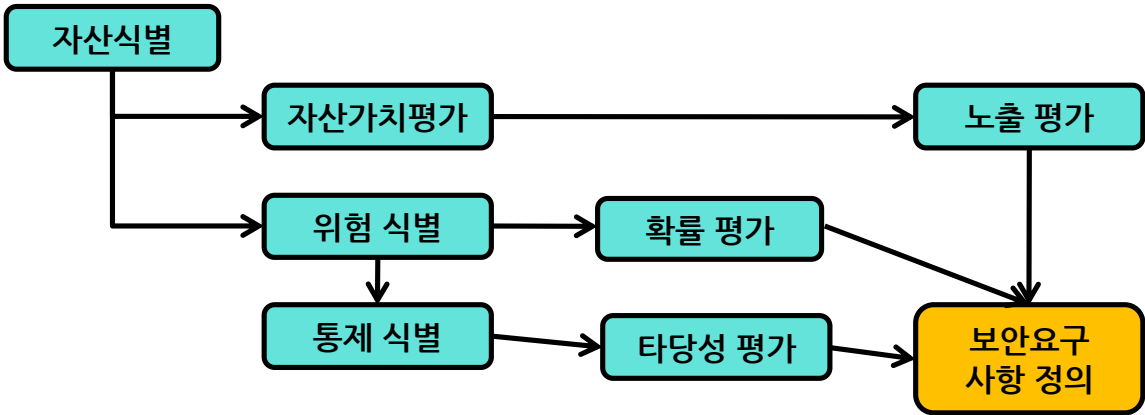
- 기술적 타당성과 제안된 통제비용을 **평가**하는 단계

❖ 학습내용

[3] 정보보호 고려사항

2. 정보보호 관리(계속)

◆ 사전의 위험을 평가 하는 방법(계속)



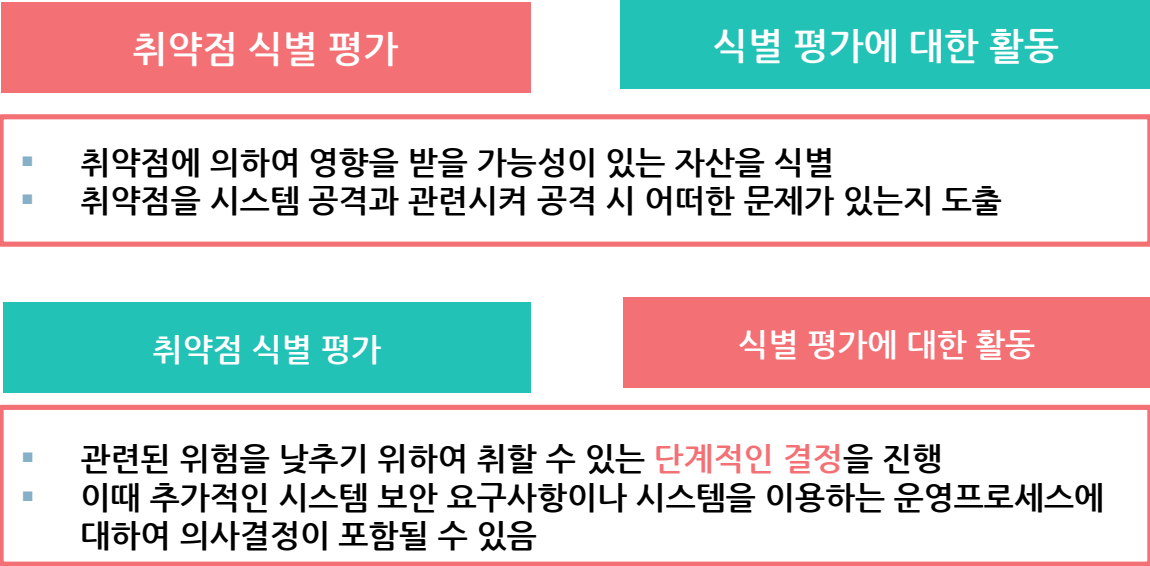
- 노출, 위협, 통제 평가가 시스템 보안 요구사항 집합을 유도하기 위해 이용되는 단계
- 이 요구사항은 시스템 기반구조나 응용 시스템에 관한 요구사항임

❖ 학습내용

[3] 정보보호 고려사항

2. 정보보호 관리(계속)

- ◆ 시스템 생명주기 위험 평가
 - 시스템을 도입하기 이전의 사전의 위험을 평가하는 방법
 - 대비하여 시스템을 도입하고 운영 및 폐기하는데 있어서 일련의 과정 중 위험에 대하여 평가하는 방법



3. 정보보호를 위한 설계

- ◆ 설계 시 고려 사항

보안결정은
명시적인 보안 정책에 기초하라.

- 보안정책은 **보안결정을 명료**하게 하는 것부터 시작됨
- 보안을 위한 행동과 결정이 명시적으로 나타나 있어야 함

❖ 학습내용

[3] 정보보호 고려사항

3. 정보보호를 위한 설계(계속)

◆ 설계 시 고려 사항(계속)

하나의 장애지점을 피하라.

- 하나의 장애지점이 **전체적인 시스템 고장**으로 귀결되지 않아야 함을 의미

보안을 유지하며 실패하라.

- 보안을 유지하는 시스템은 실패하더라도 **보안은 유지**하여야 함(Fail-secure)
- [예] 시스템이 고장이나 장애, 전원이 차단될 때에 아무나 중요 데이터를 열람하는 경우

보안과 유용성 사이에 균형을 맞추어라.

- 보안과 유용성은 서로 반대되는 경우가 있음
- [예]시스템 접속자가 편리하기 위하여 패스워드를 1자리로 만드는 경우, 수십 번의 패스워드를 요구하는 경우
- 적정 수준에서의 **균형**이 필요

사회 공학의 가능성을 인식하라.

- [예] 만일 회사의 상급자나 친한 사람관계에서 시스템 접근 가능자에게 권한을 요청하는 경우
- 모든 사람이 가진 생체인식기법이나 디지털 서명 등의 방법으로 보다 **강력한 보안제도** 유지

위험을 줄이기 위해 중복성과 다양성을 이용하라.

- [예] 중요한 서류문서 파일의 이름 변경하여 중복 저장
- [예] 한글, 워드의 두 버전으로 다양하게 문서를 작성
- 위험을 대비하여 **안전한 내용보관**을 위한 일

❖ 학습내용

[3] 정보보호 고려사항

3. 정보보호를 위한 설계(계속)

◆ 설계 시 고려 사항(계속)

모든 입력을 검증하라.

- 보안 침해 사고 중 입력 값의 조작으로 수행되는 경우가 많음
- Sql-injection 방법의 경우 일반 입력필드의 sql문장을 넣으면 엉뚱한 행위가 나타나는 방식
- 이러한 사고들은 **입력 값 검증**으로도 충분히 예방 가능

자산을 구분하라.

- [예] 어떤 회사의 시스템을 인사, 회계, 사업판단 등의 시스템으로 **논리적, 물리적으로 분리**하는 경우
- 한 곳의 사고가 있더라도 다른 시스템으로는 영향 받지 않을 것임

배치를 위하여 설계하라.

- 배치: 시스템을 **올바르게 설정**하는 것
- [예] 전체시스템에서 보안설정을 관리자가 쉽게 설정하는 경우

복구 가능성을 위해 설계하라.

- 시스템을 아무리 보안상황을 잘 대비하도록 구축하였다 하더라도 문제가 발생할 수 있음
- 항상 보안침해 상황이 발생하더라도 **보다 빠르게 복구**할 수 있도록 **복구 가능성**을 위해 설계하여야 함

❖ 핵심정리

1. 서비스 지향(SOA) 개발 방법론

- 필요한 소프트웨어나 정보시스템을 매번 중복으로 구현하는 것보다 해당 서비스를 어떤 **체계에 의하여** 제공 받아서 사용하는 개념
- 표준 기술로 **SOAP, UDDI, WSDL**의 웹 서비스의 표준 요소기술을 사용

2. 관점지향(AOP) 개발 방법론

- 소프트웨어를 개발 하는 경우 어떤 **중심적인 내용을 추상화**하여 이를 중심으로 프로그램을 개발하는 기법
- 프로그램의 변경, 재 사용시 절단관심사는 프로그래밍을 더욱 복잡하게 해주는 요소임

3. 정보보호 고려사항

- 정보보호 활동의 3대 위협은 **기밀성, 무결성, 가용성의 위협**임
- 시스템을 설계하는 데 있어서 **정보보호**를 위한 몇 가지 사항을 고려하여야 함