

# 소프트웨어 공학

## Tutorial #2: StarUML

Eun Man Choi  
emchoi@dgu.ac.kr

# Contents

- **StarUML 개요**
  - StarUML 소개 및 특징
  - 주요 기능
  - StarUML 화면 소개
- **StarUML 설치**
  - StarUML 다운 & 설치하기
- **연습**
  - 사용 사례 다이어그램 그리기
  - 클래스 다이어그램 그리기
  - 순서 다이어그램 그리기

# StarUML 개요

## ● StarUML 소개 및 특징

### ● 소개

- Rational Rose, Together와 같은 상업적 도구에 준하는 기능을 갖춘 오픈 소스 소프트웨어 모델링 도구 및 플랫폼

### ● 특징

- UML 표준 명세에 기반한 모델 작성과 UML2.0 표기법을 지원
- 완전한 UML Profile 개념을 제공하여 플랫폼에 독립적인 모델을 작성할 수 있도록 지원
- 뛰어난 확장성과 유연성을 제공
- 모델을 소스 코드로 생성하고, 소스 코드로부터 모델을 Reverse Engineering 할 수 있음
- 심플한 인터페이스로 학습 및 사용 용이 ⇒ 개발 효율 증대

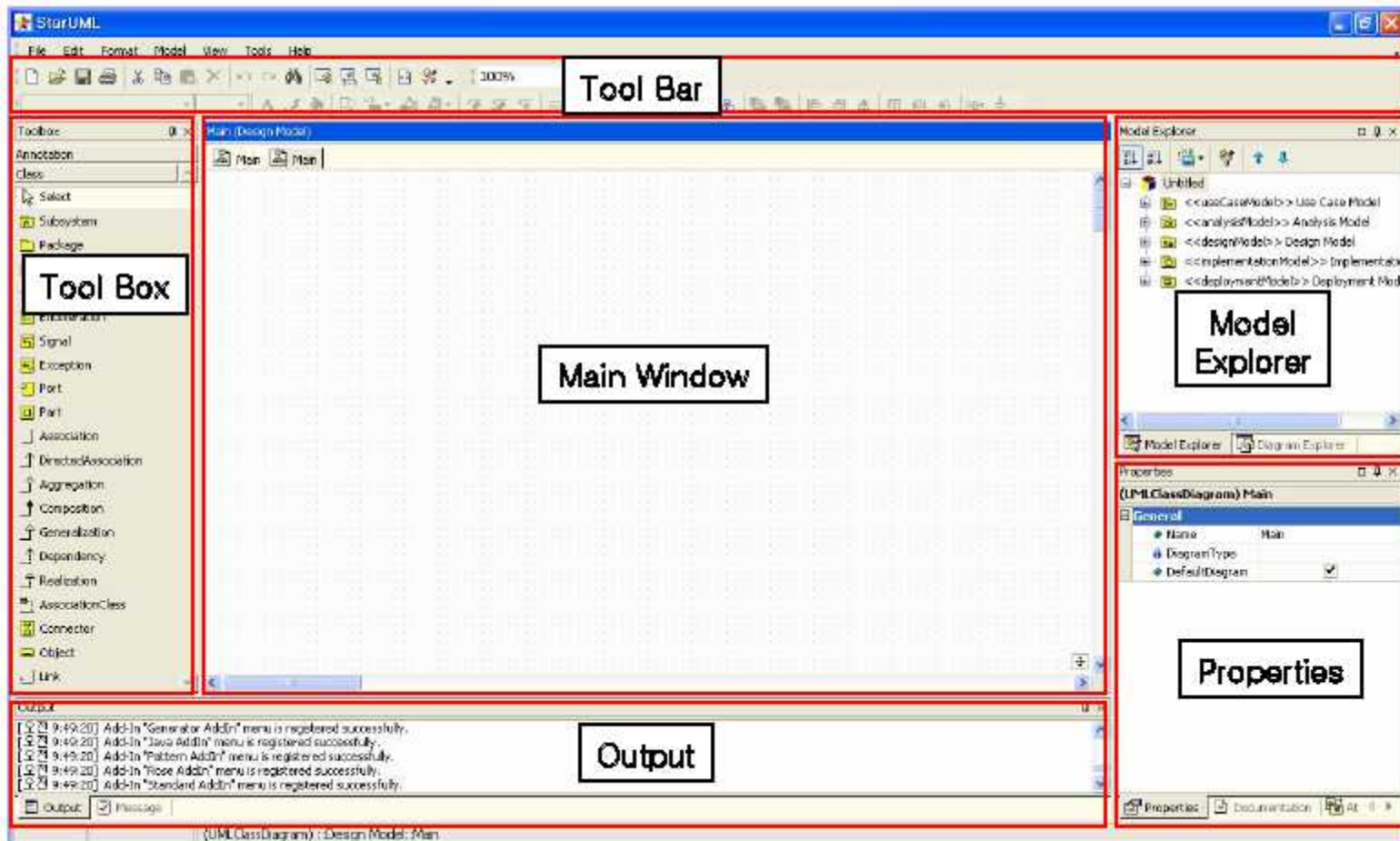
# StarUML 개요

## ● StarUML 주요기능

- StarUML를 이용하면 UML Diagram을 빠르고 쉽게 그릴 수 있음
- 모델을 소스 코드로 생성하고, 소스 코드로부터 모델을 Reverse Engineering 가능
- .NET, J2EE와 같은 플랫폼 환경을 적용
- StarUML에서 그릴 수 있는 Diagram 종류
  - Class Diagram
  - Use Case Diagram
  - Sequence Diagrams
  - Collaboration Diagrams
  - Statechart Diagram
  - Activity Diagram
  - Component Diagram
  - Deployment Diagram
  - Composite Structure Diagram

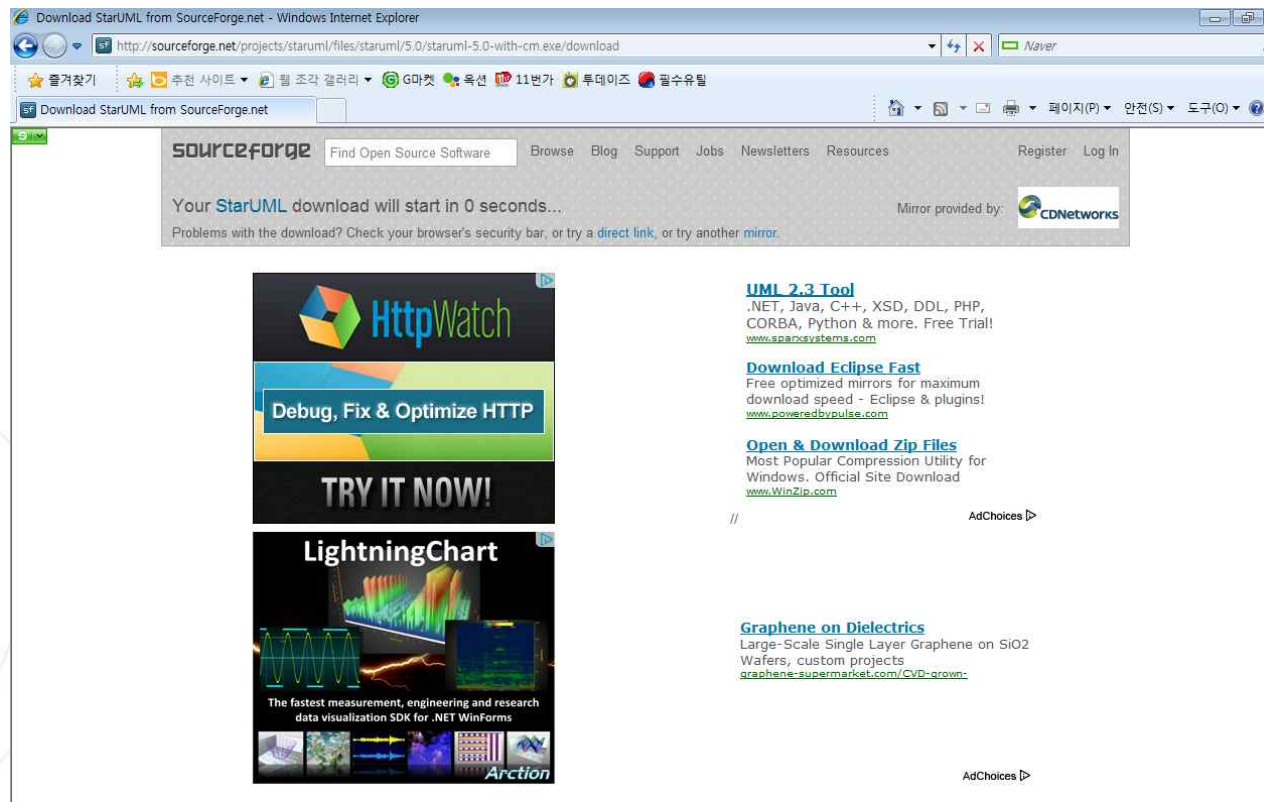
# StarUML 개요

- StarUML 화면 소개



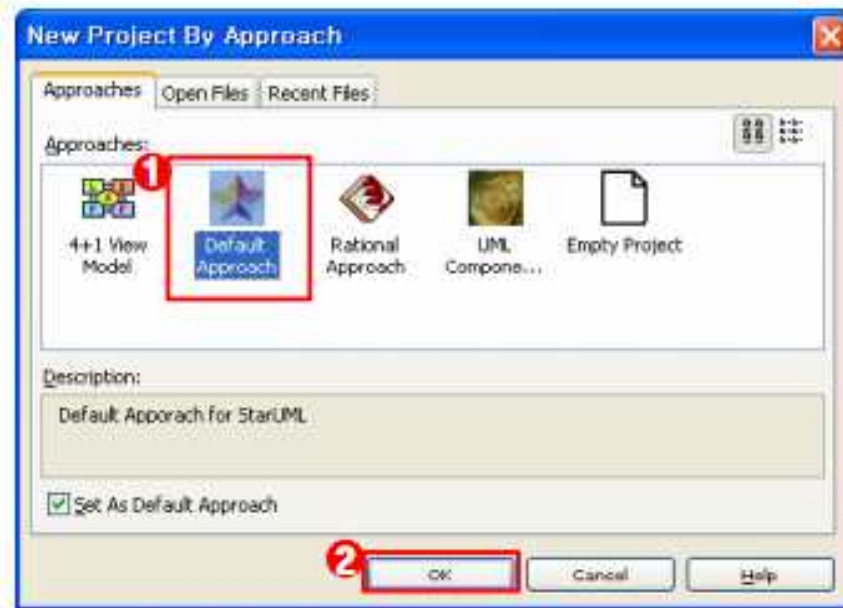
# StarUML 설치

- 아래의 사이트에서 StarUML을 다운 받아 설치
  - <http://sourceforge.net/projects/staruml/files/staruml/5.0/staruml-5.0-with-cm.exe/download>



# StarUML : 다이어그램 생성 방법

- Step Step 1 (계속)
  - StarUML을 실행하고 새 Project를 생성
  - Approach는 'Default Approach'를 선택

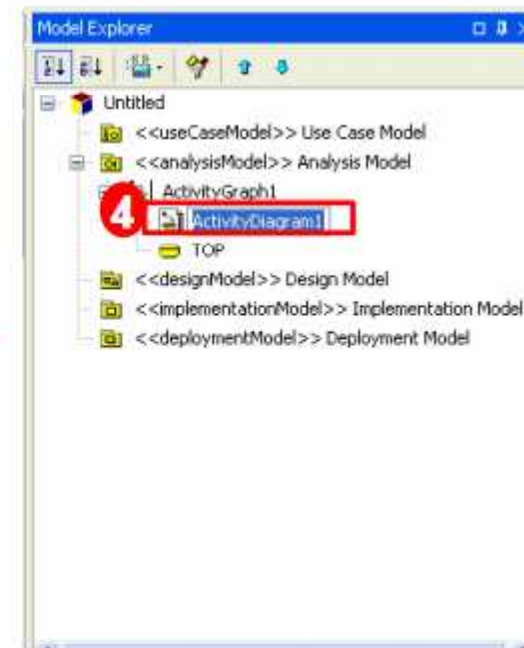
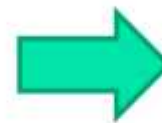
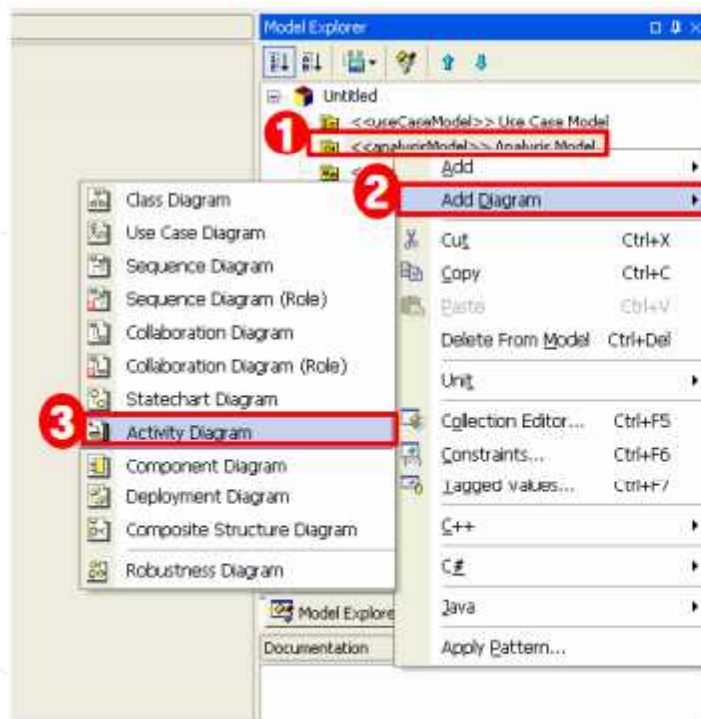


# StarUML : 다이어그램 생성 방법

- Step Step 2 (계속)

- 각 다이어그램 생성방법

- Model Explorer의 각 다이어그램에 적합한 Model 선정 -> 오른쪽버튼클릭 -> Add Diagram -> 원하는 다이어그램 선택 -> 이름변경

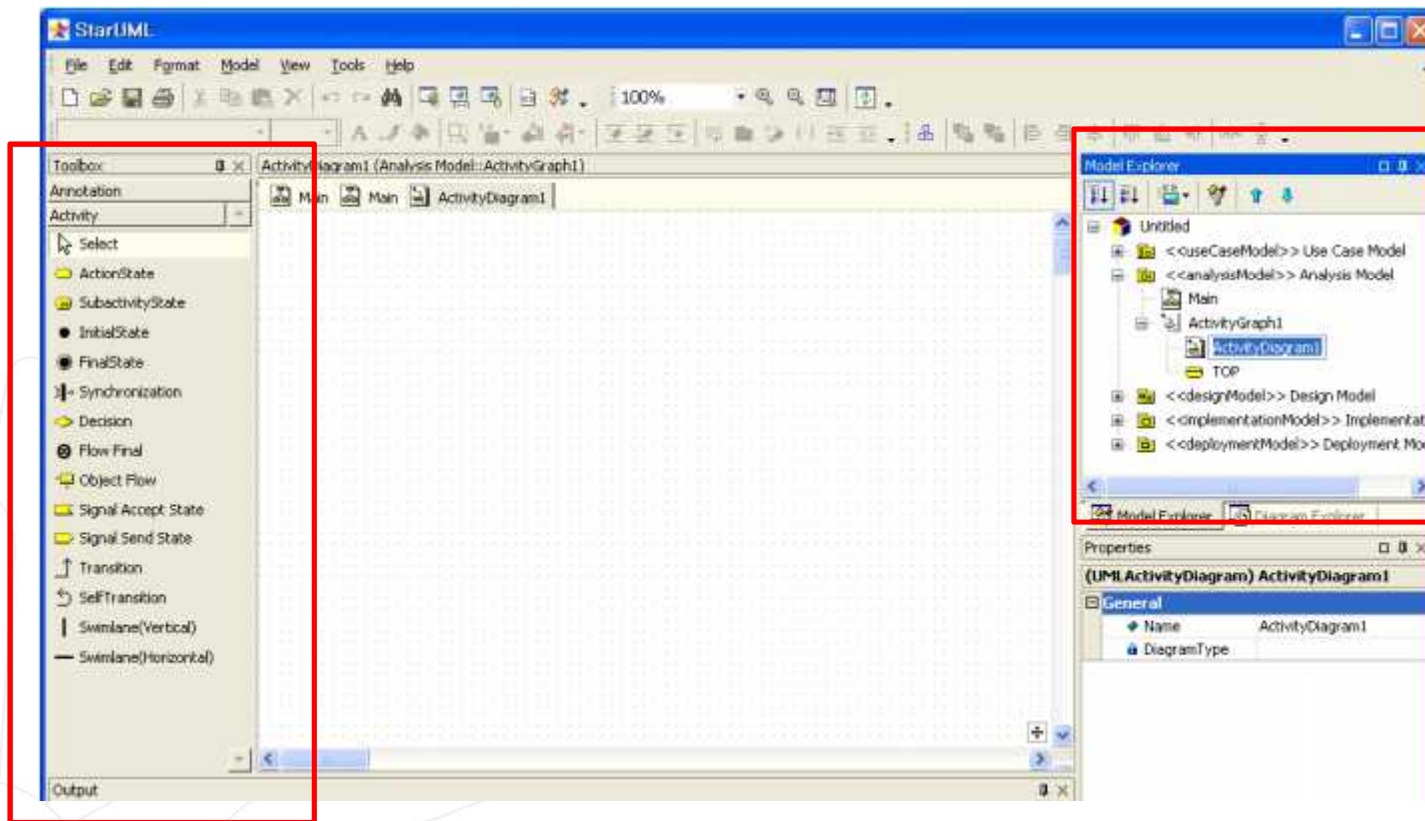




# StarUML : 다이어그램 생성 방법

## ● Step Step 3

- 선택한 다이어그램이 Model Explorer에 생성되고, Toolbox와 Main Window가 다이어그램의 종류에 맞게 생성



# StarUML연습 - Use Case Diagram

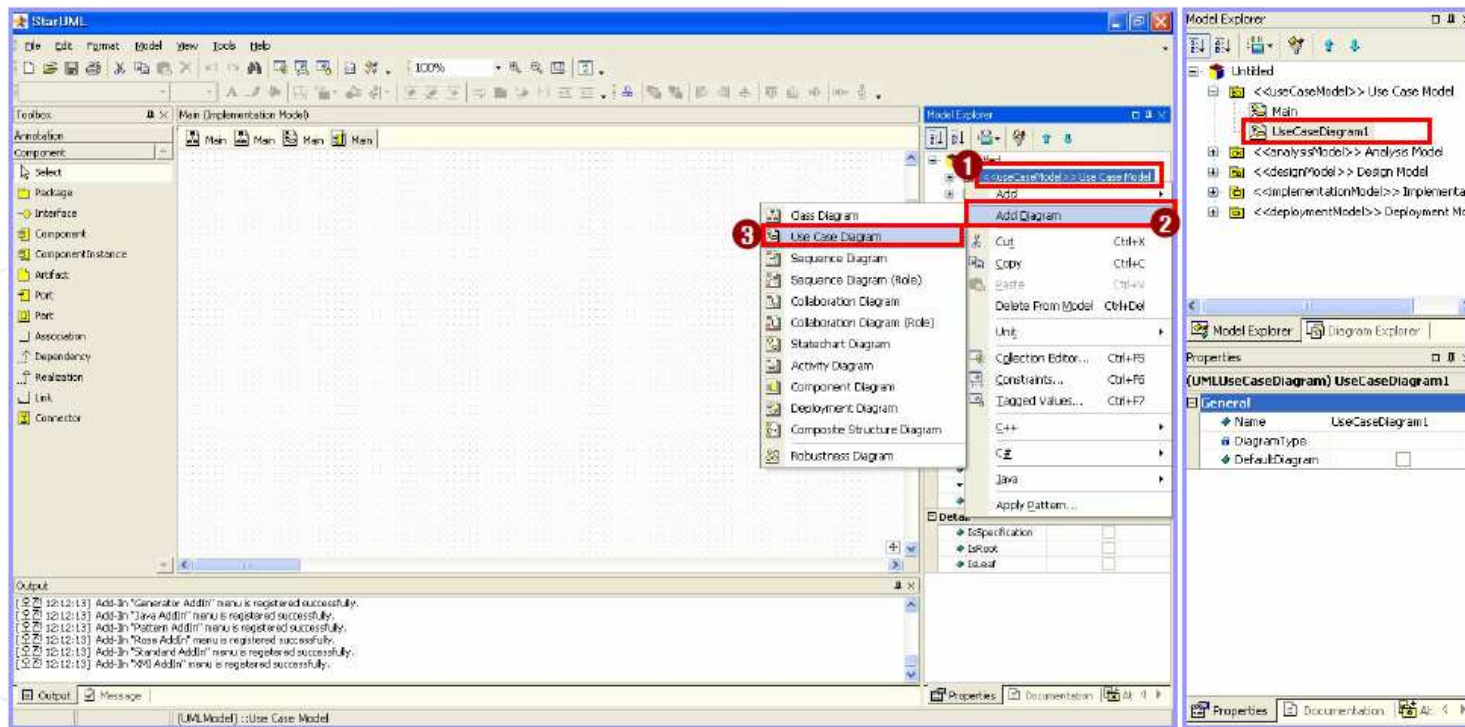
- **사용사례**

- StarUML의 기능을 소개 위한 간단한 수강신청 시스템
  - 수강신청 요구사항

이 시스템은 수강 신청 페이지에 접속하여 수강신청을 할 수 있도록 도와주는 것이다.  
시스템에 등록된 학교 학생이 수강신청 페이지에 로그인 후 과목을 선택하고, 수업을 선택하여 잔여좌석을 확인한다.  
만약, 잔여좌석이 없을 경우, 과목선택부터 다시 하고 잔여좌석이 있을 경우 수강신청을 완료하게 된다.

# StarUML 연습 – Use Case Diagram

- Use Case Diagram 그리기
  - Use Case Diagram을 그리기 위해서 Diagram을 생성
    - Model Explorer의 <<usecaseModel>> Use Case Model 선택 -> 마우스 오른쪽 버튼 클릭 -> Add Diagram -> Use Case Diagram 선택



# StarUML 연습 – Use Case Diagram

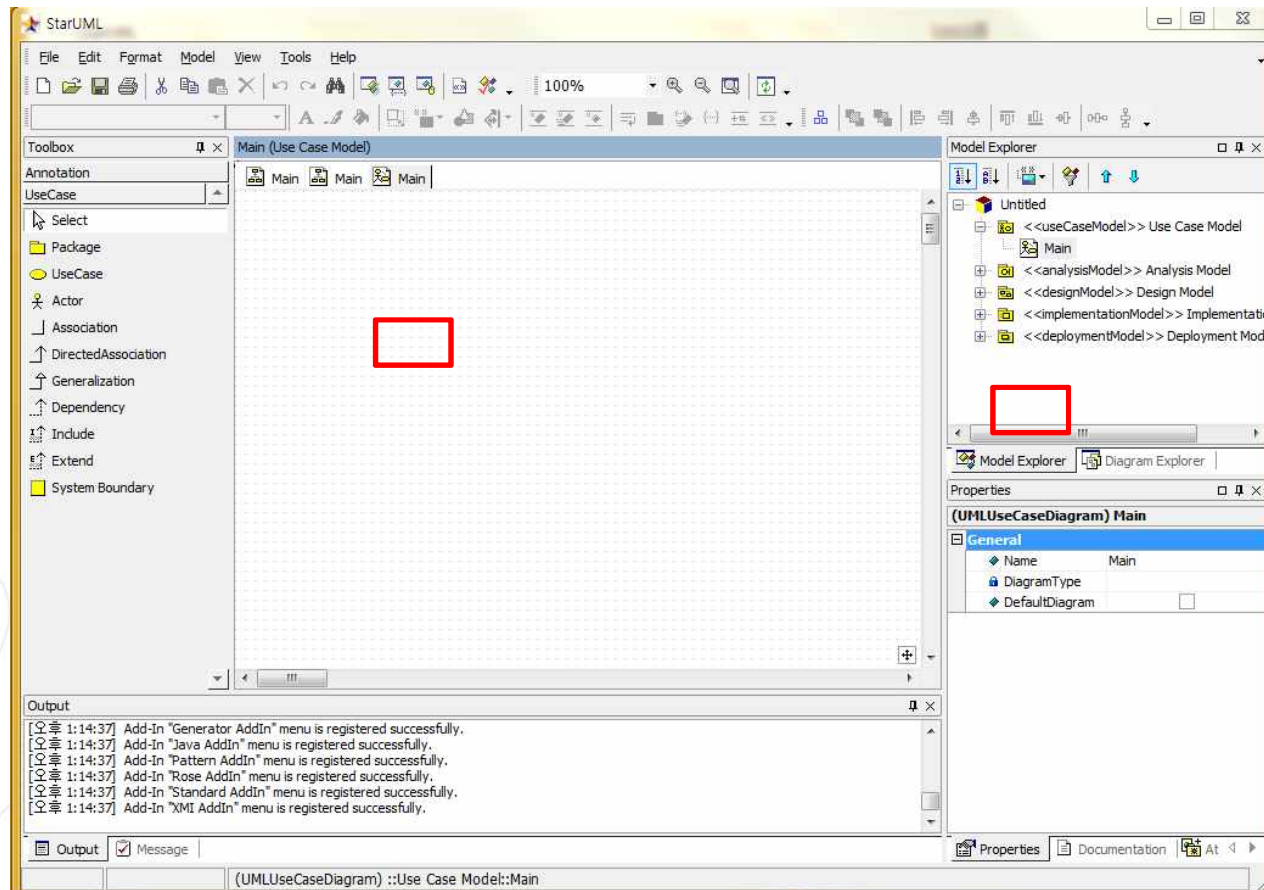
- Use Case Diagram Tool Bar



Select	Diagram 요소를 선택합니다.
Package	모델 요소들을 논리적으로 그룹화 할 때 사용합니다.
UseCase	시스템이 제공하는 기능을 의미합니다.
Actor	시스템을 사용하는 사용자나 외부 시스템을 의미합니다.
Association	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다.
DirectedAssociation	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다. 종속, 포함등의 관계를 표시할 수 있습니다.
Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용합니다.
Dependency	어떤 유스케이스를 위해 다른 요소의 존재가 요구 되어지는 의존적인 관계를 의미합니다
Include	어떤 유스케이스를 수행하는 데, 반드시 수행해야 하는 유스케이스를 나타내기 위해 사용합니다.
Extend	한 유스케이스가 특정 시점에 여러 가지 형태로 분류될 경우에 사용합니다.
System Boundary	시스템과 외부 시스템의 경계를 의미합니다.

# StarUML 연습 – Use Case Diagram

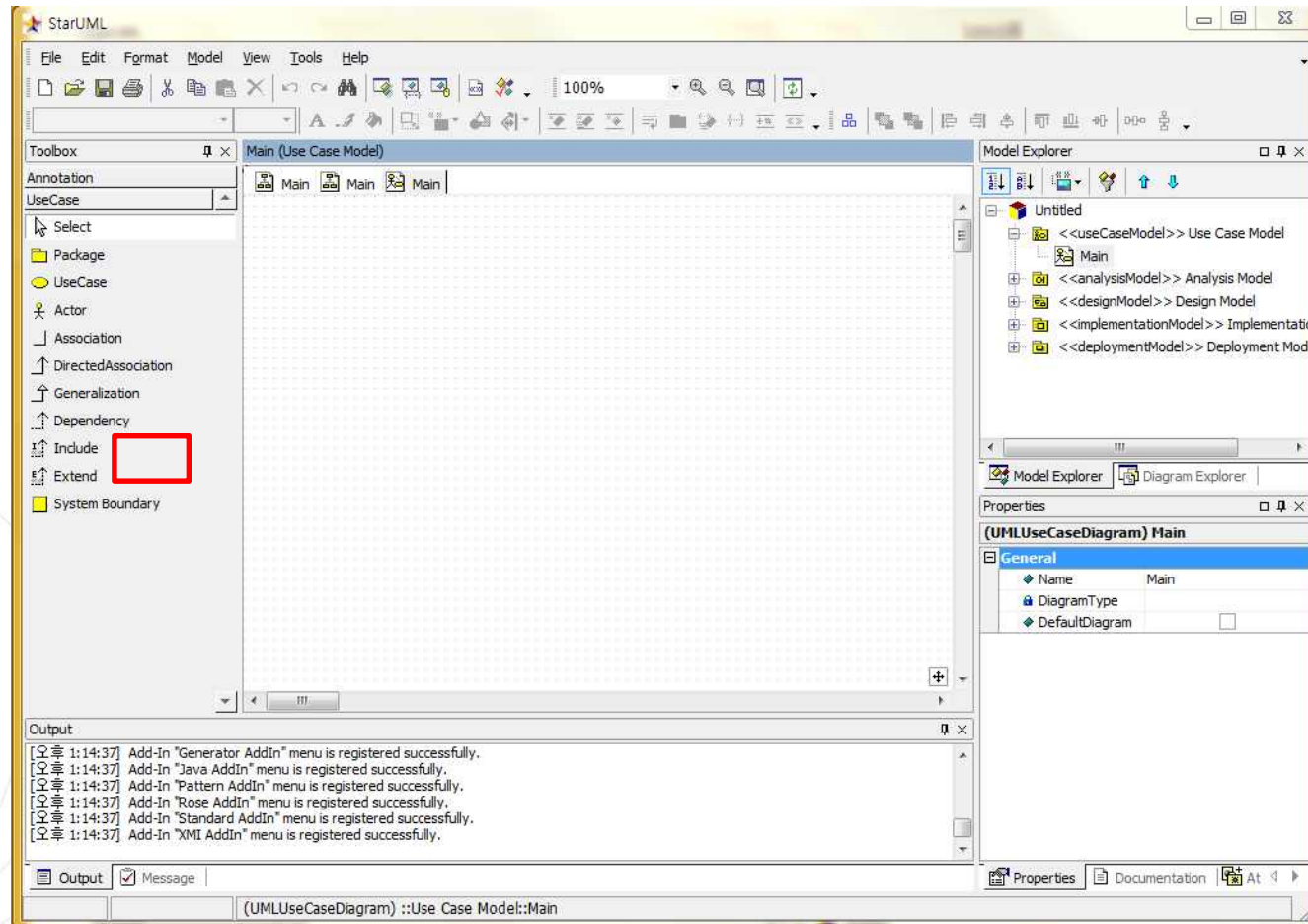
- Use Case Diagram – Main 선택





# StarUML 연습 – Use Case Diagram

- Use Case Diagram – ToolBox -> Actor 선택



# StarUML 연습 – Use Case Diagram

- Use Case Diagram – Actor 박스 이름 기입

원하는 이름으로 기입

StarUML

File Edit Format Model View Tools Help

100%

Toolbox

Annotation

UseCase

Select

Package

UseCase

Actor

Association

DirectedAssociation

Generalization

Dependency

Include

Extend

System Boundary

Main (Use Case Model)

Main Main Main

Model Explorer

Diagram Explorer

Properties

(UMLActor) Bae, Rankyoung

General

- Name Bae, Rankyoung
- Stereotype
- Visibility PUBLIC
- IsAbstract
- Attributes (Collection)[0]
- Operations (Collection)[0]

Detail

- IsSpecification
- IsRoot
- IsLeaf

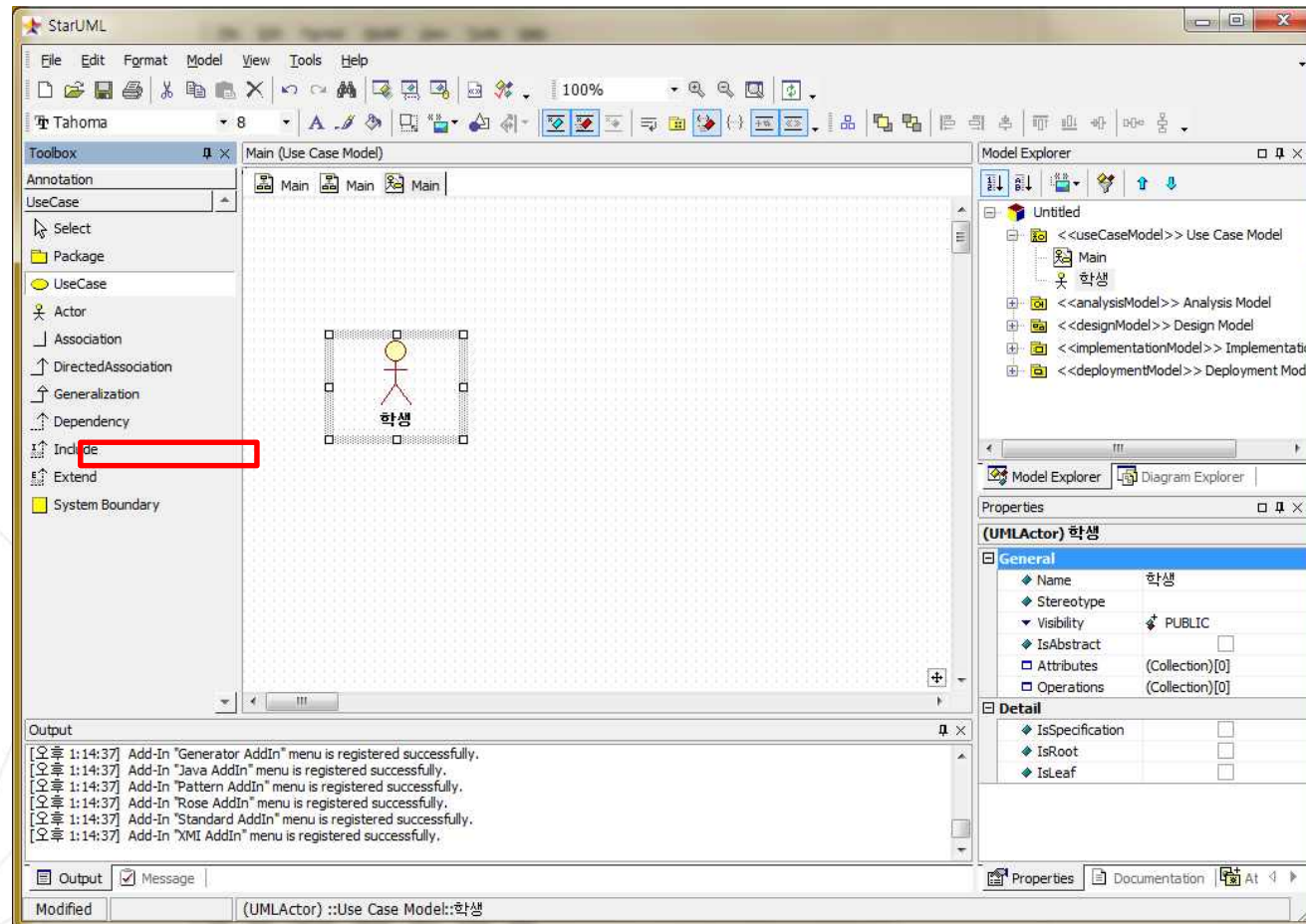
Output

[오류 1:14:37] Add-In "Generator AddIn" menu is registered successfully.  
[오류 1:14:37] Add-In "Java AddIn" menu is registered successfully.  
[오류 1:14:37] Add-In "Pattern AddIn" menu is registered successfully.  
[오류 1:14:37] Add-In "Rose AddIn" menu is registered successfully.  
[오류 1:14:37] Add-In "Standard AddIn" menu is registered successfully.  
[오류 1:14:37] Add-In "XMI AddIn" menu is registered successfully.

Modified (UMLActor) ::Use Case Model::Bae, Rankyoung

# StarUML 연습 – Use Case Diagram

- Use Case Diagram – UseCase 선택





# StarUML 연습 – Use Case Diagram

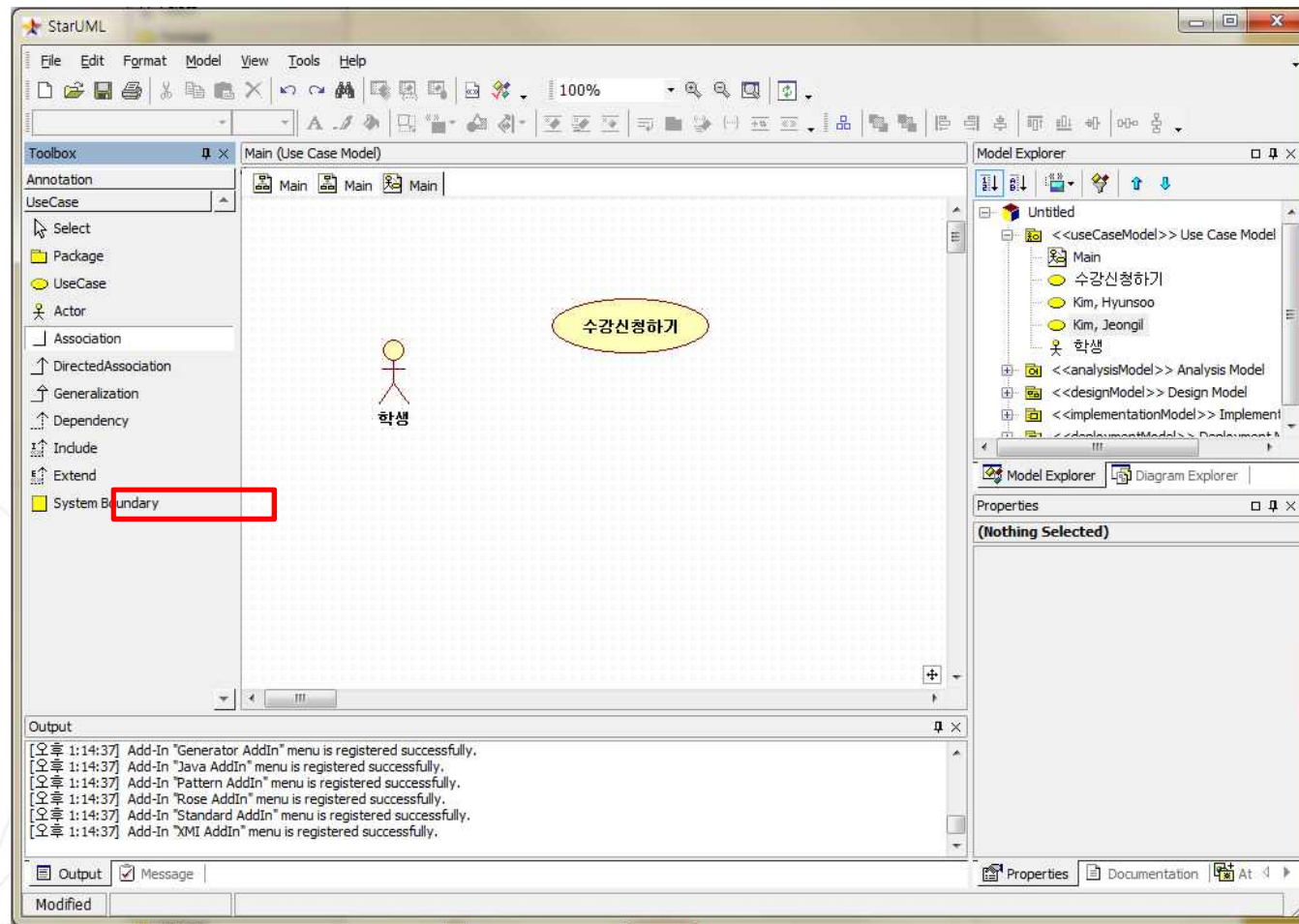
- Use Case Diagram – UseCase 이름 기입

The screenshot shows the StarUML interface. On the left is the Toolbox with various UML elements. The main workspace displays a Use Case Diagram with an actor labeled '학생' (Student) and a use case. A red box highlights the use case name field, and a callout box with the text '사용할 이름 기입' (Enter the name to be used) points to it. On the right, the Model Explorer shows the project structure, and the Properties window displays details for the selected use case.

(UMLUseCase) Jung, Yoontae	
<b>General</b>	
Name	Jung, Yoontae
Stereotype	
Visibility	PUBLIC
IsAbstract	<input type="checkbox"/>
Attributes	(Collection)[0]
Operations	(Collection)[0]
ExtensionPoints	(Collection)[0]
<b>Detail</b>	
IsSpecification	<input type="checkbox"/>
IsRoot	<input type="checkbox"/>
IsLeaf	<input type="checkbox"/>

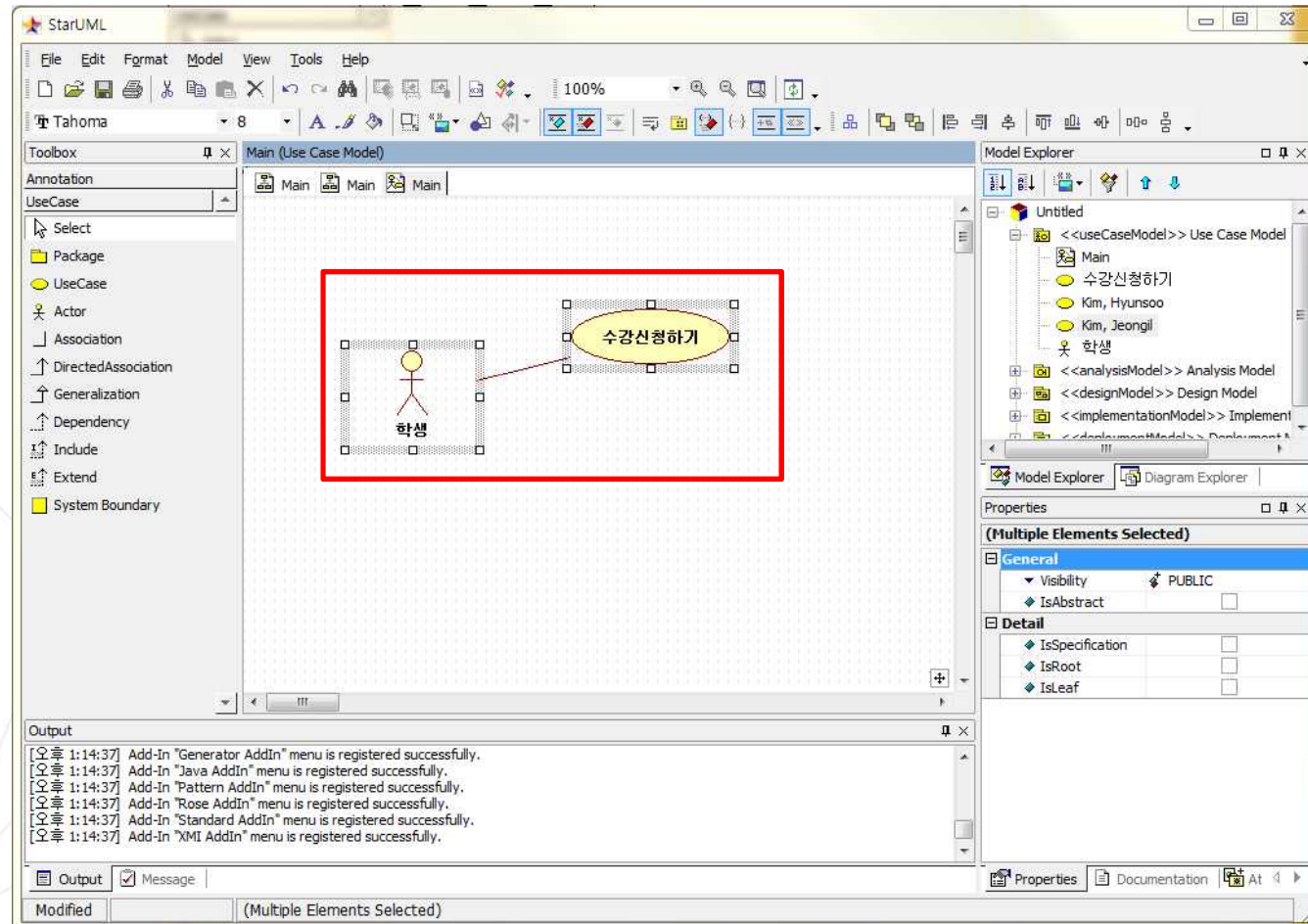
# StarUML 연습 – Use Case Diagram

- Use Case Diagram – Association 선택



# StarUML 연습 – Use Case Diagram

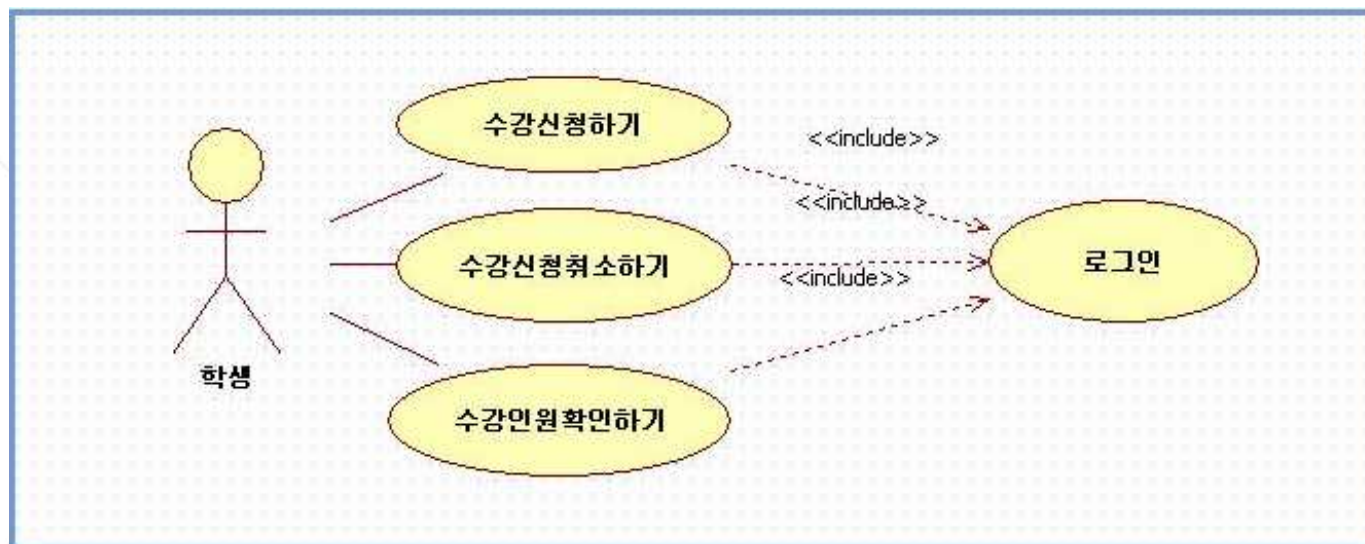
- Use Case Diagram – 두 요소를 선택한 후(Shift) 가운데 드래그



# StarUML 연습 – Use Case Diagram

- Use Case Diagram

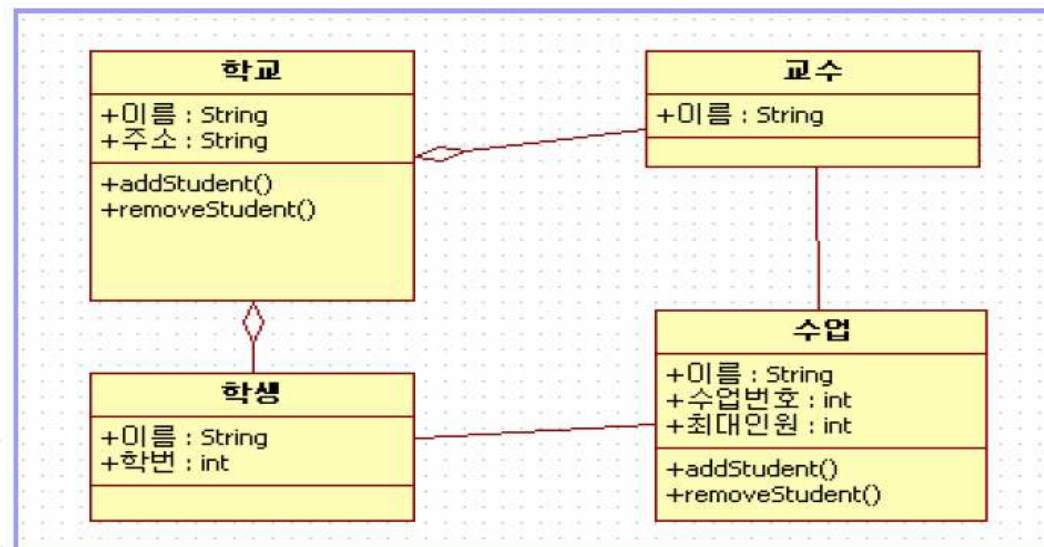
대기 상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다.  
로그인을 하고 과목 목록을 확인한 뒤 과목을 선택한다.  
선택한 과목의 수업을 선택하면 수강신청이 완료된다.



# StarUML 연습 – Class Diagram

- 10page의 수강신청 시스템의 요구사항을 Class Diagram으로..

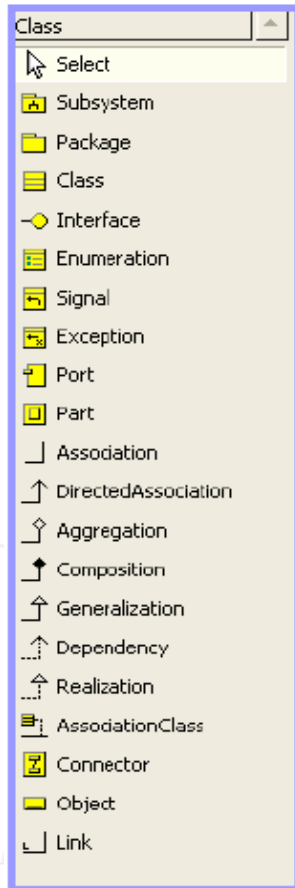
학교는 0명 이상의 학생으로 구성된 집합연관이고 학생은 여러 수업을, 수업은 여러 명의 학생을 수용할 수 있는 다대다 관계이다. 교수 한 명당 0개 이상의 수업을 담당한다.





# StarUML 연습 – Class Diagram

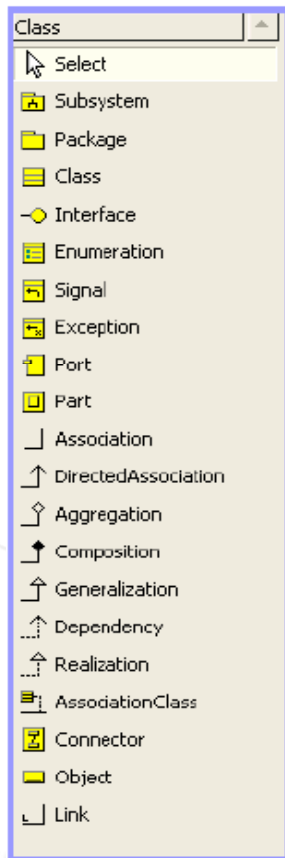
## ▪ Class Diagram Tool Bar(1/2)



Select	Diagram 요소를 선택합니다.
Subsystem	물리적인 시스템의 부분 또는 전체를 의미합니다.
Package	관련된 클래스들을 모아서 모델링 한 것을 의미합니다.
Class	객체의 Attribute와 Method을 모델링 한 것을 의미합니다.
Interface	클래스에서 Method의 선언부분만 모델링 한 것을 의미합니다.
Enumeration	미리 정의된 값들을 리스트로 가지는 데이터타입을 의미합니다.
Signal	객체간의 비동기적 통신 신호를 의미합니다.
Exception	실행 오류시 Operation에 의해 발생하는 신호를 의미합니다.
Port	인터페이스와 클래스의 연결 단자를 의미합니다.
Part	클래스 내부의 특정한 부분을 의미합니다.

# StarUML 연습 – Class Diagram

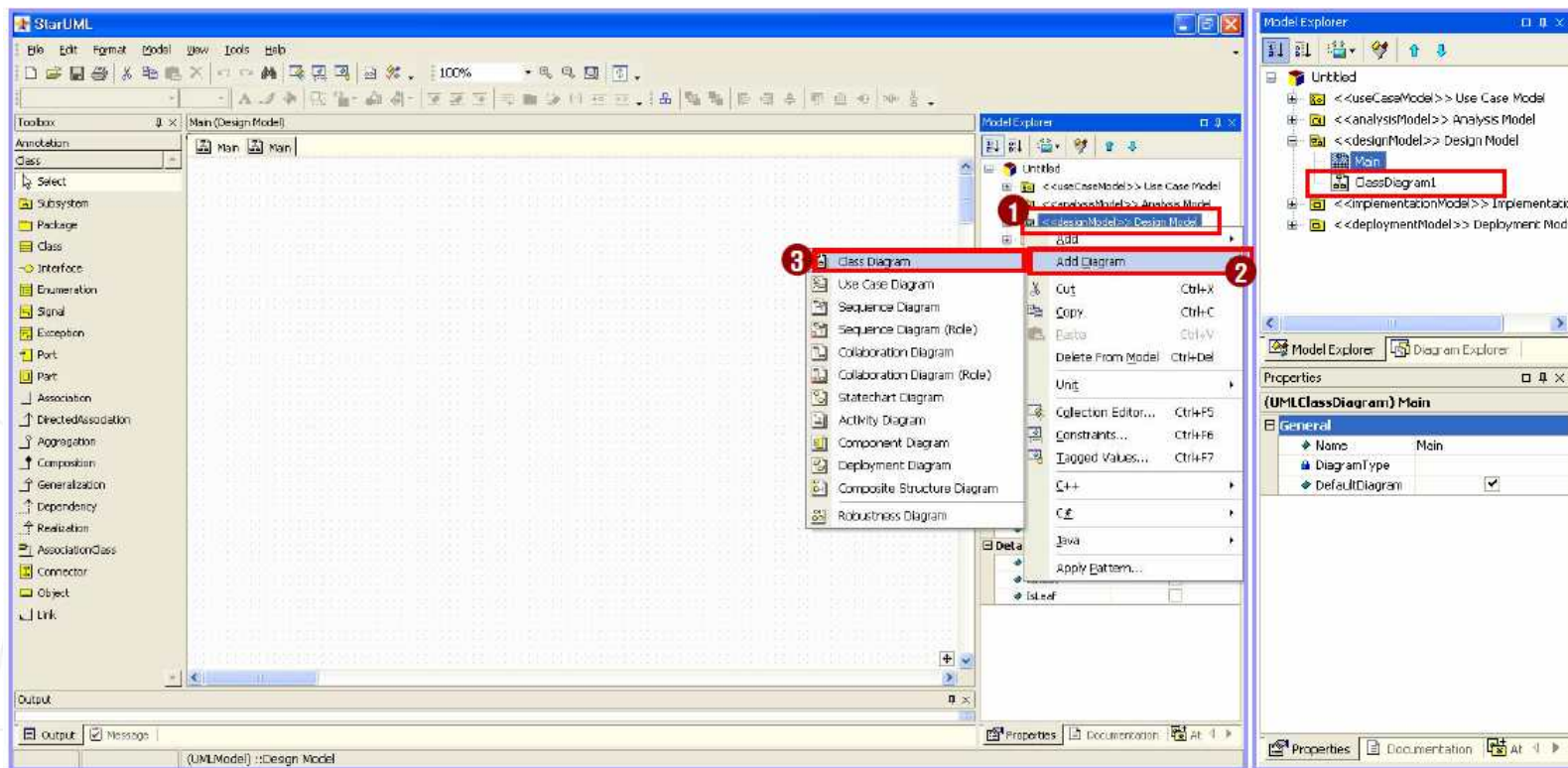
## ▪ Class Diagram Tool Bar (2/2)



Association	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 가능합니다.
DirectedAssociation	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 불가능합니다.
Aggregation	한 클래스가 다른 클래스를 포함하는 관계일 때 사용합니다.
Composition	한 클래스가 다른 클래스에 완전히 종속되는 관계일 때 사용합니다.
Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용합니다.
Dependency	한 클래스의 변화가 다른 클래스의 변화에 영향을 주는 관계를 의미합니다.
Realization	인터페이스와 클래스와의 연결에 사용합니다.
AssociationClass	클래스와 연과의 연결에 사용합니다.
Connector	Part 사이의 연결에 사용합니다.
Object	특정 클래스의 인스턴스를 의미합니다.
Link	객체 사이의 연결에 사용합니다.

# StarUML 연습 – Class Diagram

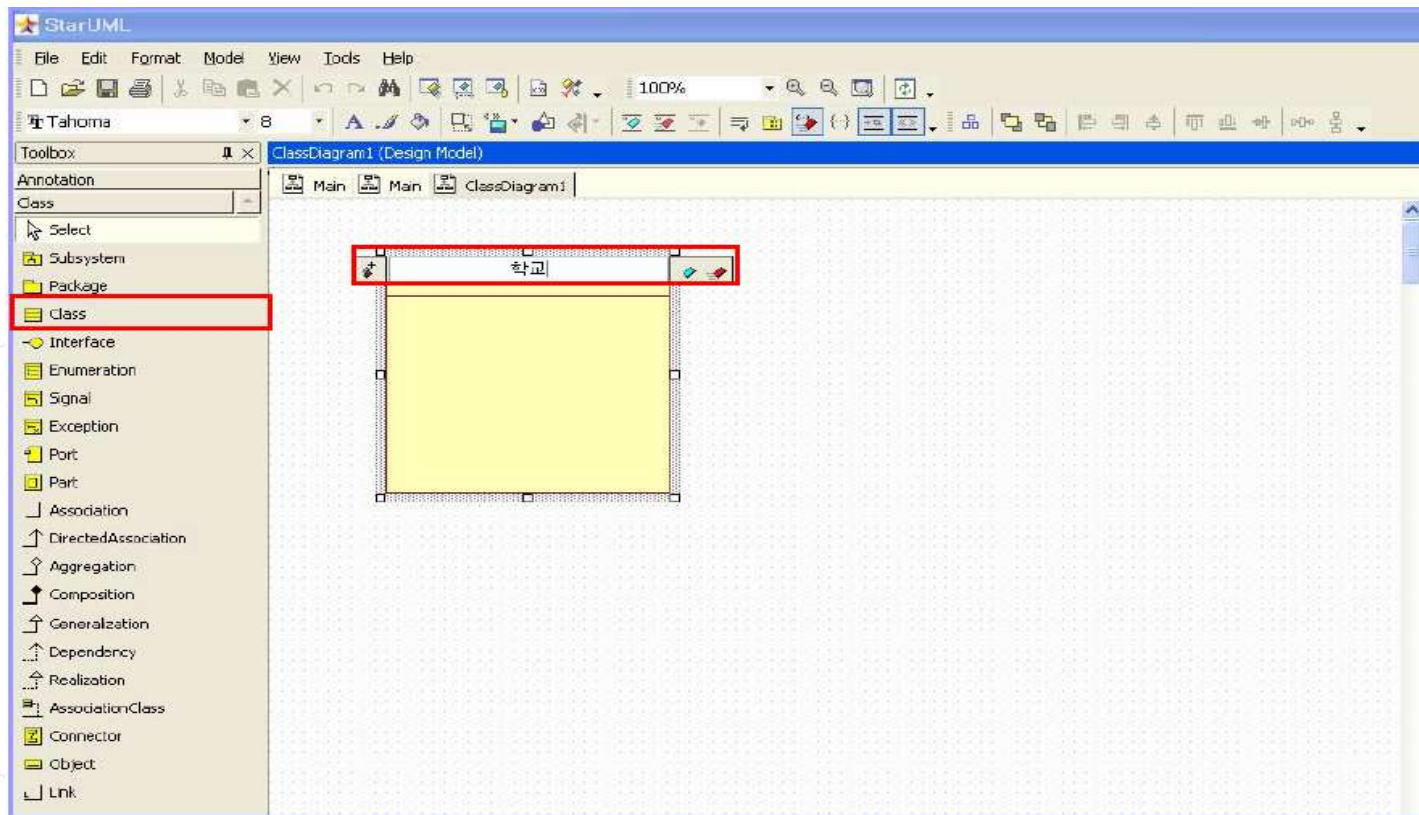
- Use Case Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 <<designModel>> Design Model 선택 -> 마우스 오른쪽 버튼 클릭 -> Add Diagram -> Class Diagram 선택





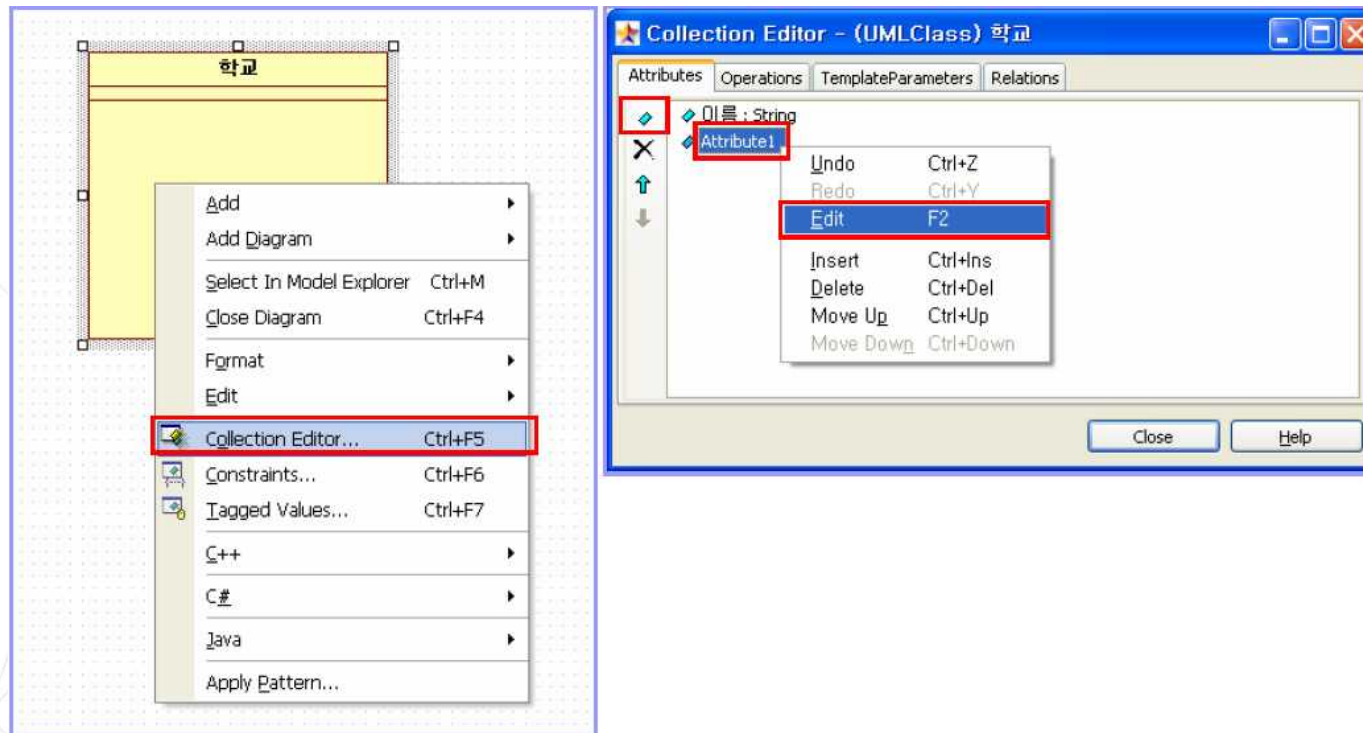
# StarUML 연습 – Class Diagram

- 시스템을 구성하는 학교 클래스를 표현합니다.
  - Toolbox의 Class 에서 Class 아이콘 클릭 -> 마우스로 Class 크기 지정 -> Class 이름 입력 -> Enter



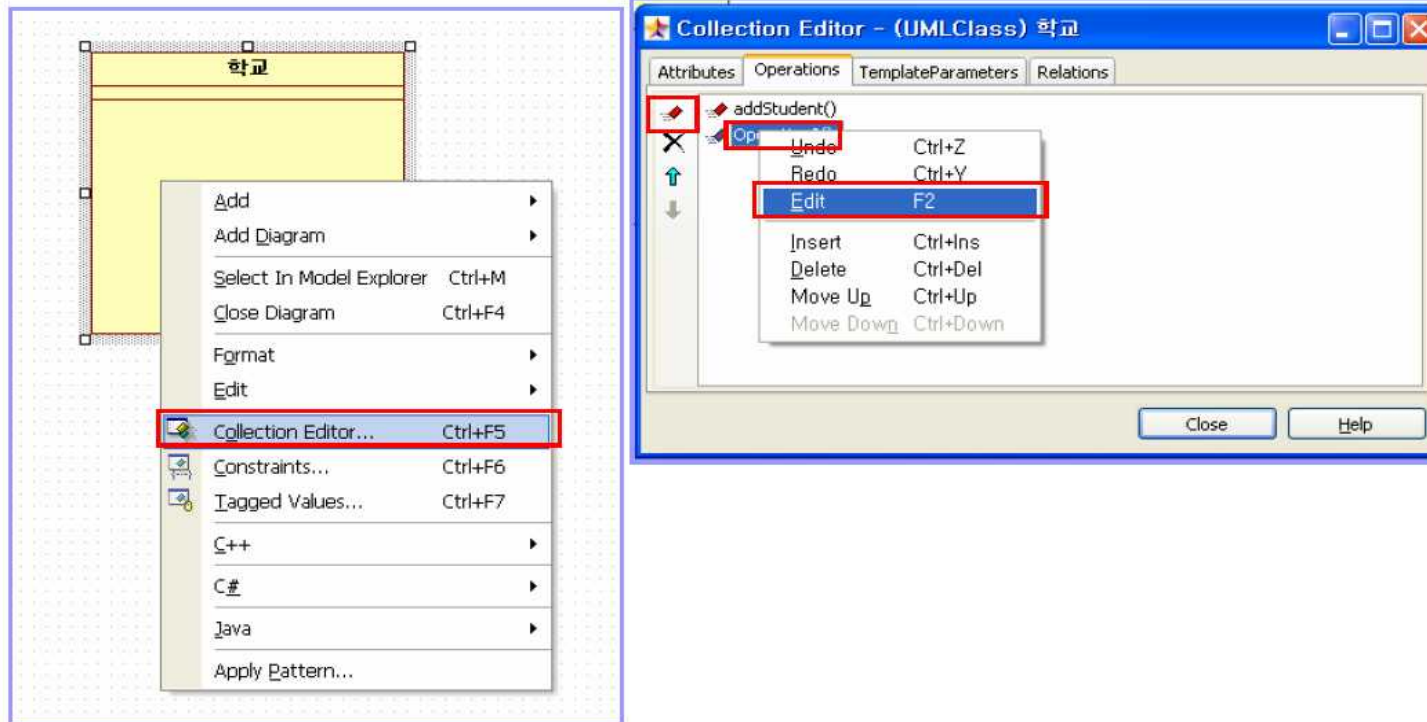
# StarUML 연습 – Class Diagram

- 학교 Class에 Attributes를 입력합니다.
  - 학교 Class 선택 -> 마우스 오른쪽 버튼 클릭 -> Collection Editor... 클릭
  - Attributes 탭 -> Insert 아이콘 클릭 -> Edit 클릭-> 마우스 오른쪽 버튼 클릭 -> Name 변경



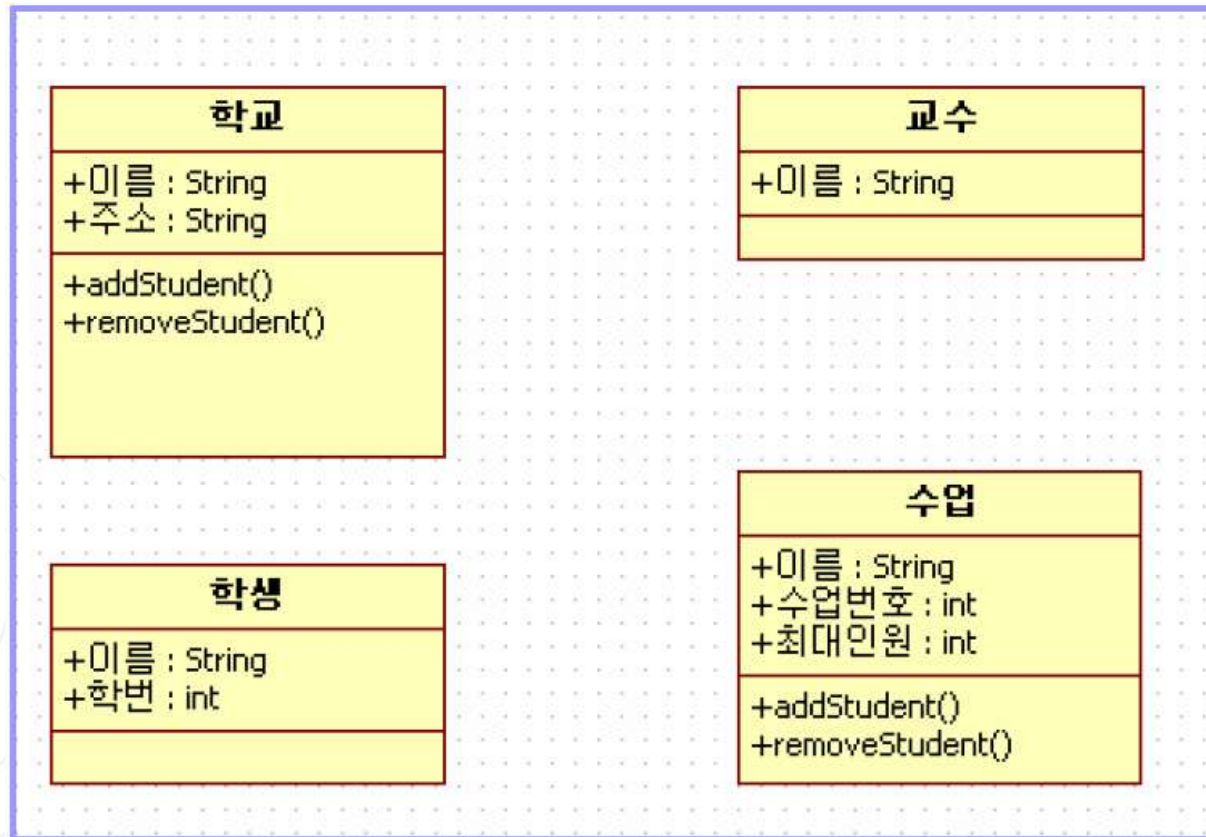
# StarUML 실습 – Class Diagram

- 학교 Class에 Operation를 입력합니다.
  - 학교 Class 선택 -> 마우스 오른쪽 버튼 클릭 -> Collection Editor... 클릭
  - Operations 탭 -> Insert 아이콘 클릭 -> Edit 클릭 -> 마우스 오른쪽 버튼 클릭 -> Name 변경



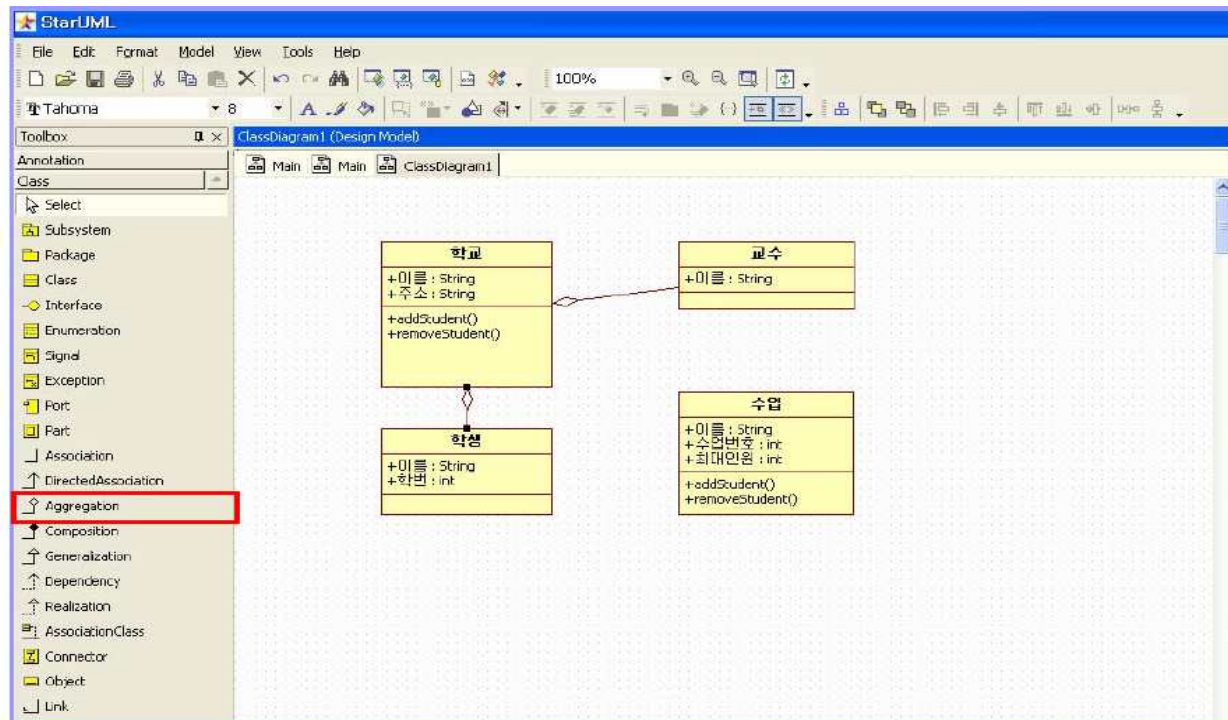
# StarUML 연습 – Class Diagram

- 학교 Class 생성 방법을 참고하여 교수, 수업, 학생 Class도 다음과 같이 생성합니다.



# StarUML 연습 – Class Diagram

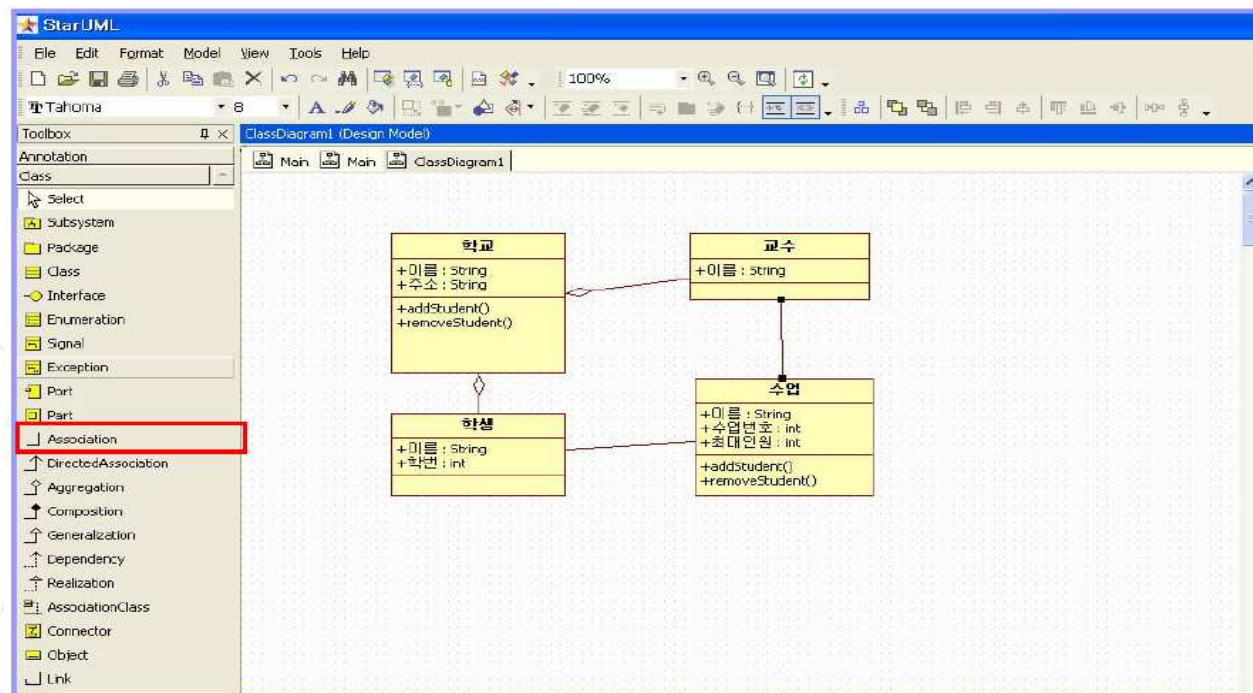
- 클래스간의 관계를 표현해 줍니다.
- 집합연관관계(한 클래스가 다른 클래스를 포함하는 관계를 의미합니다.)
  - Toolbox의 Class 에서 Aggregation 클릭 -> Class와 Class 마우스로 연결





# StarUML 연습 – Class Diagram

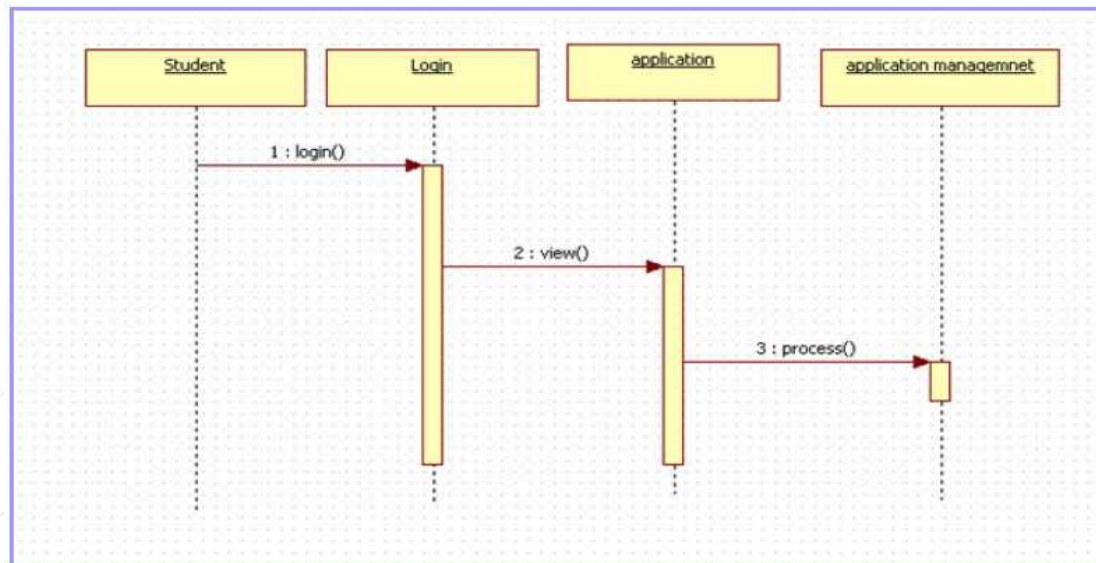
- 연관관계(한 클래스와 다른 클래스가 연관 관계가 있음을 의미함)
  - Toolbox의 Class 에서 Association 클릭 -> Class와 Class 마우스로 연결



# StarUML 실습 – Sequence Diagram

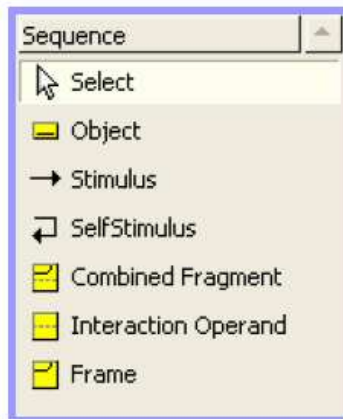
- 수강신청 시스템의 요구사항을 Sequence Diagram으로 표현하면

대기상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다. 로그인 하고 과목목록을 확인한 뒤 과목을 선택한다. 선택한 과목의 수업을 선택하면 수강신청이 완료된다.



# StarUML 실습 – Sequence Diagram

## ▪ Sequence Diagram Tool Bar



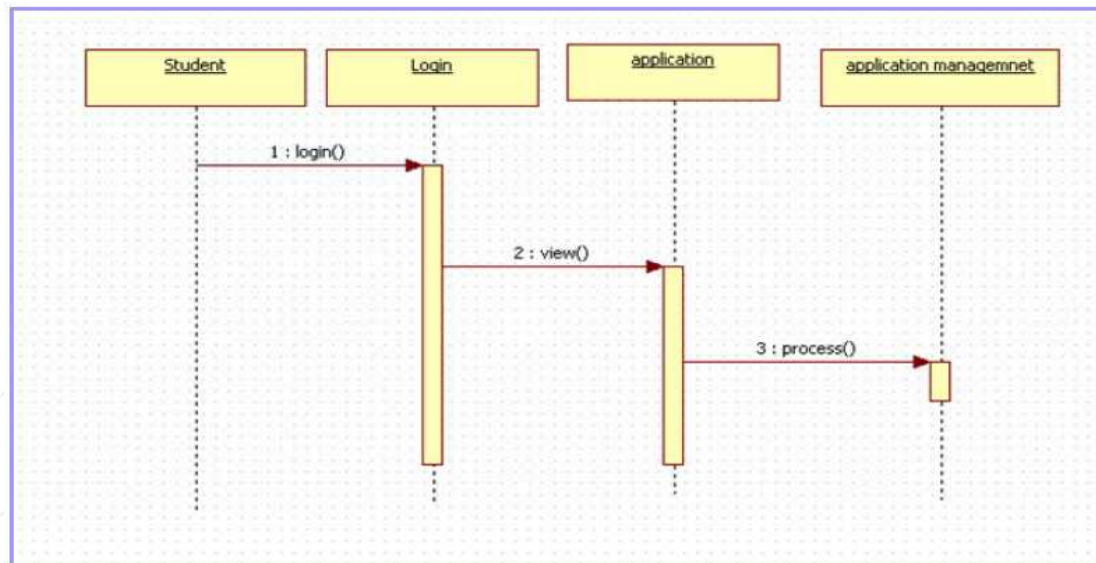
Select	Diagram 요소를 선택합니다.
Object	클래스로부터 생성된 객체를 의미합니다.
Stimulus	두 객체간의 커뮤니케이션을 의미합니다.
SelfStimulus	한 객체가 스스로 커뮤니케이션 함을 의미합니다.
Combined Fragment	Fragment 영역을 의미합니다.
Interaction Operand	조각 내부를 구분할 때 사용합니다.
Frame	시퀀스 다이어그램을 특정 영역으로 구분할 때 사용합니다.



# StarUML 실습 – Sequence Diagram

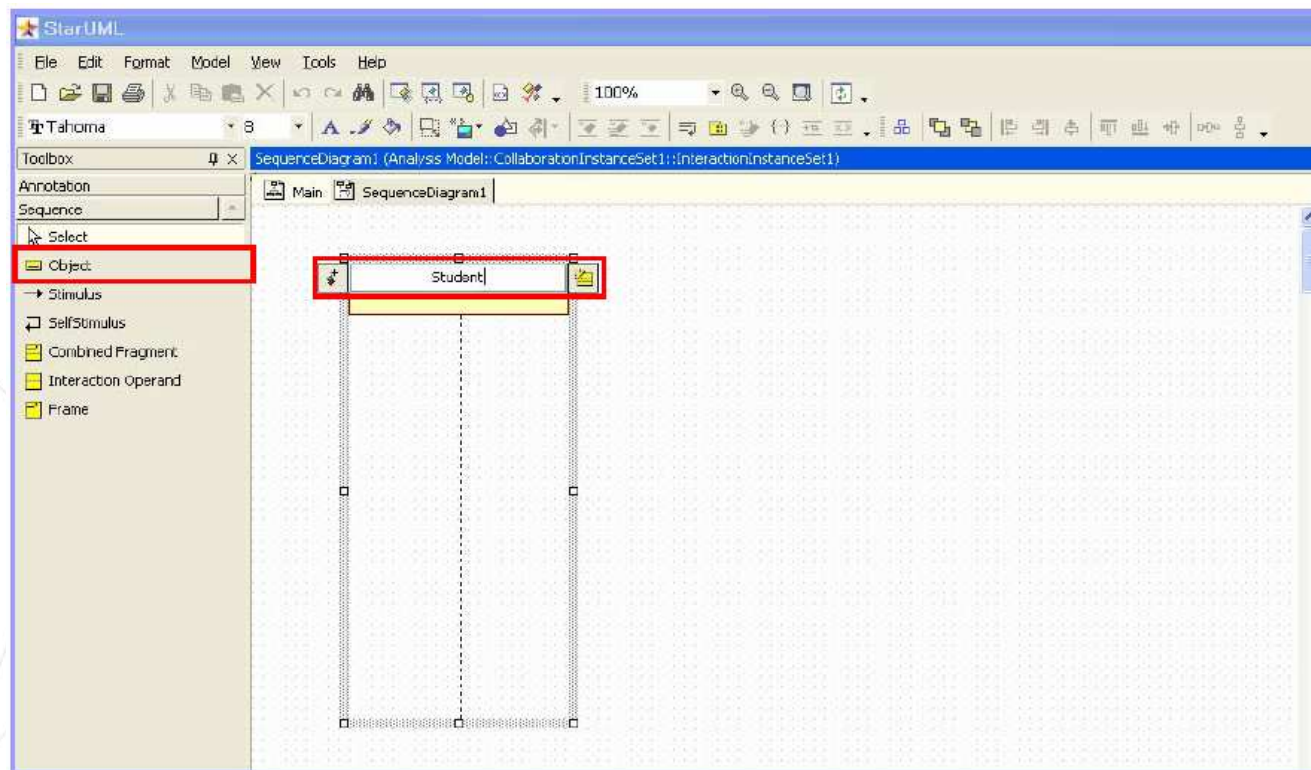
- 수강신청 시스템의 요구사항을 Sequence Diagram으로 표현하면

대기상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다. 로그인 하고 과목목록을 확인한 뒤 과목을 선택한다. 선택한 과목의 수업을 선택하면 수강신청이 완료된다.

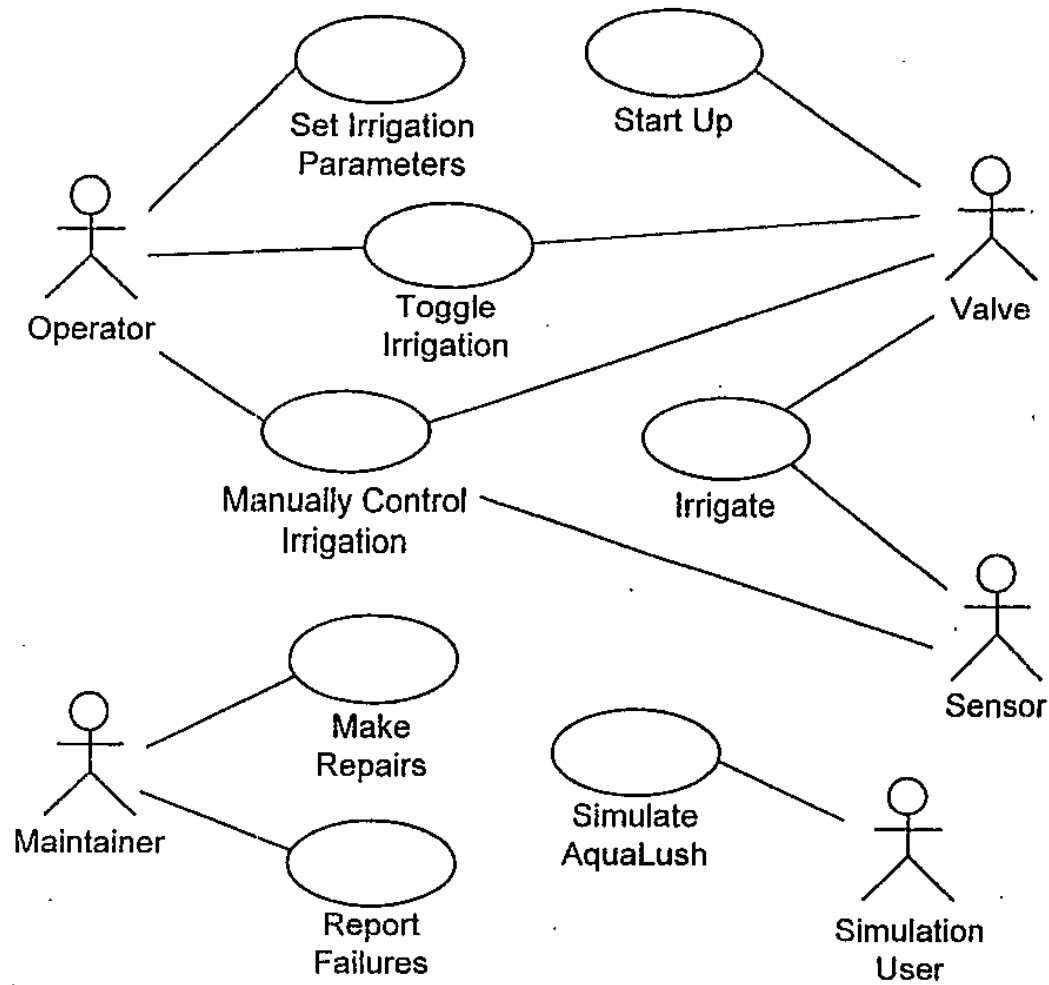


# StarUML 실습 – Sequence Diagram

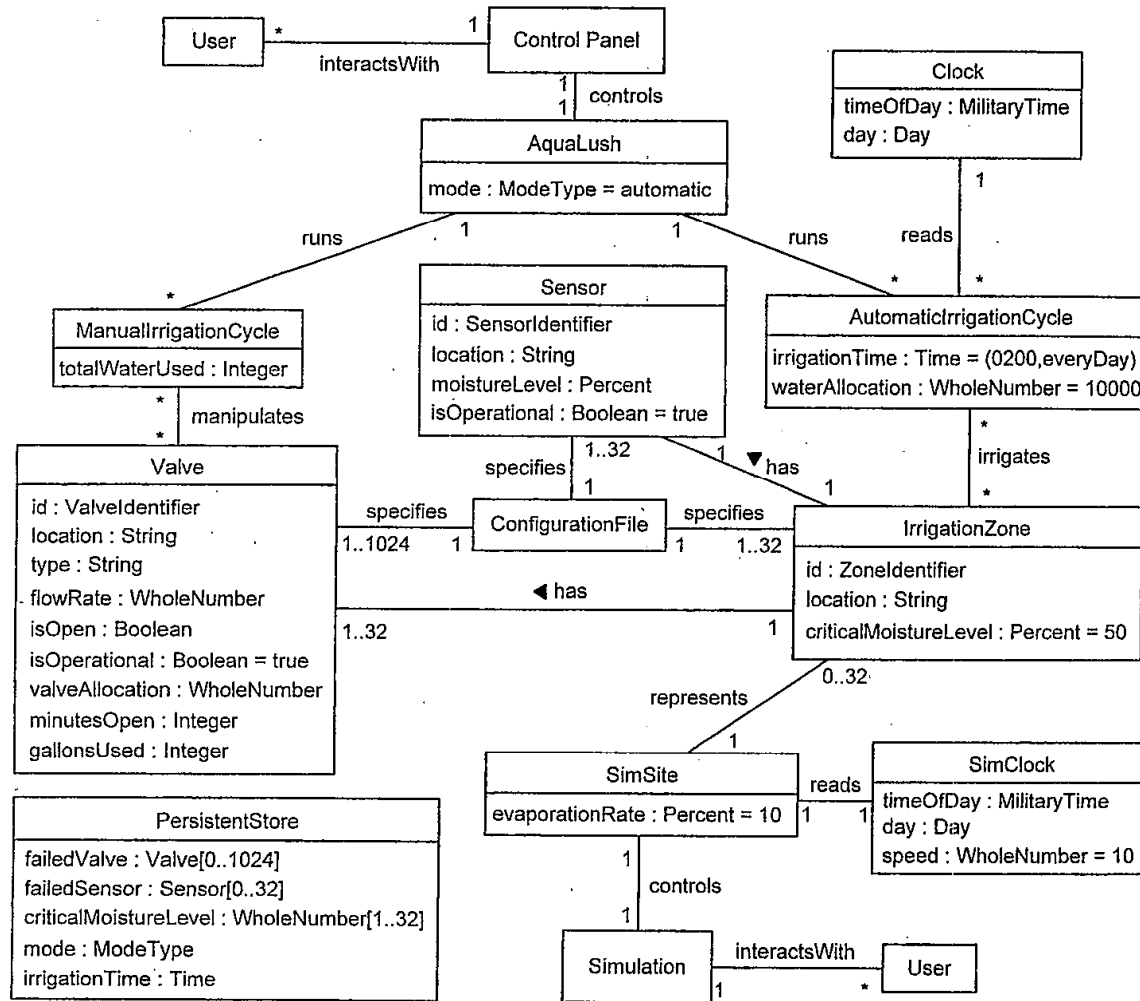
- 상호작용에 참여하는 개체들을 표시합니다.
  - Toolbox의 Sequence 에서 Object 아이콘 클릭 -> 마우스로 Object 크기 지정 -> Object 이름 입력 -> Enter



# 연습 과제 2-1: 사용 사례 다이어그램



# 연습 과제 2-2: 클래스 다이어그램



# 연습 과제 2-3: 순서 다이어그램

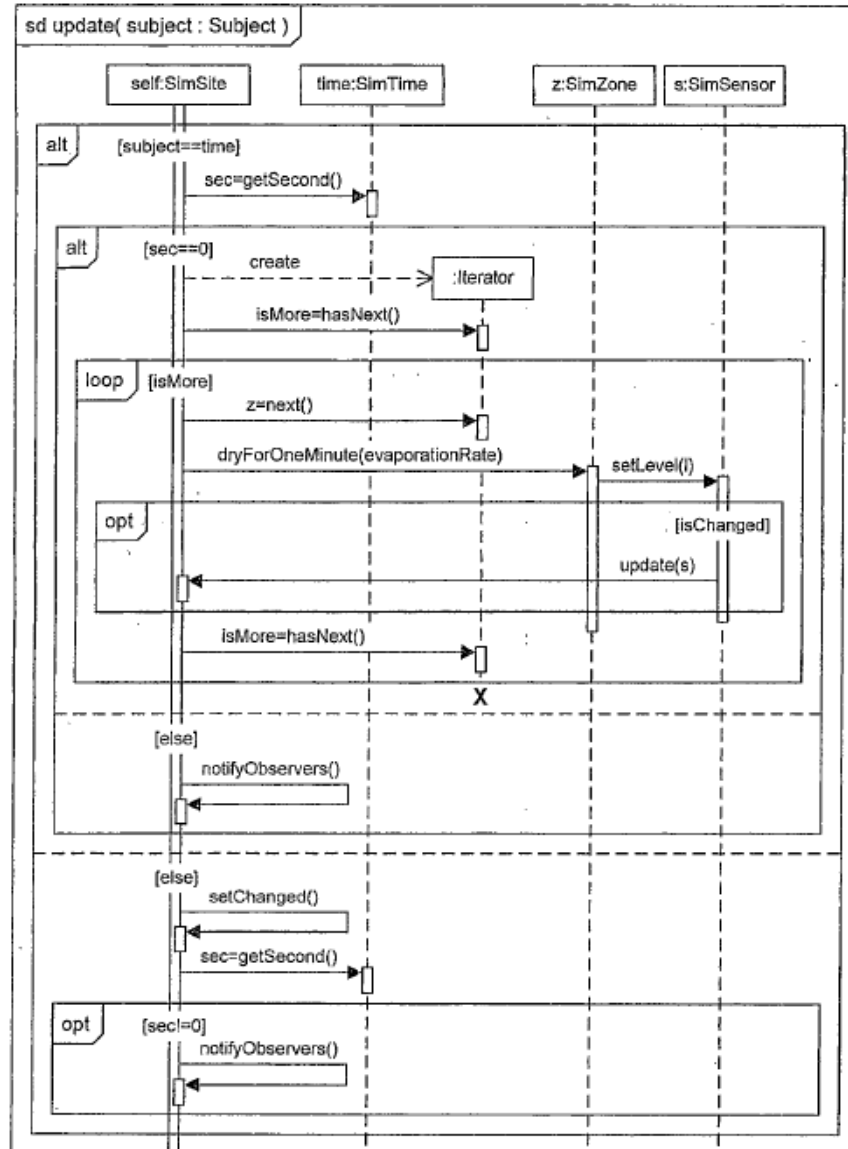


Figure 9-11-9 SimSite update() Behavior