

# StarUML

1	도구 개요
2	도구 설치 방법
3	도구 기능 소개
4	도구 활용 예제
5	FAQ
6	도구 평가
7	용어집

# 목차

- 1. 도구 개요
- 2. 도구 설치 방법
  - 2.1 StarUML 다운받기
  - 2.2 StarUML 설치하기
  - 2.3 StarUML 설치확인하기
- 3. 도구 기능 소개
  - 3.1 주요 기능
  - 3.2 StarUML 의 화면 소개
  - 3.3 마우스 사용방법
  - 3.4 공통 기능
  - 3.5 예제 소개
  - 3.6 StarUML 사용하기
  - 3.7 Use Case Diagram 그리기
  - 3.8 Class Diagram 그리기
  - 3.9 Sequence Diagram 그리기
  - 3.10 Activity Diagram 그리기
  - 3.11 Statechart Diagram 그리기
  - 3.12 Collaboration Diagram 그리기
- 4. 도구 활용 예제
  - 4.1 예제소개
  - 4.2 Use Case Diagram
  - 4.3 Class Diagram
  - 4.4 Sequence Diagram4
  - .5 Collaboration Diagram
  - 4.6 Activity Diagram
  - 4.7 Statechart Diagram
- 5. FAQ
- 6. 도구 평가
- 7. 용어집

# 1. 도구 개요

# 1. 도구 개요

소개	UML 모델링 도구로 Use Case Diagram, Class Diagram, Sequence Diagram 등 다양한 Diagram을 간편한 UI를 이용하여 쉽고 빠르게 생성할 수 있도록 편리한 기능을 제공합니다.		
주요 기능	Use Case Diagram, Class Diagram, Collaboration Diagram, Statechart Diagram, Activity Diagram, Sequence Diagram, Component Diagram, Deployment Diagram, Composite Diagram 생성		
카테고리	Design-Modeling	세부 카테고리	설계 & 모델링
커버리지	Design-Modeling	도구 난이도	하
라이선스 형태 / 비용	GPL (GNU Public License) / 무료	사전 설치 도구	-
사용 환경	운영체제	Windows	
	Eclipse 환경	-	
특징	<ul style="list-style-type: none"><li>• UML 표준 명세에 기반한 모델 작성과 UML2.0 표기법을 지원합니다.</li><li>• 완전한 UML Profile 개념을 제공하여 플랫폼에 독립적인 모델을 작성할 수 있도록 지원합니다.</li><li>• 뛰어난 확장성과 유연성을 제공합니다.</li><li>• 심플한 인터페이스로 학습 및 사용이 용이하며 개발 효율을 높일 수 있습니다.</li></ul>		
적용회사 / 프로젝트	SK 텔레콤 / WCDMA NMS (Widearea-CDMA Network Management System)		
관련 도구	UMLet, VioletUMLEditor, Rational Rose, Together		
제작사	StarUML		
공식 홈페이지	<a href="http://staruml.sourceforge.net/ko/">http://staruml.sourceforge.net/ko/</a>		
개발자	-		

## 2. 도구 설치 방법

## 2. 도구 설치 방법

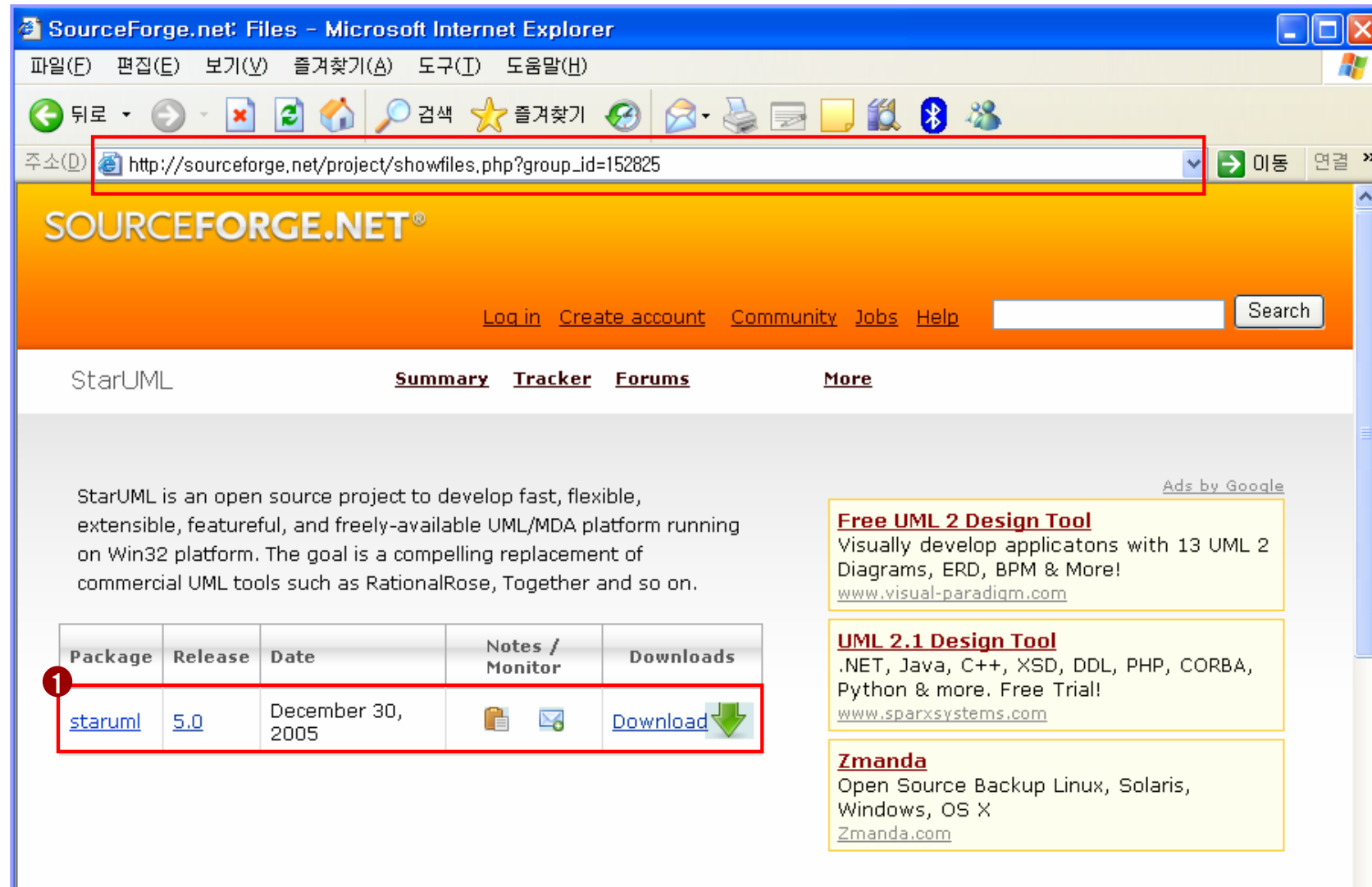
### 세부 목차

- 2.1 StarUML 다운받기
- 2.2 StarUML 설치하기
- 2.3 StarUML 설치확인하기

## 2. 도구 설치 방법

### 2.1 StarUML 다운 받기 (1/4)

- 다음의 사이트에서 StarUML을 다운 받습니다.
  - [http://sourceforge.net/project/showfiles.php?group\\_id=152825](http://sourceforge.net/project/showfiles.php?group_id=152825)



## 2. 도구 설치 방법

### 2.1 StarUML 다운 받기 (2/4)

- staruml-5.0-uml-cm.exe 파일을 클릭합니다.

SourceForge.net: Files - Microsoft Internet Explorer

주소(D) http://sourceforge.net/project/showfiles.php?group\_id=152825&package\_id=169190&release\_id=437438

**SOURCEFORGE.NET**

[Log in](#) [Create account](#) [Community](#) [Jobs](#) [Help](#)

StarUML **Summary** [Tracker](#) [Forums](#) [More](#)

**You have selected the 5.0 release.**  
Please choose the file that best matches your architecture or operating system from the list of files contained in this release.

[Options](#)

Package	Release	Filename	Size	Architecture
<a href="#">staruml</a>				
Latest	<a href="#">5.0</a> (2005-12-30 07:58)			
		<a href="#">staruml-5.0-with-cm.exe</a>	22725959	i386
		<a href="#">staruml-src.zip</a>	20999693	i386
Totals:	1	2	43725652	

**Free UML 2 Design Tool**  
Visually develop applications with 13 UML 2 Diagrams, ERD, BPM & More!  
[www.visual-paradigm.com](http://www.visual-paradigm.com)

**UML 2.1 Design Tool**  
.NET, Java, C++, XSD, DDL, PHP, CORBA, Python & more. Free Trial!  
[www.sparxsystems.com](http://www.sparxsystems.com)

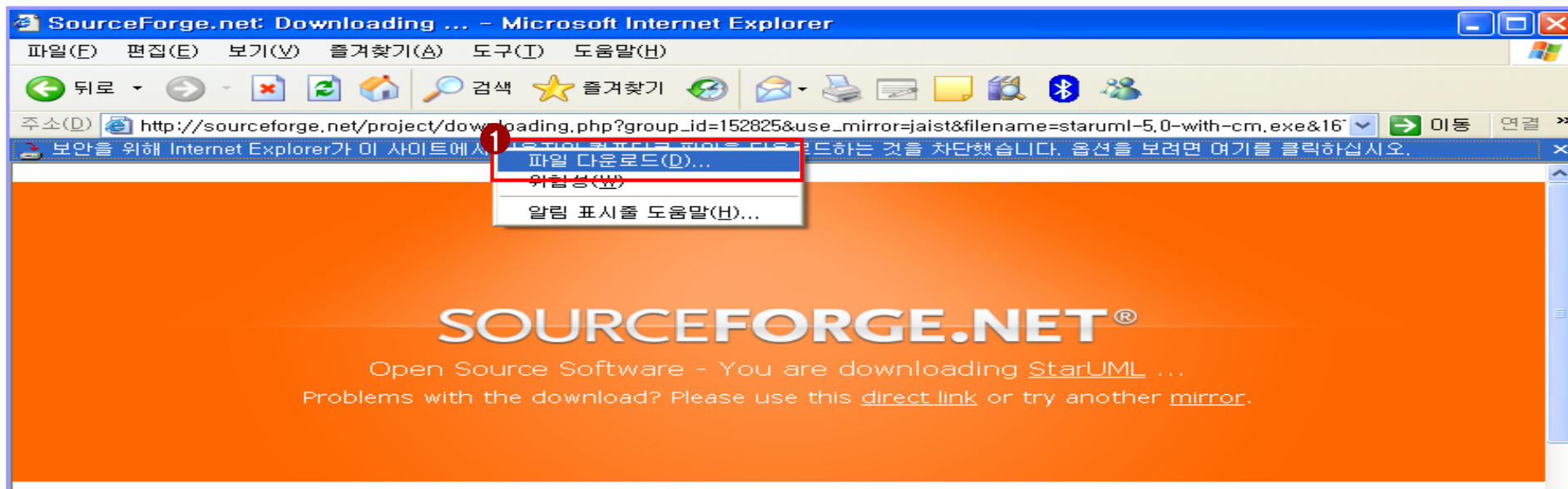
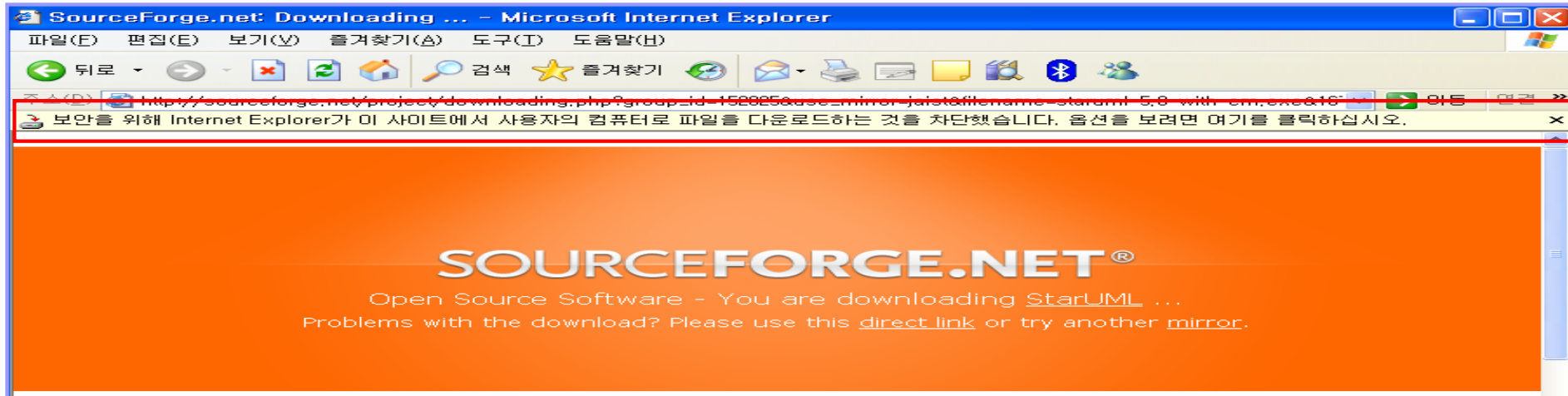
**Zmanda**  
Open Source Backup Linux, Solaris, Windows, OS X  
[Zmanda.com](http://Zmanda.com)

**UML 2.1 Modeling Tools**  
Superior Usability & Easy To Learn UML

## 2. 도구 설치 방법

### 2.1 StarUML 다운 받기 (3/4)

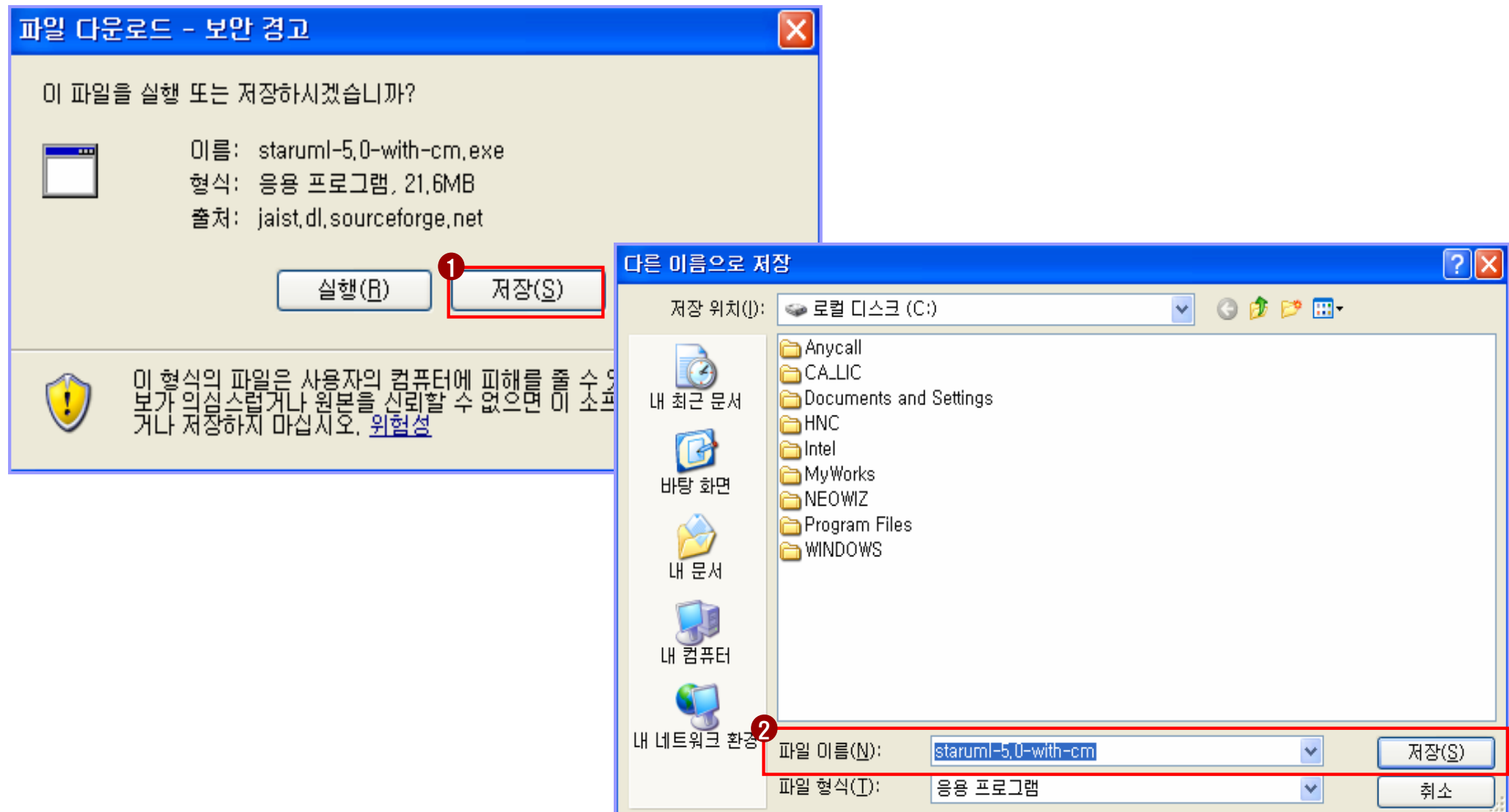
- 보안을 유지하기 위해 Internet Explorer에서 다음처럼 다운로드를 차단하는 경우가 있습니다. 차단을 해제하고 파일 다운로드를 클릭하여 파일을 다운 받습니다.



## 2. 도구 설치 방법

### 2.1 StarUML 다운 받기 (4/4)

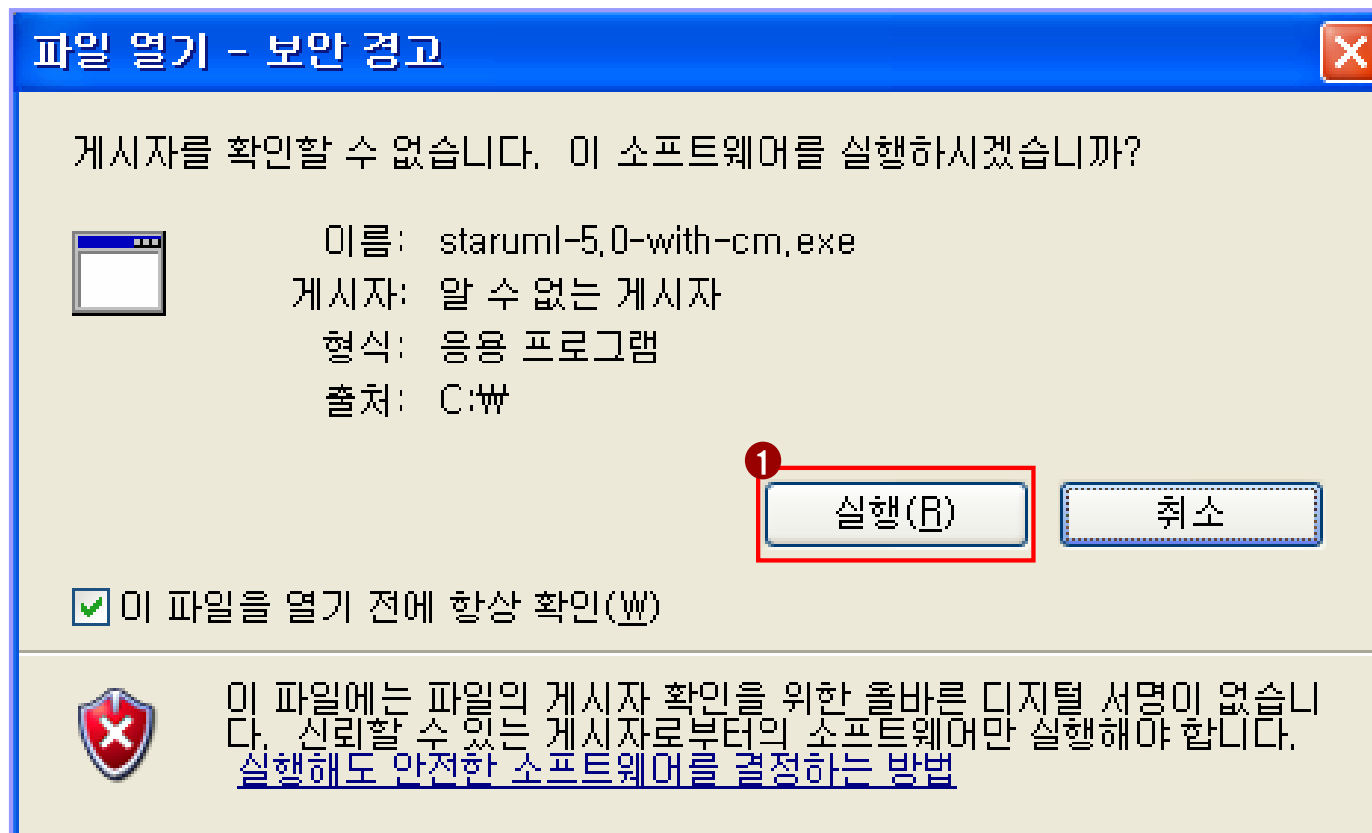
- 파일 다운로드 창에서 저장을 눌러 로컬디스크(C:)에 저장 합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 [1/8]

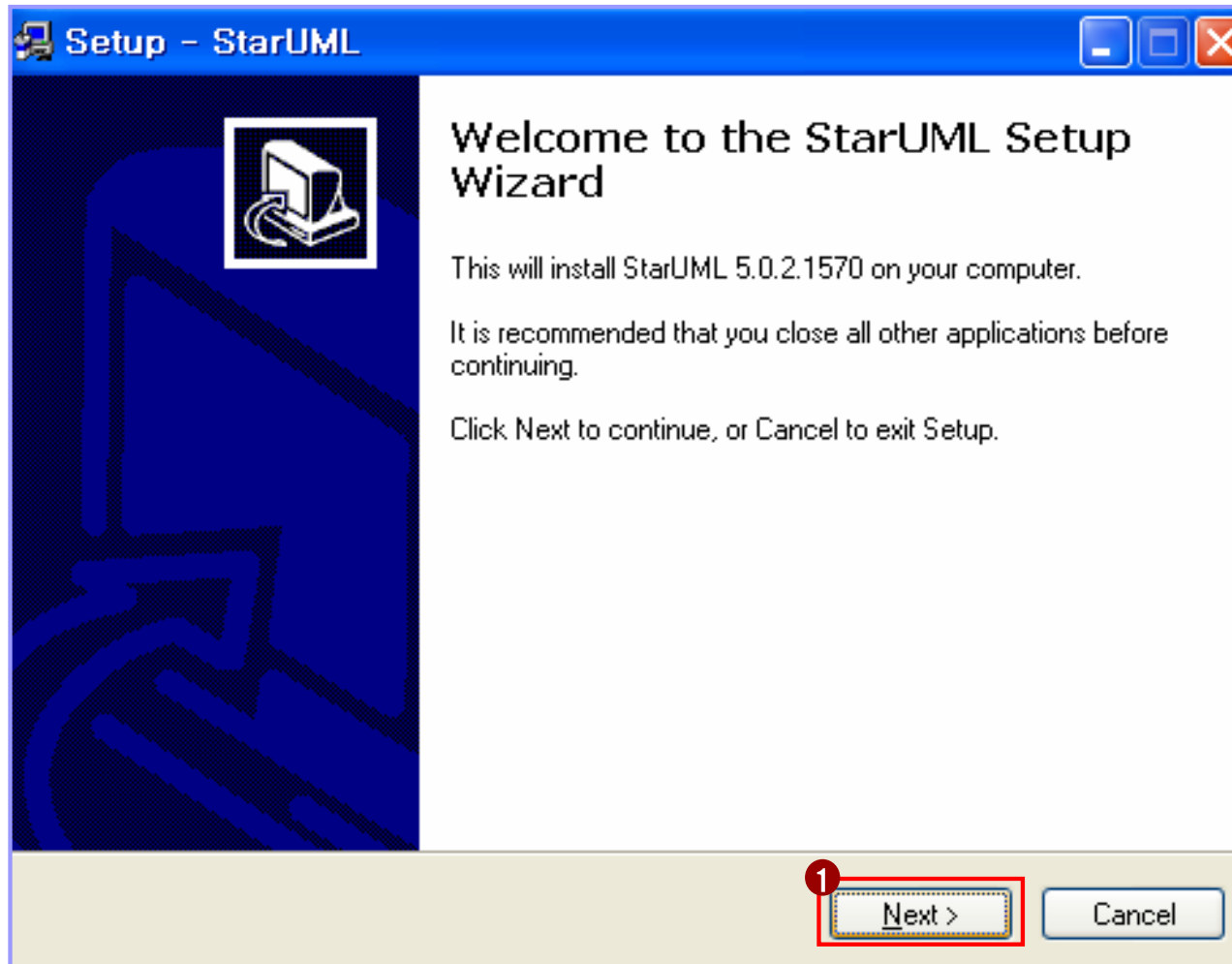
- 로컬 디스크에 다운 받은 starUML-5.0-with-cm.exe 파일을 실행시키면 설치가 시작됩니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 [2/8]

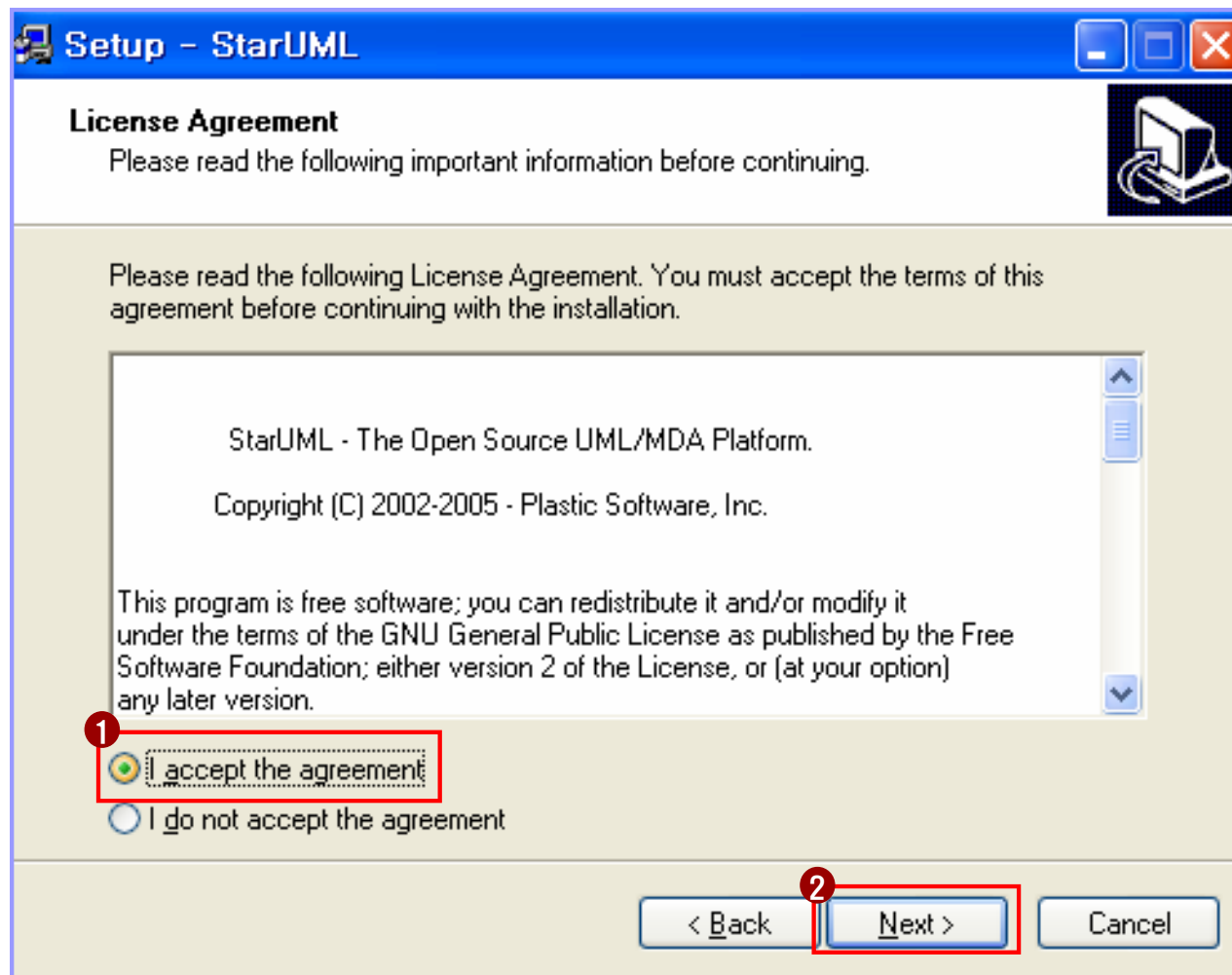
- StarUML설치 마법사창에서 Next 버튼을 클릭합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 (3/8)

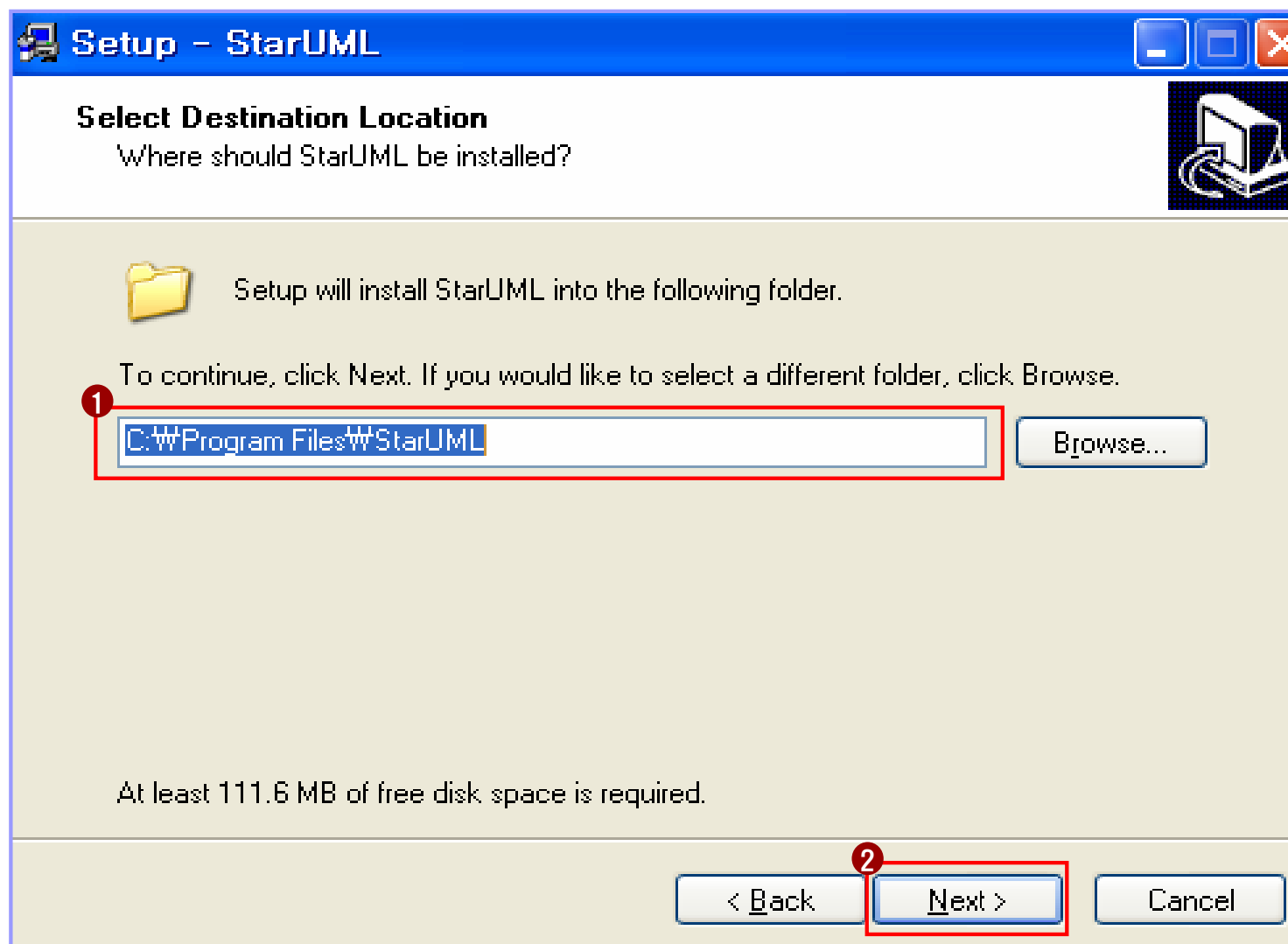
- 라이선스 동의화면에서 I accept the agreement 버튼을 클릭하고 Next 버튼을 클릭합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 [4/8]

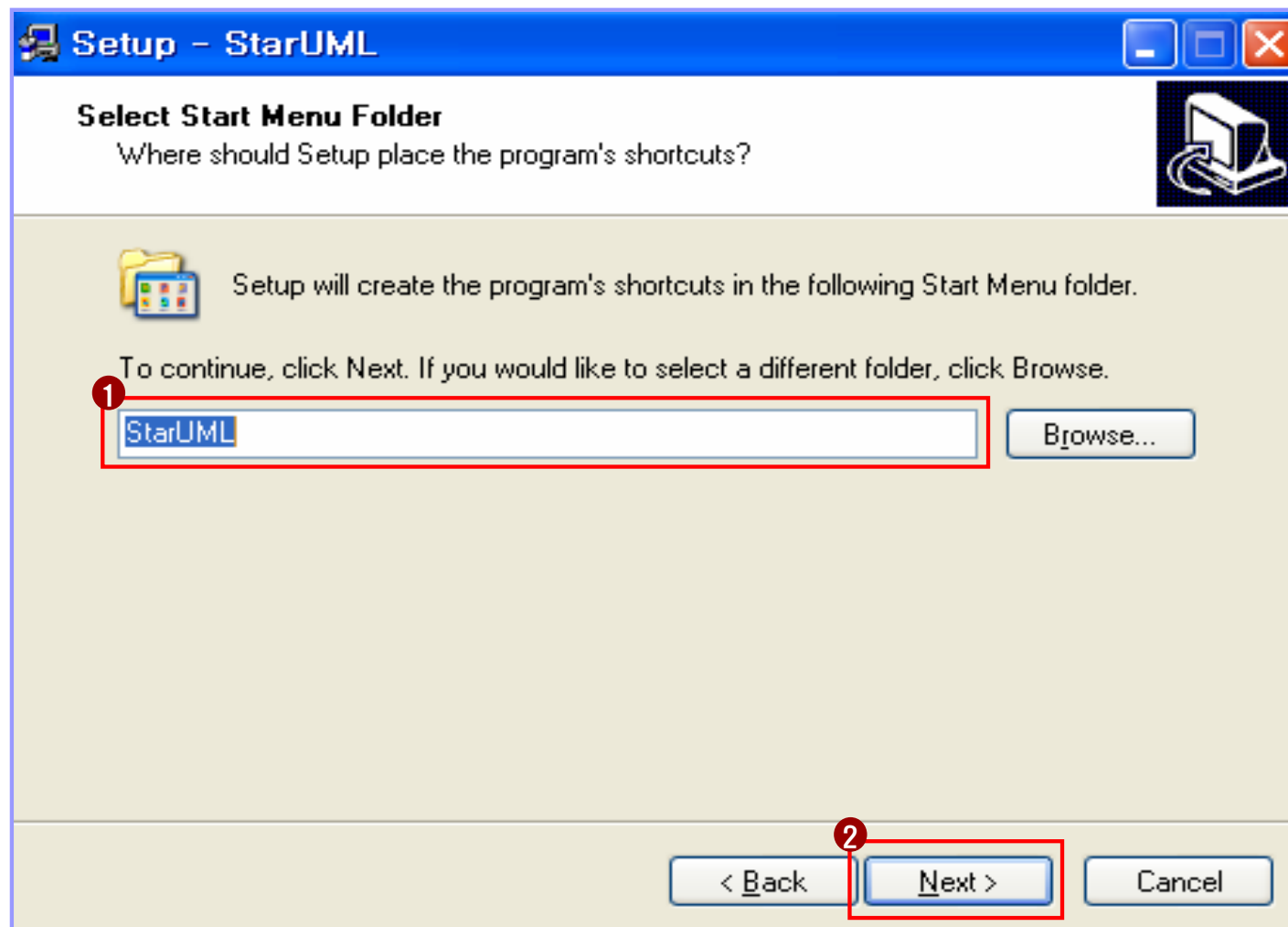
- 설치 파일을 저장할 위치를 지정하고 Next버튼을 클릭합니다.
  - 본 예제에서는 C:\Program Files\StarUML 위치에 파일을 설치합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 (5/8)

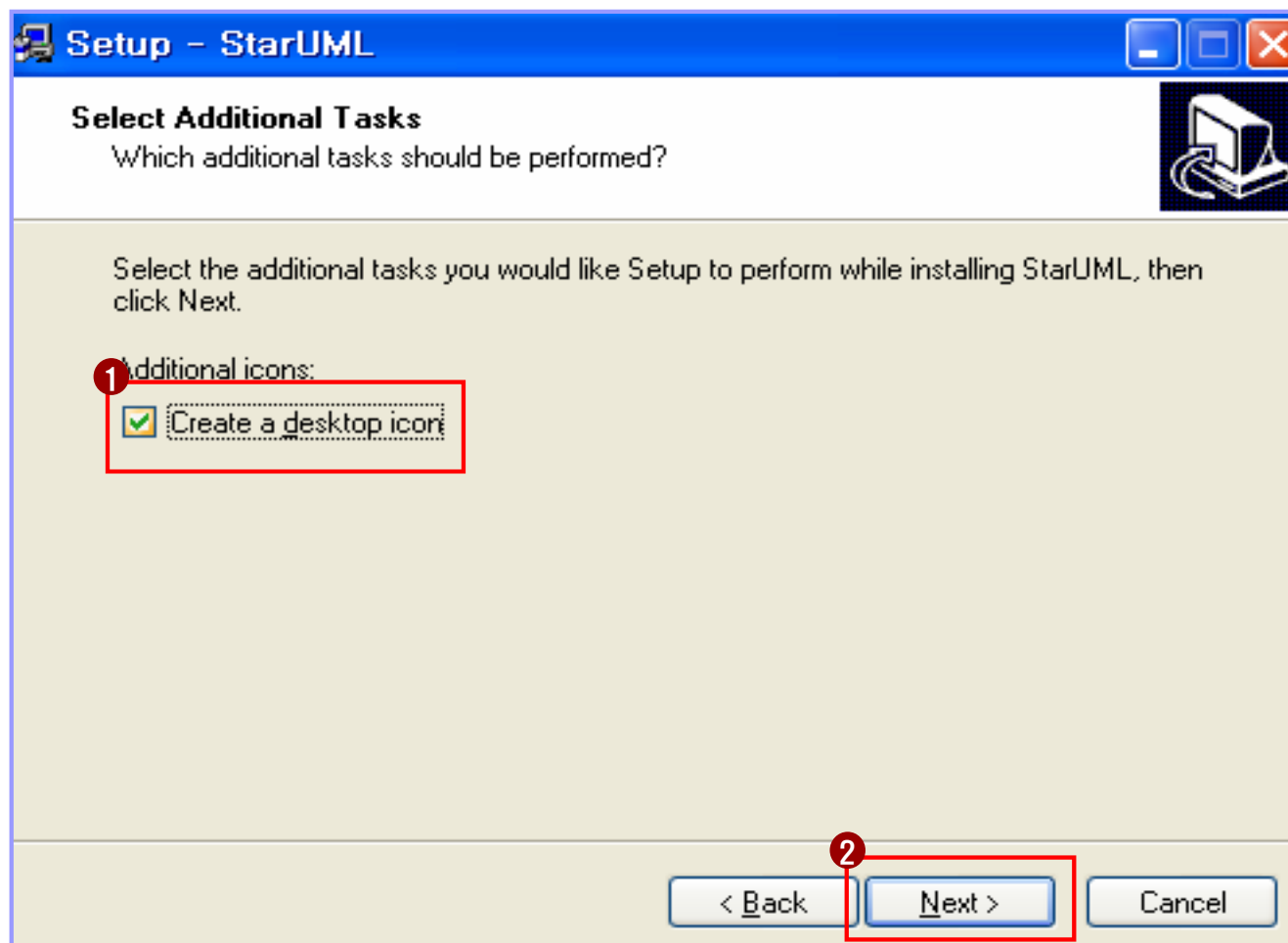
- Setup will create the program's shortcuts in the following Star Menu folder.
  - 본 예제에서는 StarUML로 지정합니다.
  - Next 버튼 클릭합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 [6/8]

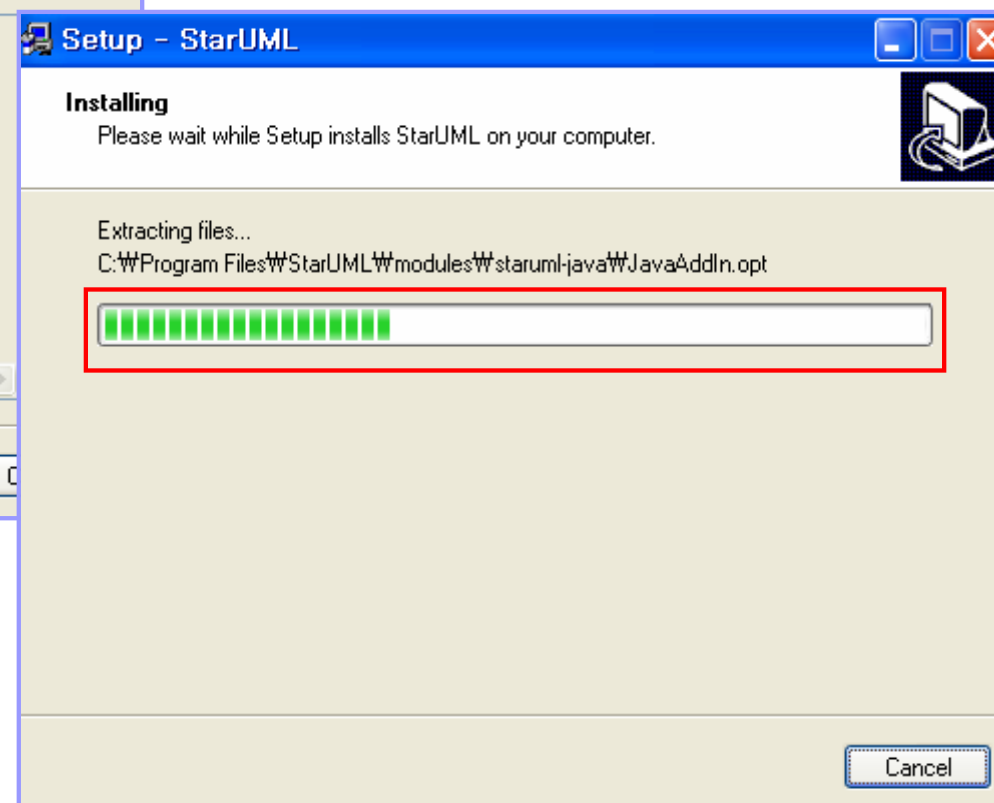
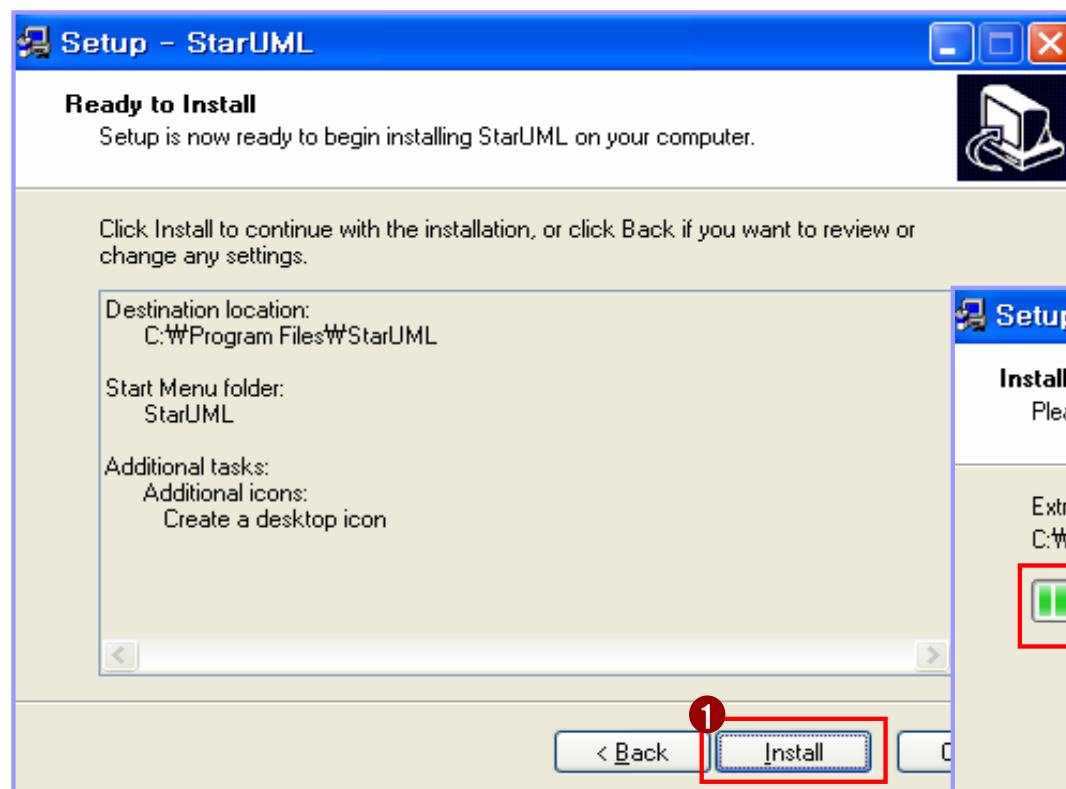
- 바탕화면에 StarUML 아이콘 표시 여부를 선택합니다.
  - 아이콘 표시 하기 위해서는 Create a desktop icon을 클릭합니다.
  - 본 예제에서는 바탕화면에 아이콘을 표시합니다.
  - Next 버튼을 클릭합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 (7/8)

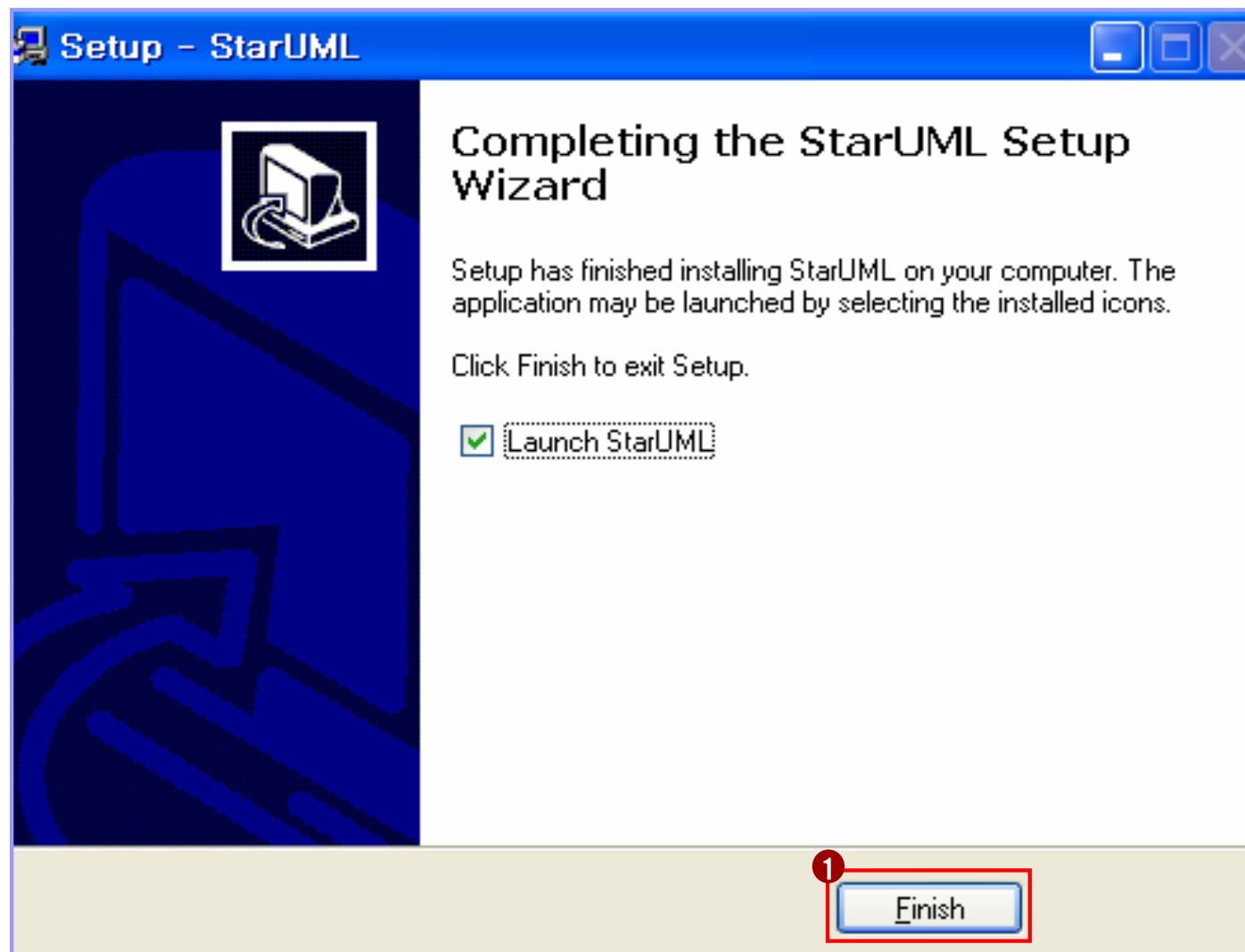
- StarUML 설치를 시작합니다.
  - Install 버튼을 클릭합니다.



## 2. 도구 설치 방법

### 2.2 StarUML 설치하기 (8/8)

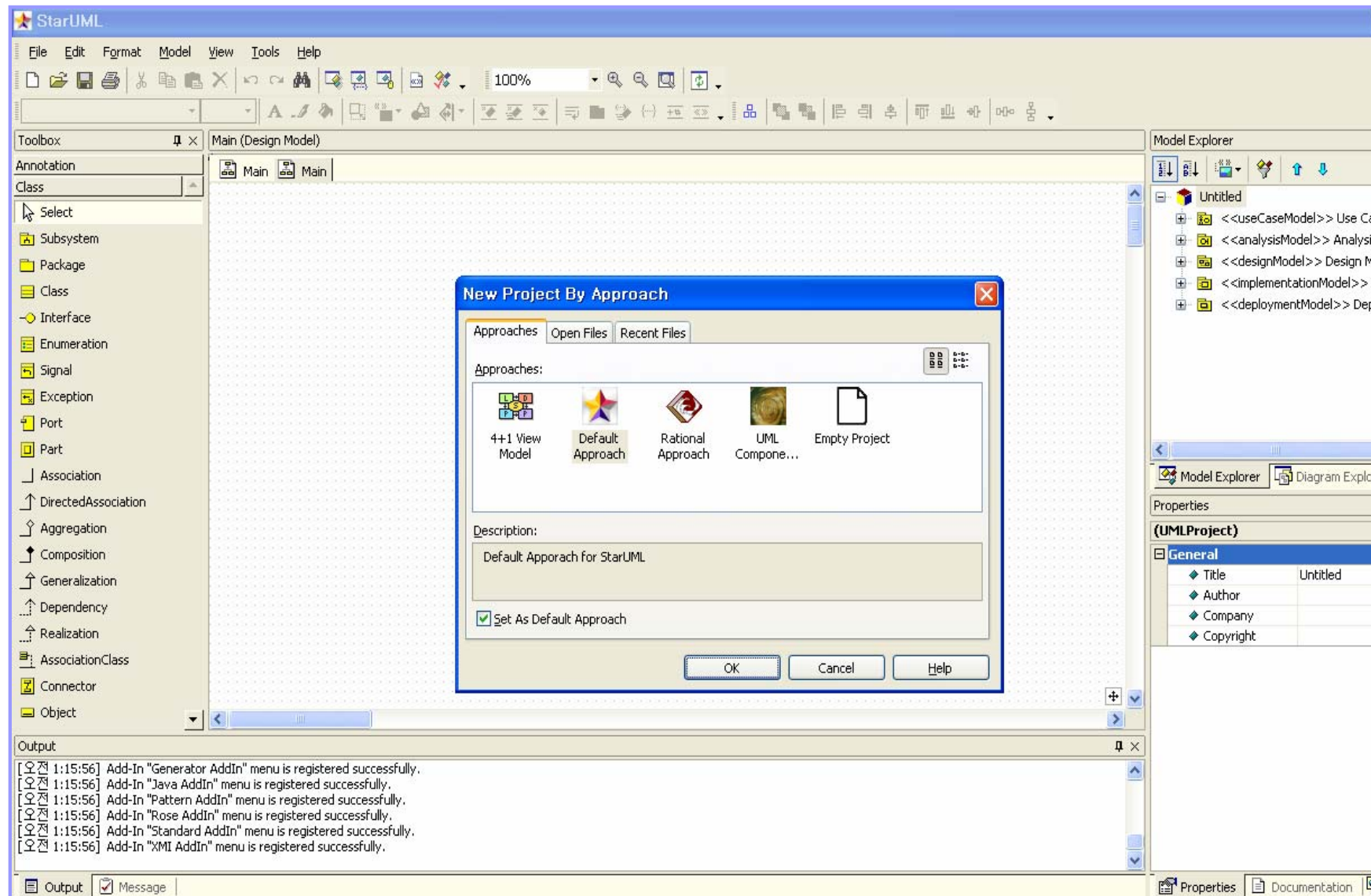
- StarUML 설치를 마칩니다.
  - 설치 완료 후, Finish 버튼을 클릭합니다.



## 2. 도구 설치 방법

### 2.3 StarUML 설치 확인하기

- StarUML 실행 화면입니다.



### **3. 도구 기능 소개**

## 3. 도구 기능 소개

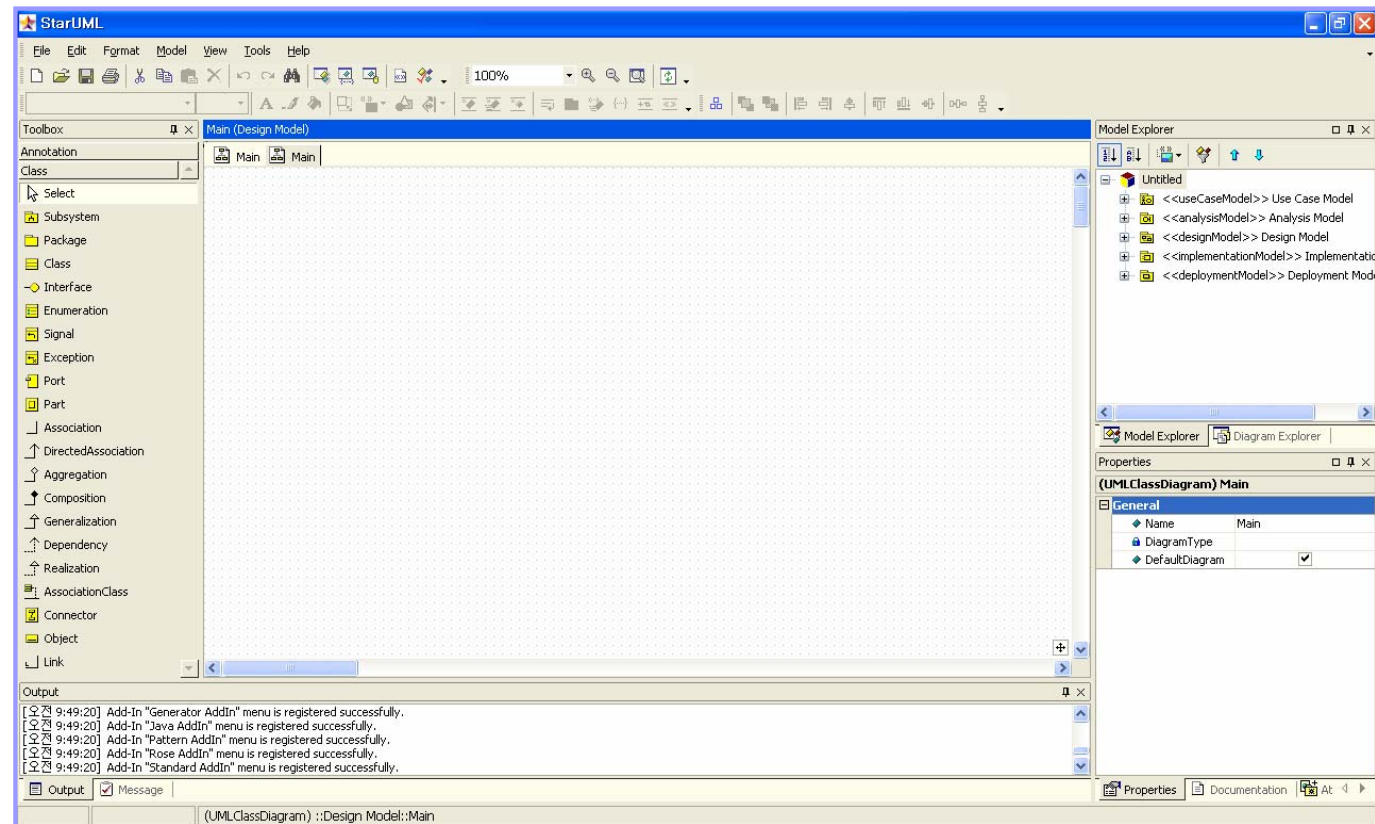
### 세부 목차

- 3.1 주요 기능
- 3.2 StarUML 의 화면 소개
- 3.3 마우스 사용방법
- 3.4 공통 기능 3.5 예제 소개
- 3.6 StarUML 사용하기
- 3.7 Use Case Diagram 그리기
- 3.8 Class Diagram 그리기
- 3.9 Sequence Diagram 그리기
- 3.10 Activity Diagram 그리기
- 3.11 Statechart Diagram 그리기
- 3.12 Collaboration Diagram 그리기

## 3. 도구 기능 소개

### 3.1 주요 기능

- StarUML를 이용하면 UML Diagram을 빠르고 쉽게 그릴 수 있습니다.
- 모델을 소스 코드로 생성하고, 소스 코드로부터 모델을 Reverse Engineering 할 수 있습니다.
- .NET, J2EE와 같은 플랫폼 환경을 적용시킬 수 있습니다.
- 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

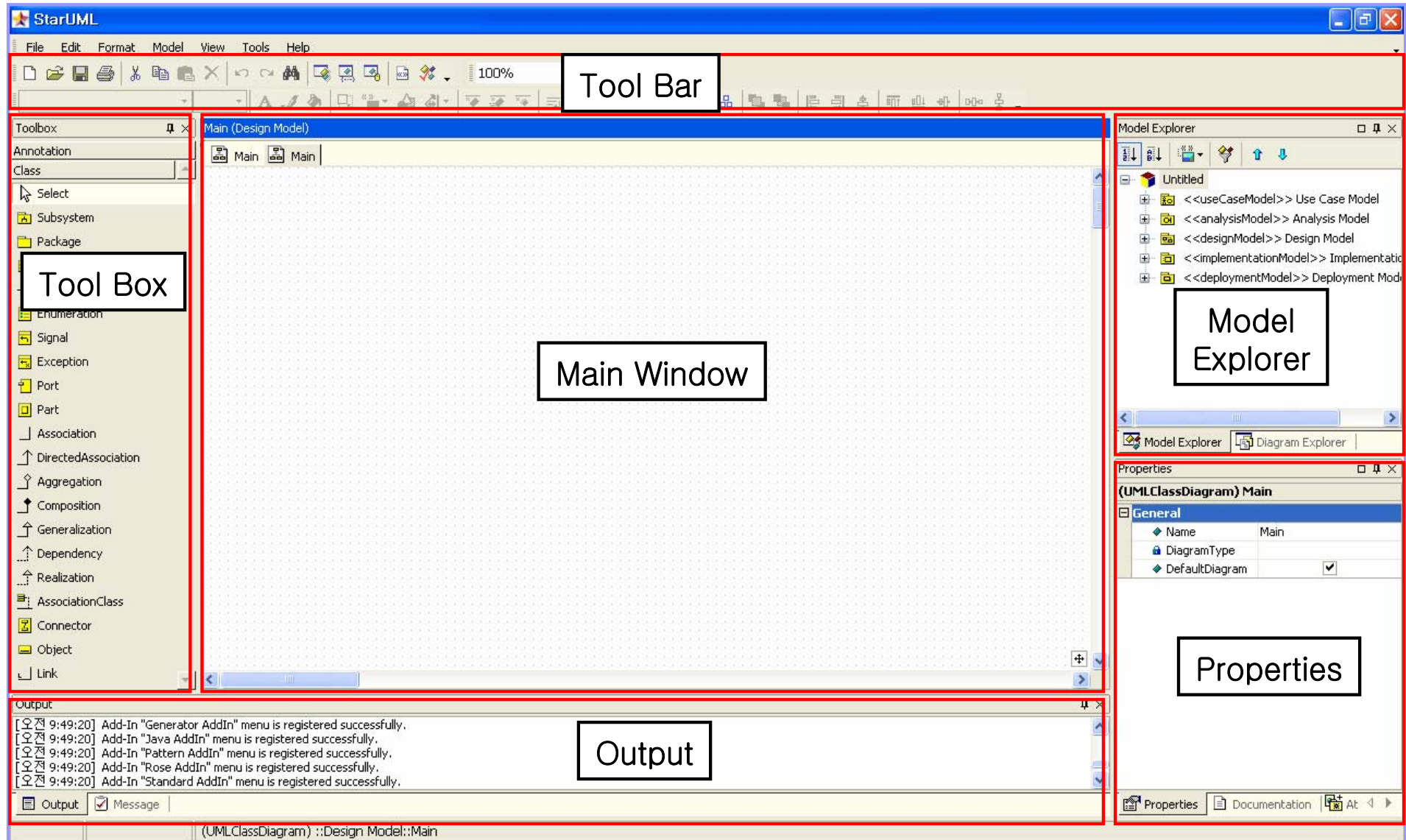


[StarUML 메인 화면]

# 3. 도구 기능 소개

## 3.2 StarUML 의 화면 소개

- Violet UML Editor는 아래 그림과 같이 6개 화면으로 구성되어 있습니다.



### 3. 도구 기능 소개

#### 3.3 마우스 사용방법 (1/3)

- Violet UML Editor에서 마우스 사용방법입니다.

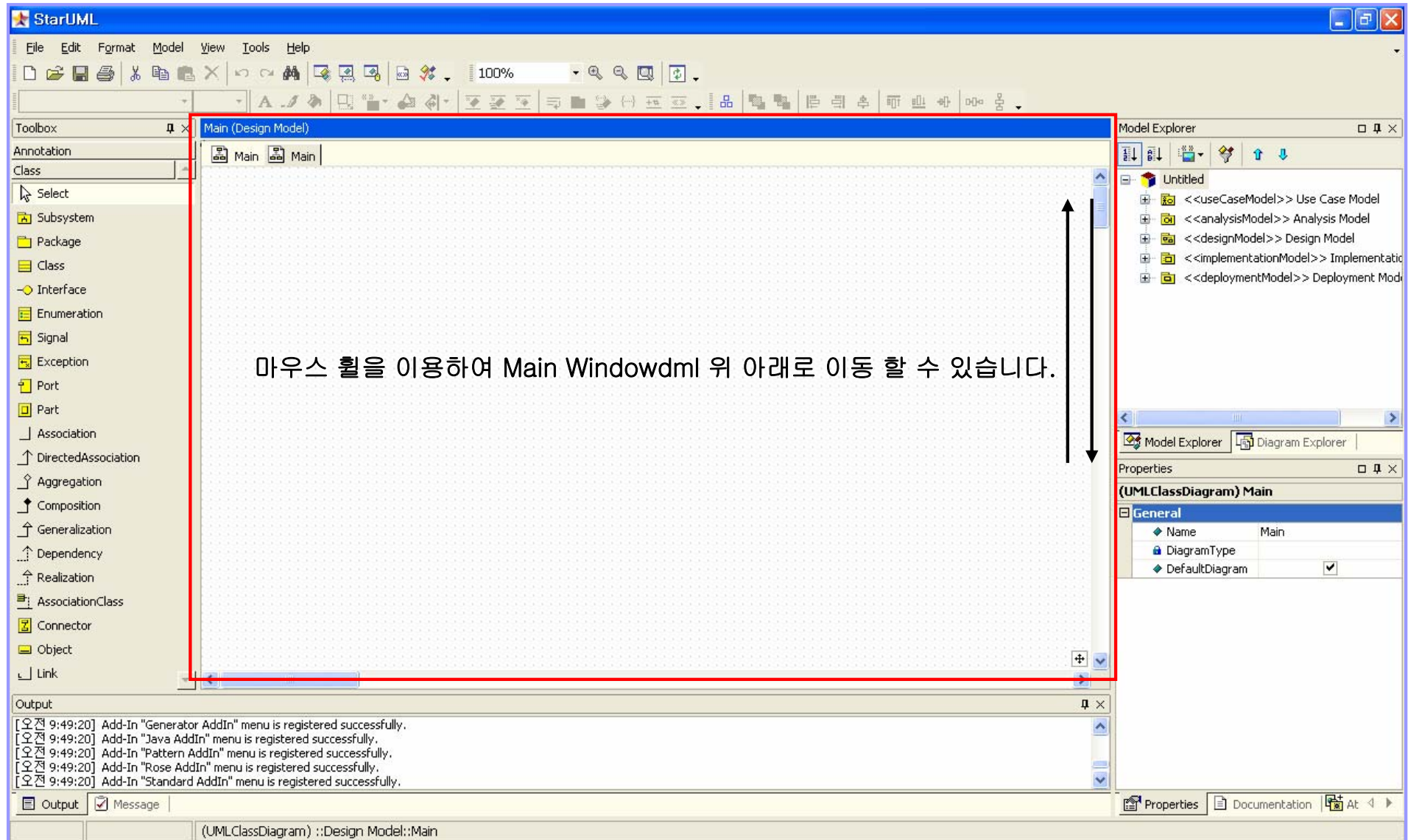


1 . 마우스 휠	Main Window의 화면 이동
2. 마우스 왼쪽 버튼	선택된 Diagram 요소를 Main Window 화면 영역에 추가하기

# 3. 도구 기능 소개

## 3.3 마우스 사용방법 (2/3)

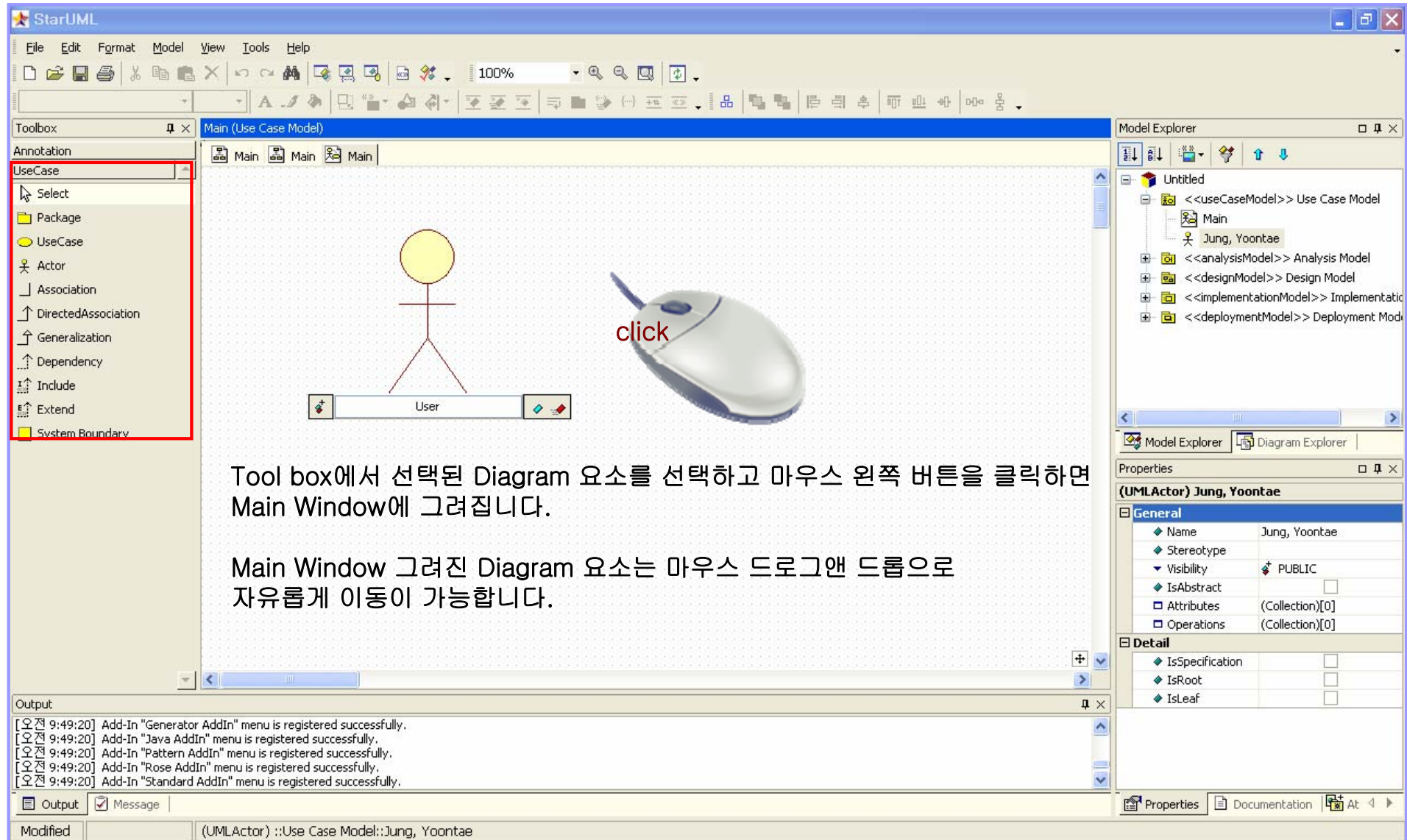
- StarUML에서 마우스 휠 사용방법 입니다.



# 3. 도구 기능 소개

## 3.3 마우스 사용방법 (3/3)

- Violet UML Editor에서 마우스 왼쪽 버튼 사용방법 입니다.



# 3. 도구 기능 소개

## 3.4 공통 기능 (1/4)

- StarUML 공통 Tool Bar
  - 표준 도구모음



	New Project[Ctrl+N]	새로운 프로젝트를 만듭니다.
	Open[Ctrl+O]	프로젝트 파일을 불러옵니다.
	Save[Ctrl+S]	프로젝트 파일을 저장합니다.
	Print[Ctrl+P]	다이어그램을 프린트로 인쇄합니다.
	Cut[Ctrl+X]	선택된 요소들을 클립보드에 복사하고 잘라냅니다.
	Copy[Ctrl+C]	선택된 요소들을 클립보드에 복사합니다.
	Paste[Ctrl+V]	클립보드의 내용을 현재 선택된 요소(혹은 다이어그램)로 붙여 넣습니다.
	Delete[Del]	다이어그램에 선택된 뷰 요소들을 삭제합니다.
	Undo[Ctrl+Z]	사용자가 가장 최근에 수행한 액션을 취소합니다.
	Redo[Ctrl+Y]	사용자가 취소한 액션을 다시 실행합니다.
	Find[Ctrl+F]	요소를 검색합니다.
	Collection Editor[Ctrl+F5]	현재 선택된 모델 요소의 하위 요소들을 편집하기 위한 컬렉션 편집기가 나타납니다.
	Constraints[Ctrl+F6]	현재 선택된 모델 요소의 제약사항을 편집하기 위한 제약사항 편집기가 나타납니다.
	TaggedValues[Ctrl+F7]	현재 선택된 모델 요소의 확장속성을 편집하기 위한 확장속성 편집기가 나타납니다.
	Profiles[Ctrl+F8]	프로파일 관리자가 나타납니다.
	Verify Model[F9]	현재 프로젝트에 존재하는 모델 요소들을 검사하기 위한 모델 검사 대화상자가 나타납니다.

# 3. 도구 기능 소개

## 3.4 공통 기능 (2/4)

- StarUML 공통 Tool Bar
  - 서식 도구모음







Tahoma	Font Face	선택된 뷰 요소들의 글꼴 모양을 지정합니다.
8	Font Size	선택된 뷰 요소들의 글꼴 크기를 지정합니다.
	Font	선택된 뷰 요소들의 글꼴을 지정합니다.
	Line Color	선택된 뷰 요소들의 선 색상을 지정합니다.
	Fill Color	선택된 뷰 요소들의 채움 색상을 지정합니다.
	Auto Resize	선택된 뷰 요소들의 크기를 자동으로 결정합니다.
	Stereotype Display	선택된 뷰 요소들의 스테레오타입 표시 형식을 지정합니다.
	Show As Extended Notation	선택된 뷰 요소들을 확장 노테이션으로 표시하도록 지정합니다.
	Line Style	선택된 연결 뷰 요소들의 선 모양을 지정합니다.
	Suppress Attributes[Shift+Ctrl+A]	선택된 뷰 요소들(e.g. 클래스, 유스케이스 등)의 속성을 표시하는 영역을 감추어 나타내지 않습니다.
	Suppress Operations[Shift+Ctrl+O]	선택된 뷰 요소들(e.g. 클래스, 서브시스템 등)의 연산을 표시하는 영역을 감추어 나타내지 않습니다.
	Suppress Literals[Shift+Ctrl+L]	선택된 열거형 뷰의 속성을 표시하는 영역을 감추어 나타내지 않습니다.
	Word Wrap Name	선택된 뷰 요소들의 이름에 워드랩을 적용하여 표시합니다.
	Show Parent Name	선택된 뷰 요소들의 상위 이름을 함께 표현합니다.
	Show Operation Signature	선택된 뷰 요소들(e.g. 클래스, 서브시스템 등)의 연산을 시그너처까지 완전하게 표현합니다.
	Show Properties	선택한 뷰 요소들의 확장 속성 등의 프로퍼티를 표시합니다.
	Show Compartment Visibility	선택된 뷰 요소들이 가진 구획(e.g. 속성 구획, 연산 구획 등) 항목들의 가시성을 표시합니다.
	Show Compartment Stereotype	선택된 뷰 요소들이 가진 구획(e.g. 속성 구획, 연산 구획 등) 항목들의 스테레오타입을 텍스트로 표시합니다.

## 3. 도구 기능 소개

### 3.4 공통 기능 (3/4)

- StarUML 공통 Tool Bar
  - 보기 도구모음



100% (Combo) Zoom	현재 다이어그램의 확대/축소 비율을 선택합니다.
 Zoom In	현재 다이어그램을 확대해서 나타냅니다.
 Zoom Out	현재 다이어그램을 축소해서 나타냅니다.
 Fit To Window	현재 다이어그램 전체를 모두 볼 수 있도록 확대/축소 비율을 자동으로 설정합니다.
 Refresh[F5]	현재 다이어그램을 다시 그립니다.

# 3. 도구 기능 소개

## 3.4 공통 기능 (4/4)

- StarUML 공통 Tool Bar
  - 정렬 도구모음



	Layout Diagram	현재 다이어그램의 뷰 요소들을 자동으로 배치시킵니다.
	Bring to Front	선택된 요소들을 맨 앞에 나오도록 합니다.
	Send to Back	선택된 요소들을 맨 뒤에 나오도록 합니다.
	Align Left	선택된 요소들을 왼쪽에 맞춰 정렬합니다.
	Align Right	선택된 요소들을 오른쪽에 맞춰 정렬합니다.
	Align Middle	선택된 요소들을 가로로 가운데 정렬합니다.
	Align Top	선택된 요소들을 위쪽에 맞춰 정렬합니다.
	Align Bottom	선택된 요소들을 아래쪽에 맞춰 정렬합니다.
	Align Center	선택된 요소들을 세로로 가운데 정렬합니다.
	Space Equally, Horizontally	선택된 요소들 간의 가로 간격을 균등하게 맞춥니다.
	Space Equally, Vertically	선택된 요소들 간의 세로 간격을 균등하게 맞춥니다.

## 3. 도구 기능 소개

### 3.5 예제 소개

- Violet UML Editor의 기능을 소개 하기 위해 수강신청 시스템 구축을 예제로 하겠습니다.
  - 수강신청 요구사항 중 일부입니다.

#### ▶ 요구사항

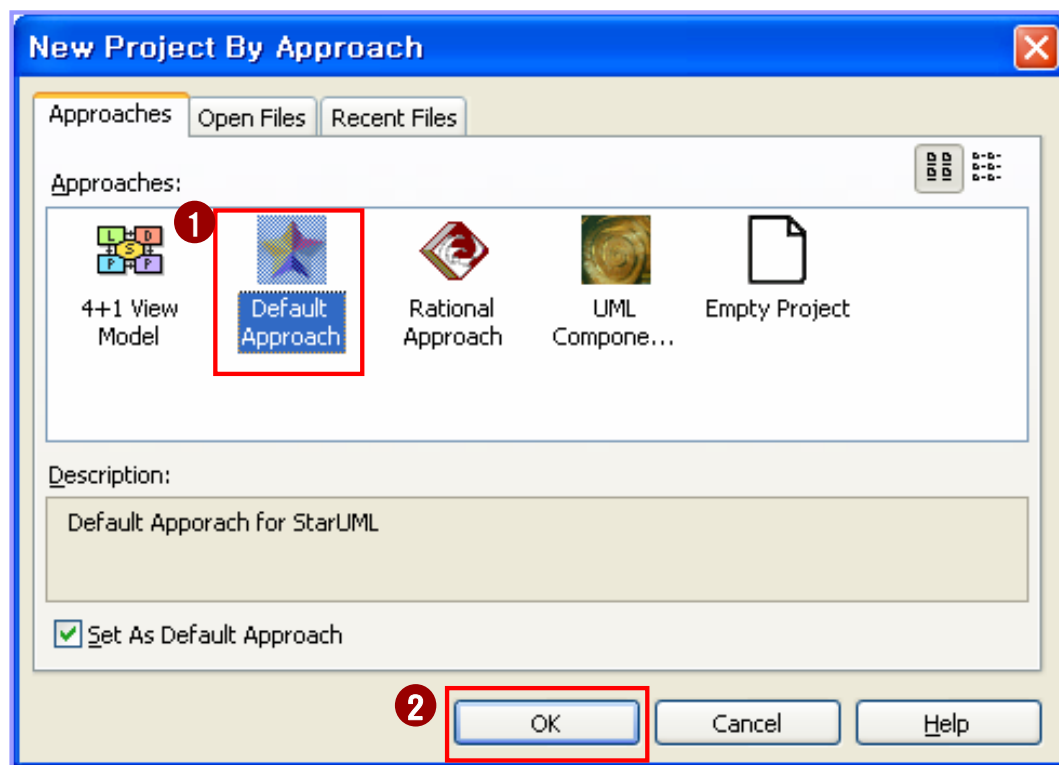
이 시스템은 수강신청페이지에 접속하여 수강신청을 할 수 있도록 도와주는 것이다. 시스템에 등록된 학교 학생이 수강신청 페이지에 로그인 후 과목을 선택하고, 수업을 선택하여 잔여좌석을 확인한다. 만약, 잔여좌석이 없을 경우 과목선택부터 다시 하고 잔여좌석이 있을 경우 수강신청을 완료 하게 된다.

- 수강신청 시스템의 구현 환경은 다음과 같습니다.
  - 운영체제 환경
    - » 사용자는 PC 기반의 Windows XP를 운영체제로 사용하고 있습니다.
    - » Web Server, DB Server, Web Application Server는 Windows NT 기반의 운영체제를 사용하고 있습니다.
  - 네트워크 환경
    - » 사용자와 Web Server간의 통신은 HTTP 방식을 사용하고 있습니다.
    - » 사용자와 Web Application Server간의 통신은 TCP/IP 방식을 사용하고 있습니다.
    - » Web Server와 DB Server, DB Server와 Web Application Server간의 통신은 TCP/IP 방식을 사용하고 있습니다.

## 3. 도구 기능 소개

### 3.6 StarUML 사용하기 (1/4)

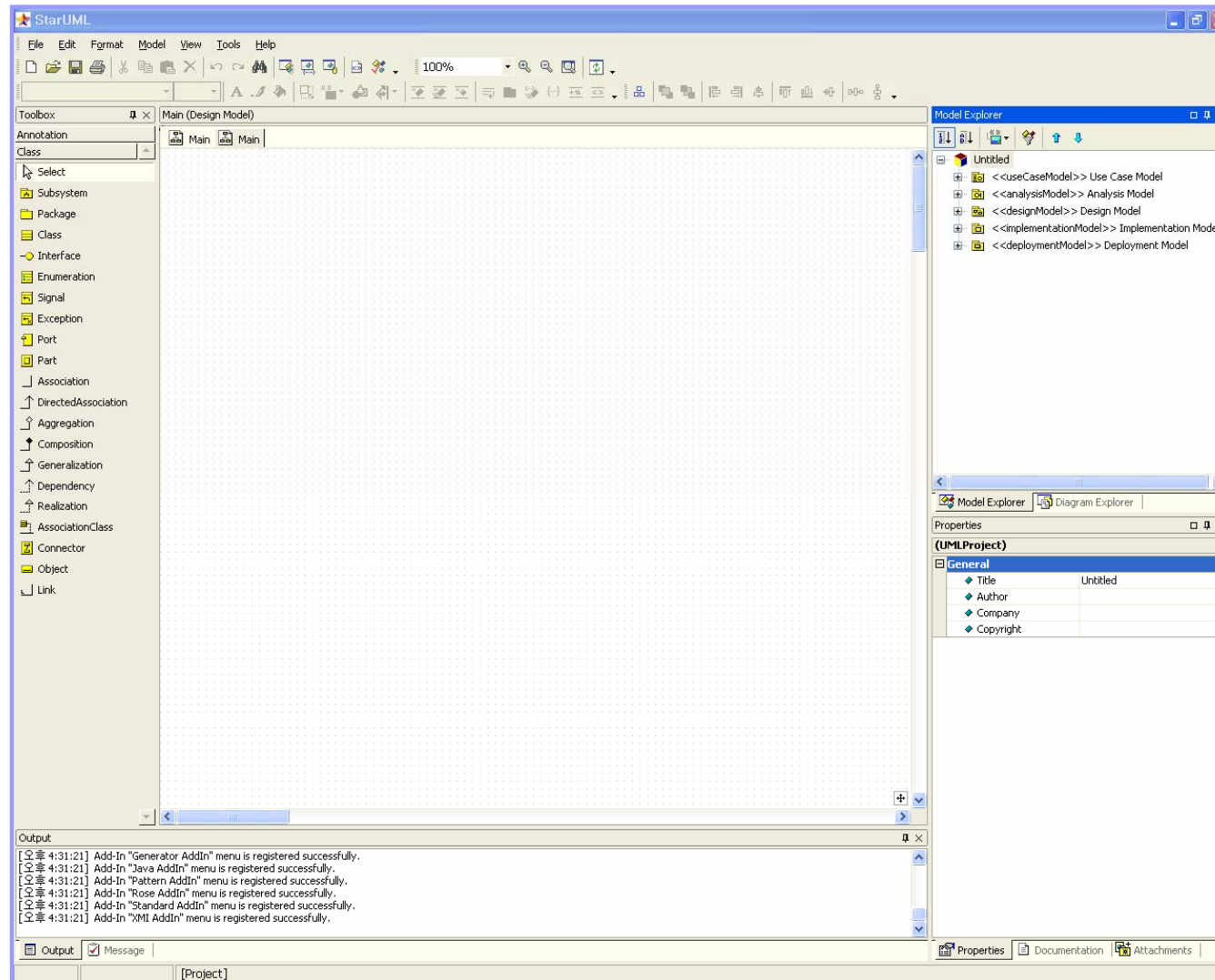
- StarUML 이용하여 UML을 작성하도록 하겠습니다.
  - StarUML을 실행하고 새 Project를 생성합니다.
    - 본 예제에서는 Project name을 Test로 하겠습니다.
  - Approach는 'Default Approach'를 선택합니다.



# 3. 도구 기능 소개

## 3.6 StarUML 사용하기 [2/4]

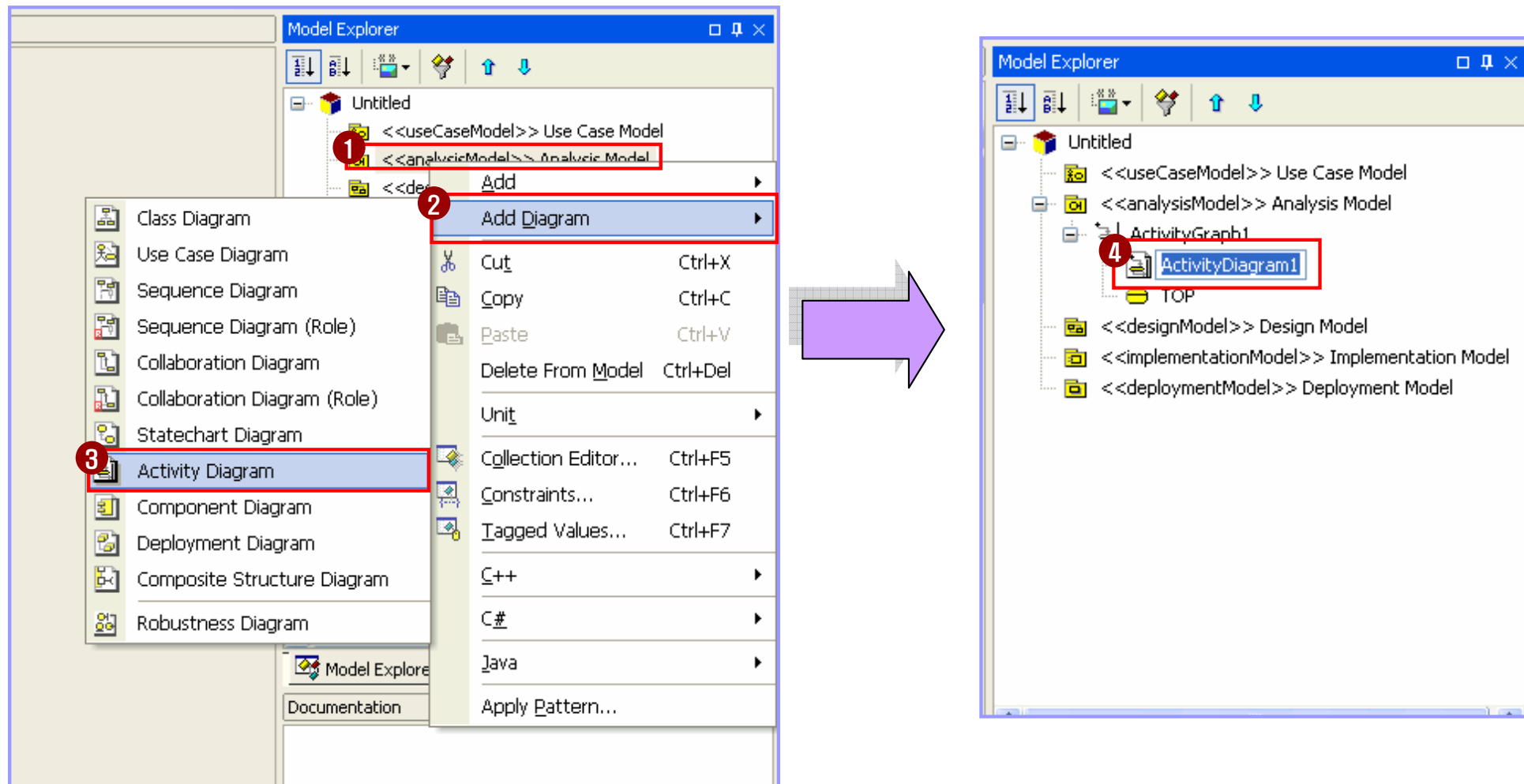
- StarUML이 'Default Approach'로 실행된 것을 확인할 수 있습니다.
  - 기본적으로 5개의 모델에 대하여 UML을 생성할 수 있습니다.



### 3. 도구 기능 소개

#### 3.6 StarUML 사용하기 (3/4)

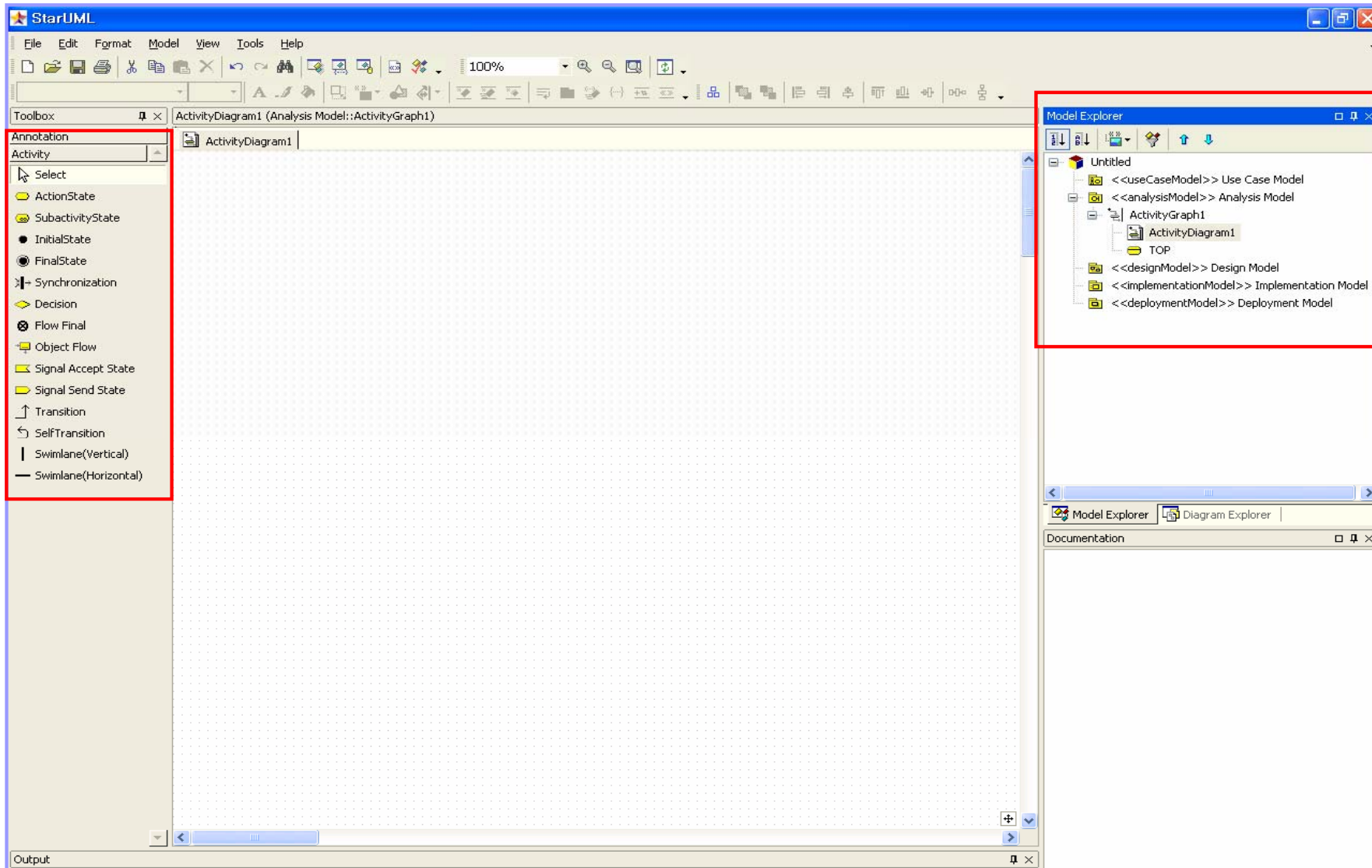
- 각 Diagram 생성방법 입니다.
  - Model Explorer의 각 Diagram에 적합한 Model 선정 -> 오른쪽 버튼 클릭 -> Add Diagram
  - > 원하는 Diagram 선택 -> 이름 변경



### 3. 도구 기능 소개

#### 3.6 StarUML 사용하기 (4/4)

- 선택한 Diagram이 Model Explorer에 생성되고, Tool Box와 Main Window가 Diagram의 종류에 맞게 생성되었습니다.

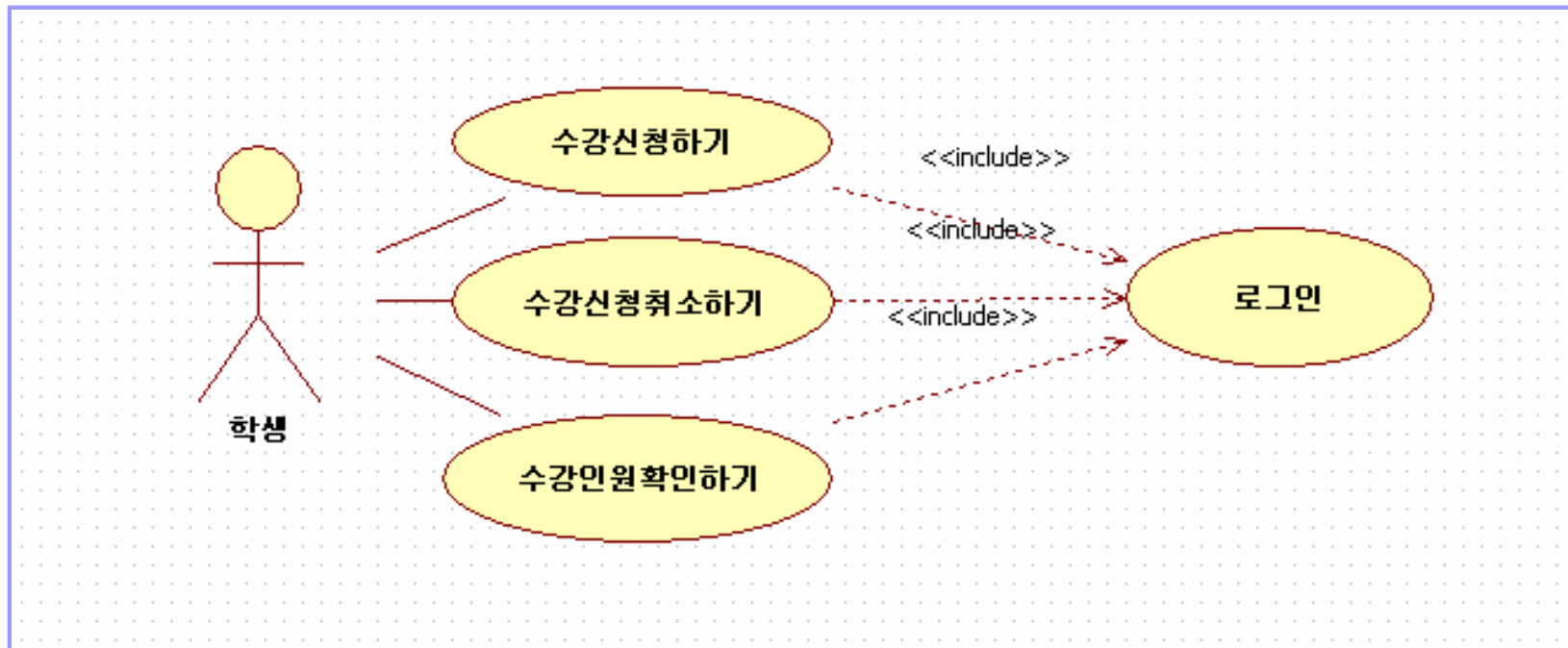


### 3. 도구 기능 소개

#### 3.7 Use Case Diagram 그리기(1/8)

- 31p의 수강신청 시스템의 요구사항을 Use Case Diagram으로 표현해 보겠습니다.

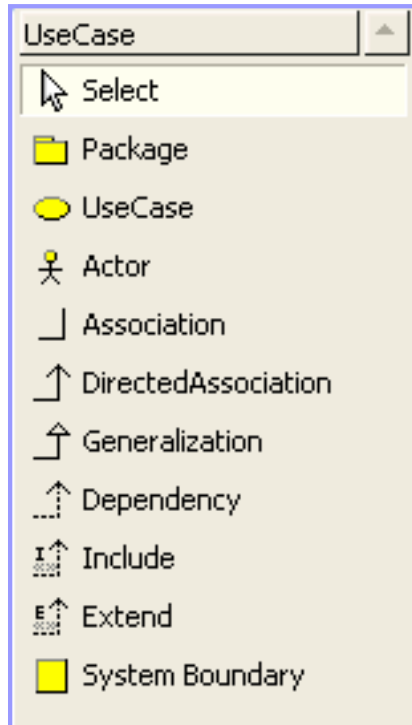
대기상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다. 로그인을 하고 과목목록을 확인한 뒤 과목을 선택한다. 선택한 과목의 수업을 선택하면 수강신청이 완료된다.



### 3. 도구 기능 소개

#### 3.7 Use Case Diagram 그리기 (2/8)

- Use Case Diagram Tool Bar

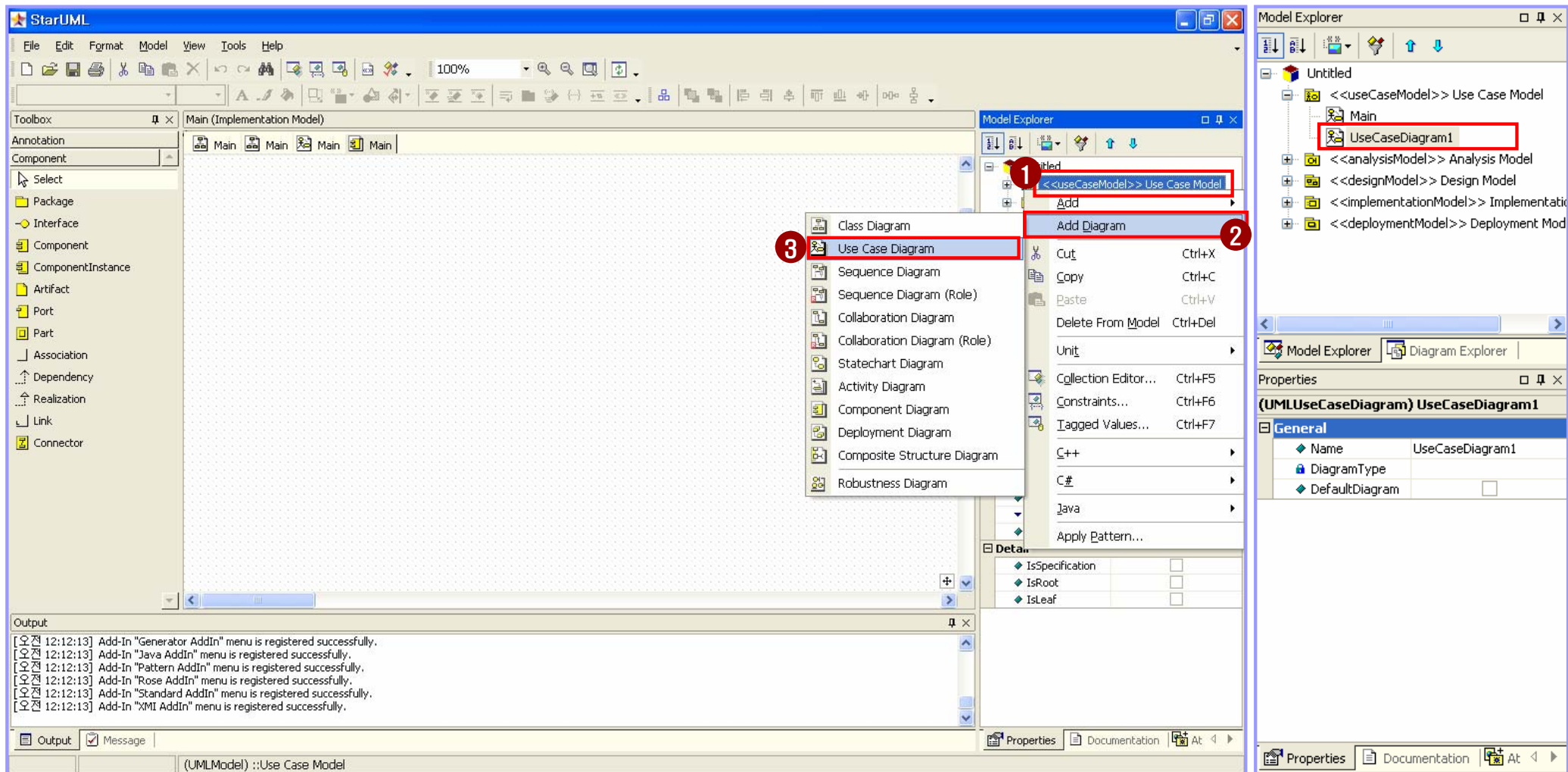


Select	Diagram 요소를 선택합니다.
Package	모델 요소들을 논리적으로 그룹화 할 때 사용합니다.
UseCase	시스템이 제공하는 기능을 의미합니다.
Actor	시스템을 사용하는 사용자나 외부 시스템을 의미합니다.
Association	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다.
DirectedAssociation	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다. 종속, 포함등의 관계를 표시할 수 있습니다.
Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용합니다.
Dependency	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.

# 3. 도구 기능 소개

## 3.7 Use Case Diagram 그리기 (3/8)

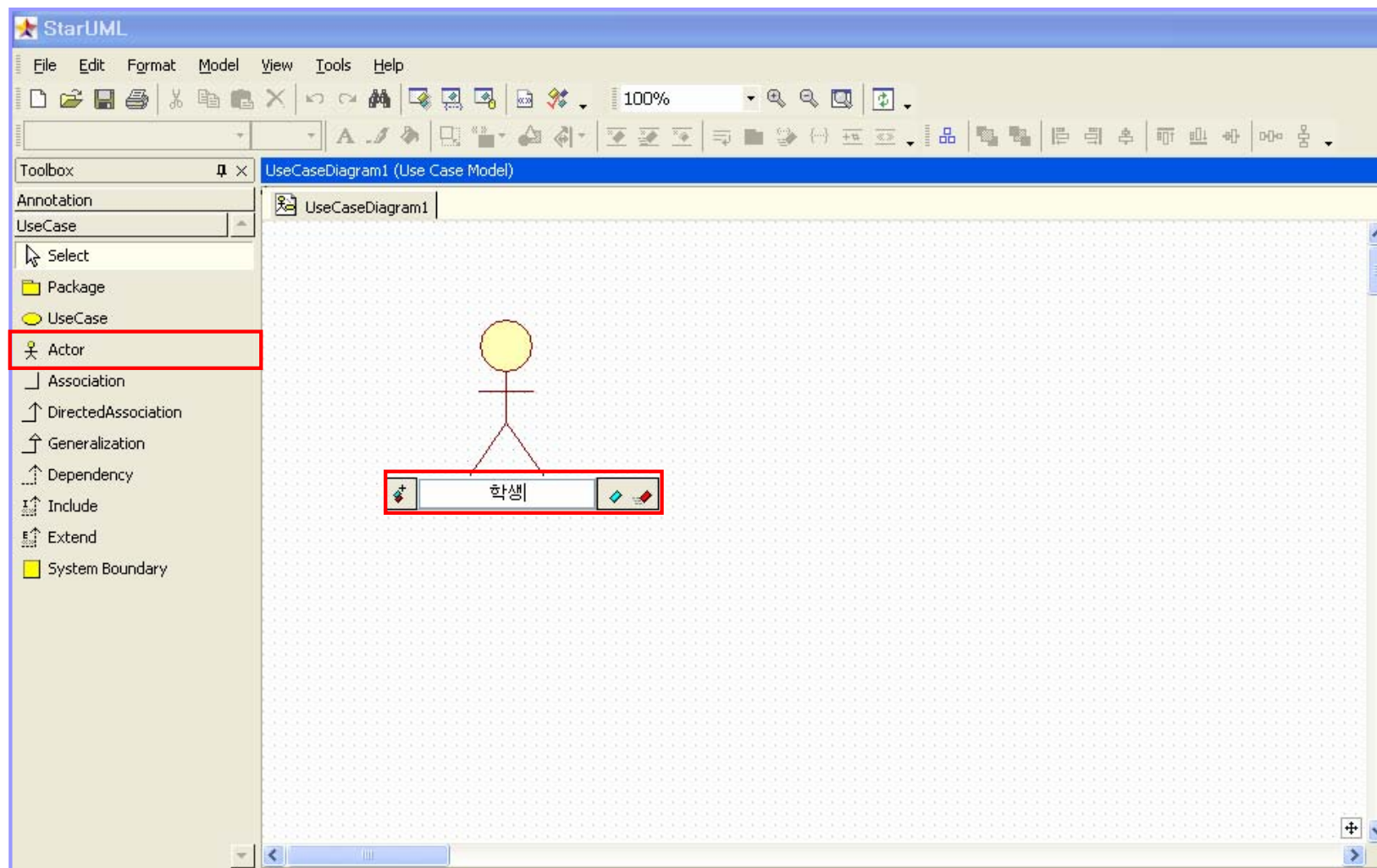
- Use Case Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 <<usecaseModel>> Use Case Model 선택 -> 마우스 오른쪽 버튼 클릭 -> Add Diagram -> Use Case Diagram 선택



## 3. 도구 기능 소개

### 3.7 Use Case Diagram 그리기 (4/8)

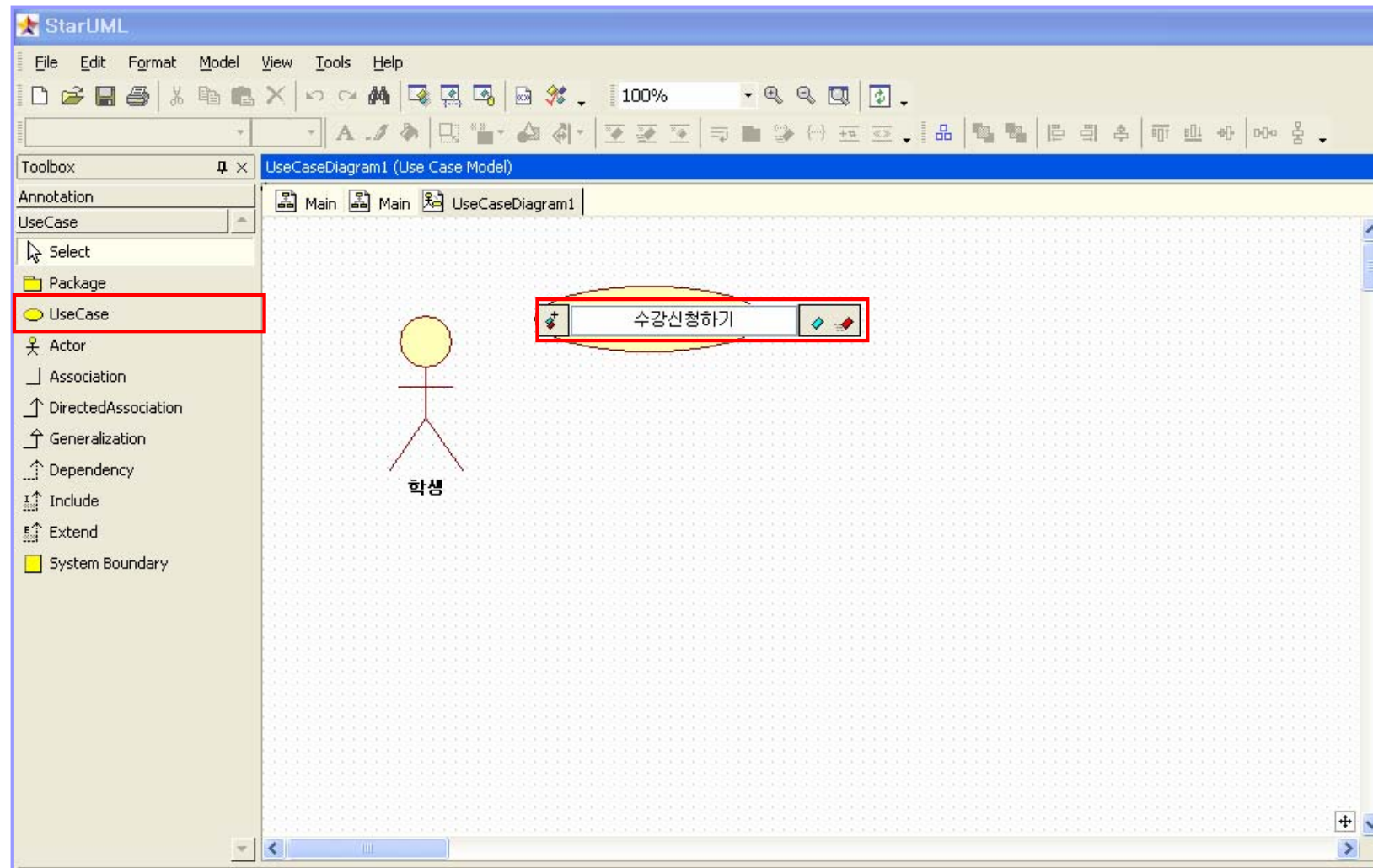
- Actor를 식별합니다.
  - Toolbox의 UseCase에서 Actor 아이콘 클릭 -> 마우스로 Actor 크기 지정 -> Actor 이름 입력 -> Enter



# 3. 도구 기능 소개

## 3.7 Use Case Diagram 그리기 (5/8)

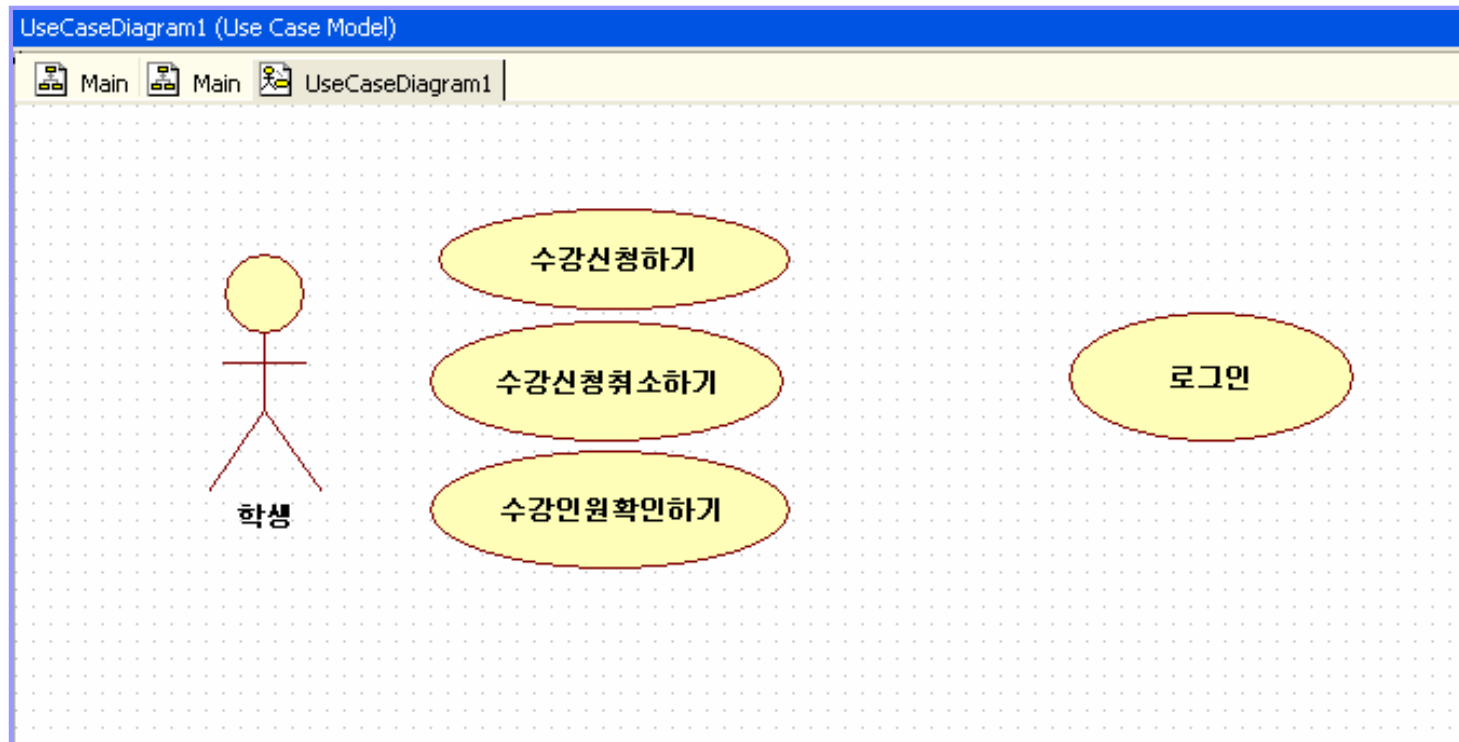
- Use case를 식별합니다.
  - Toolbox의 UseCase에서 UseCase 아이콘 클릭 -> 마우스로 Use Case 크기 지정 -> Use Case 이름 입력 -> Enter



### 3. 도구 기능 소개

#### 3.7 Use Case Diagram 그리기 (6/8)

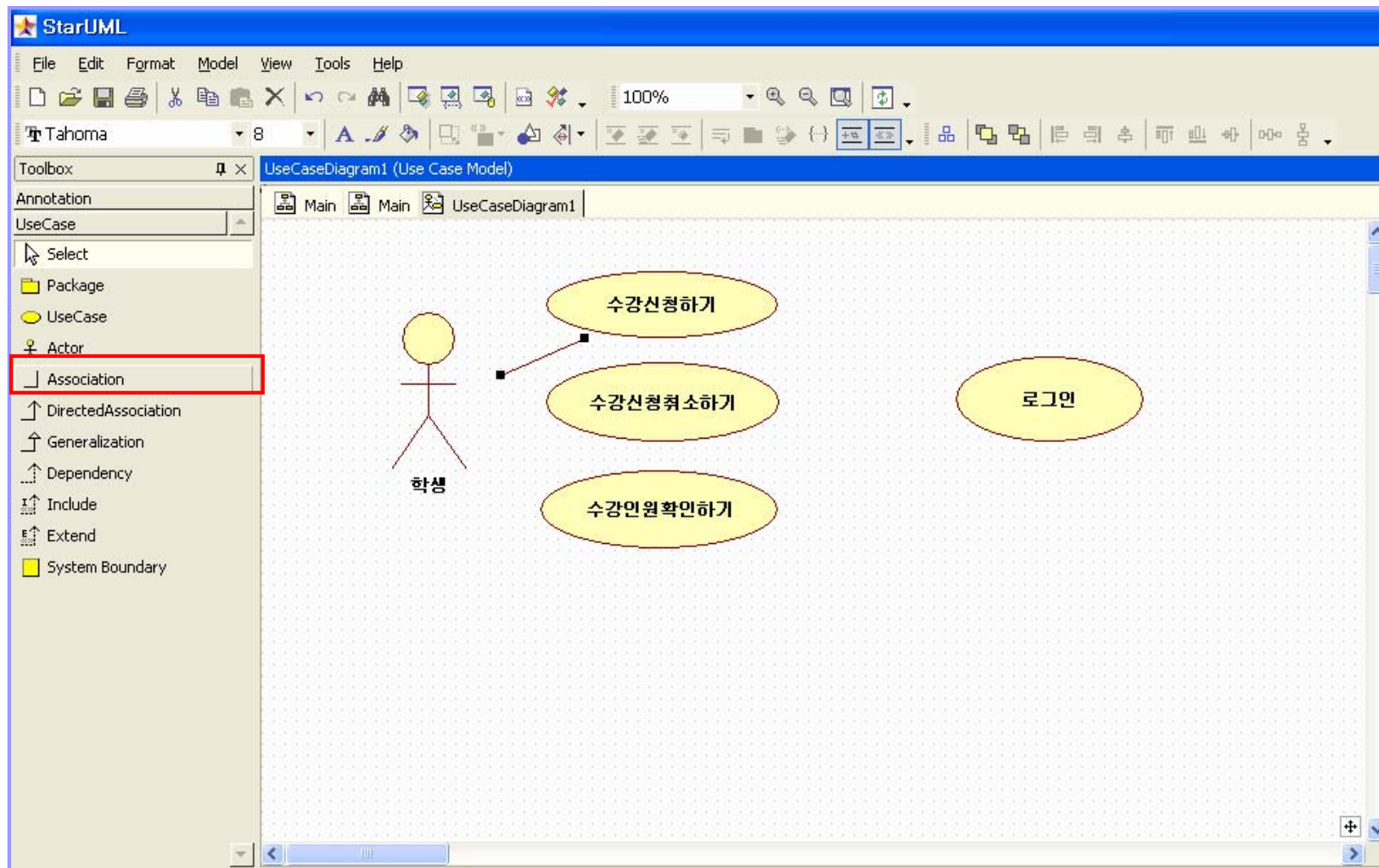
- Use case 작성화면 입니다.
  - 수강신청하기, 수강신청취소하기, 수강인원확인하기, 로그인 Use case



## 3. 도구 기능 소개

### 3.7 Use Case Diagram 그리기 (7/8)

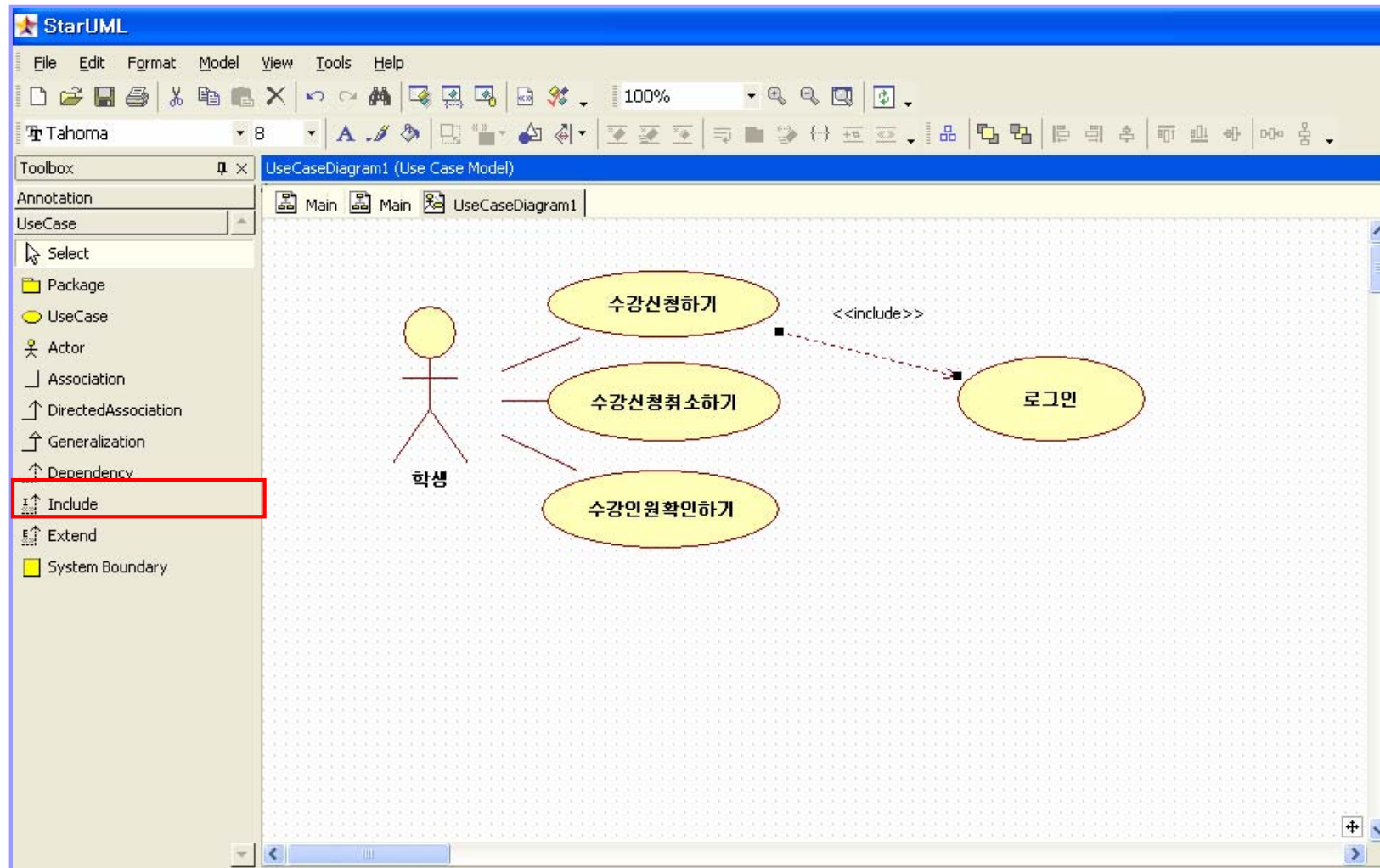
- 연관 관계를 표시합니다.
  - Toolbox의 UseCase에서 Association 클릭 -> Actor와 Use Case 마우스로 연결



### 3. 도구 기능 소개

#### 3.7 Use Case Diagram 그리기 (8/8)

- Include 관계를 표시 합니다.
  - Toolbox의 UseCase에서 Include 클릭 -> Use Case와 Use Case 마우스로 연결

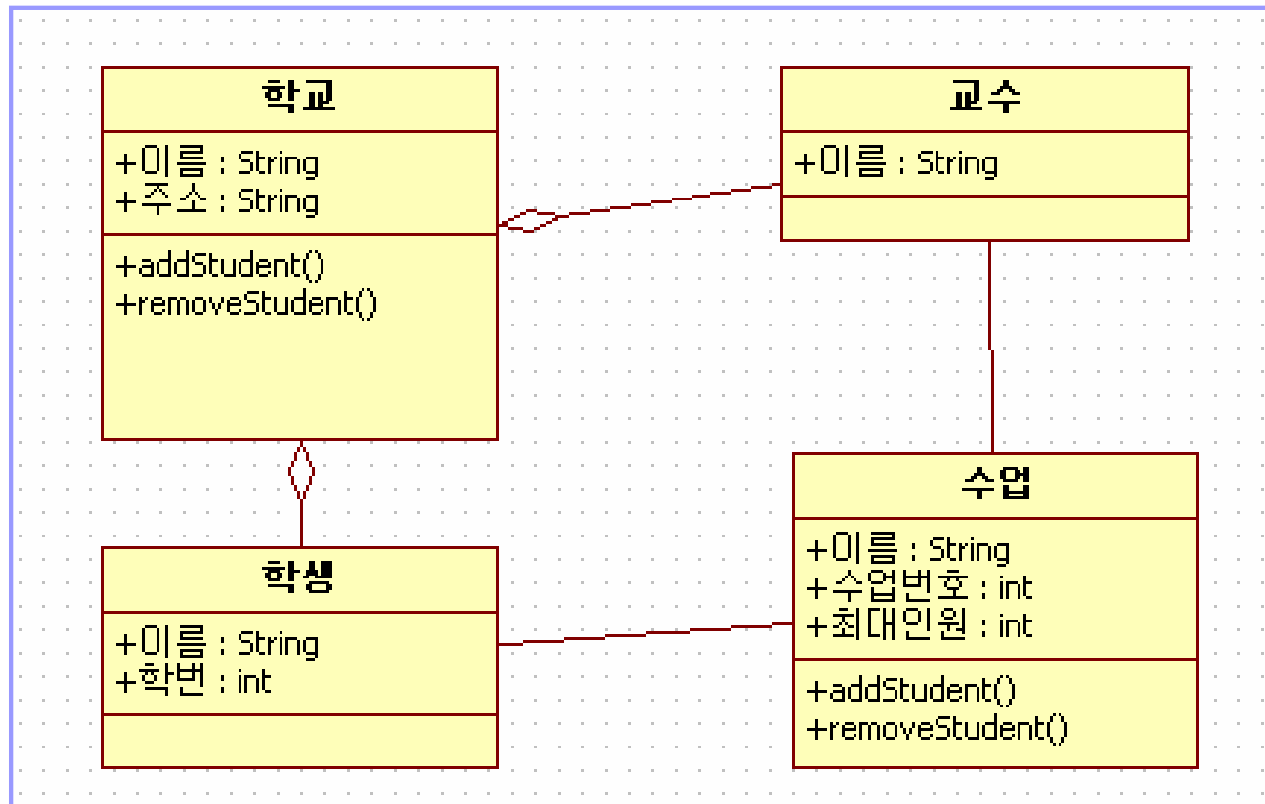


### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(1/10)

- 31p의 수강신청 시스템의 요구사항을 Class Diagram으로 표현해 보겠습니다.

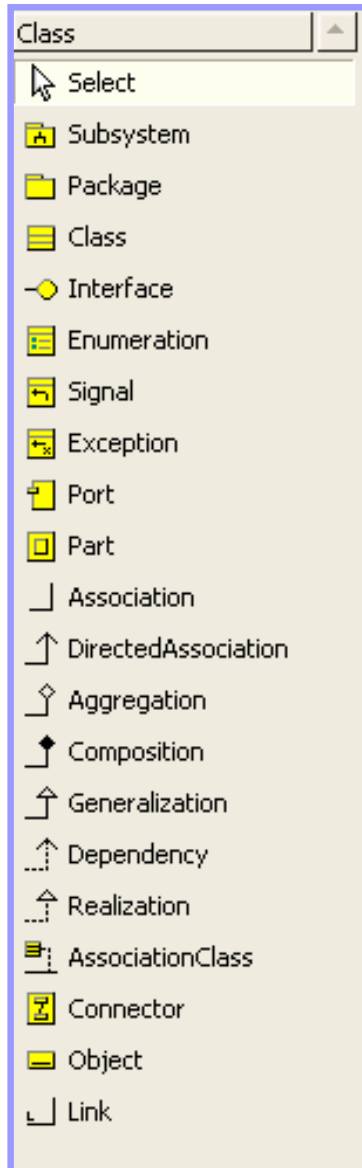
학교는 0명 이상의 학생으로 구성된 집합연관이고 학생은 여러 수업을, 수업은 여러 명의 학생을 수용할 수 있는 다대다 관계이다. 교수 한 명당 0개 이상의 수업을 담당한다.



# 3. 도구 기능 소개

## 3.8 Class Diagram 그리기 (2/10)

### • Class Diagram Tool Bar (1/2)

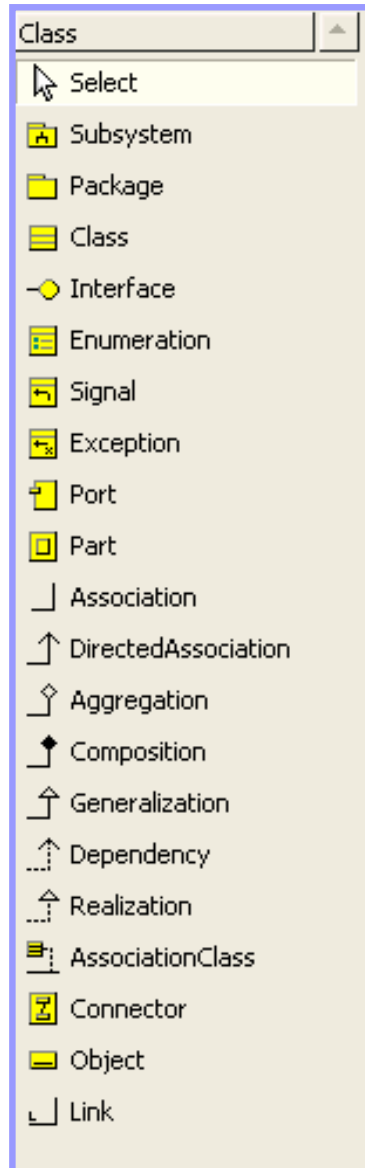


Select	Diagram 요소를 선택합니다.
Subsystem	물리적인 시스템의 부분 또는 전체를 의미합니다.
Package	관련된 클래스들을 모아서 모델링 한 것을 의미합니다.
Class	객체의 Attribute와 Method을를 모델링 한 것을 의미합니다.
Interface	클래스에서 Method의 선언부분만 모델링 한 것을 의미합니다.
Enumeration	미리 정의된 값들을 리스트로 가지는 데이터타입을 의미합니다.
Signal	객체간의 비동기적 통신 신호를 의미합니다.
Exception	실행 오류시 Operation에 의해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

# 3. 도구 기능 소개

## 3.8 Class Diagram 그리기 (3/10)

### • Class Diagram Tool Bar (2/2)

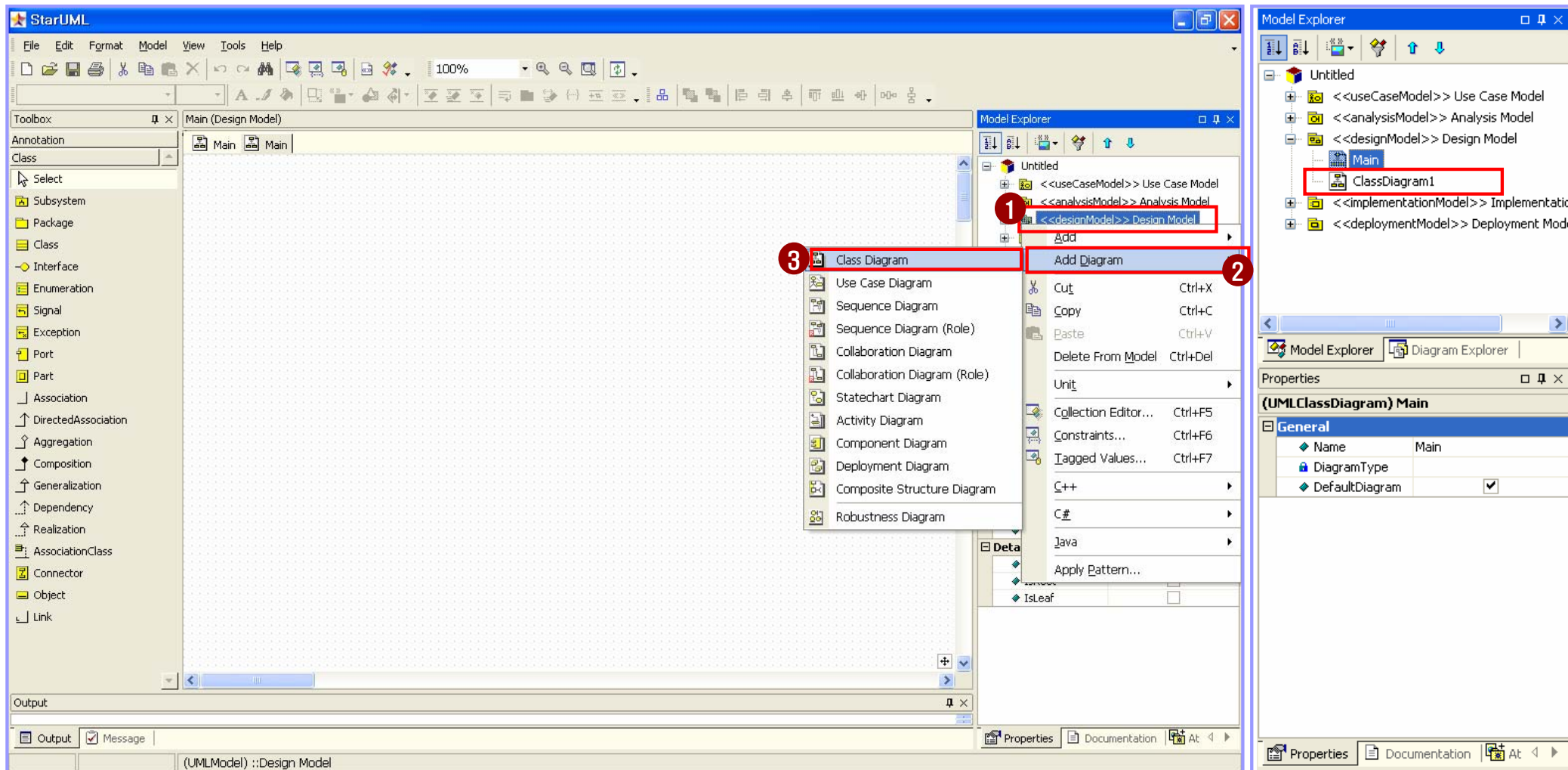


Association	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 가능합니다.
DirectedAssociation	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 불가능합니다.
Aggregation	한 클래스가 다른 클래스를 포함하는 관계일 때 사용합니다.
Composition	한 클래스가 다른 클래스에 완전히 종속되는 관계일 때 사용합니다.
Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용합니다.
Dependency	한 클래스의 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.

# 3. 도구 기능 소개

## 3.8 Class Diagram 그리기(4/10)

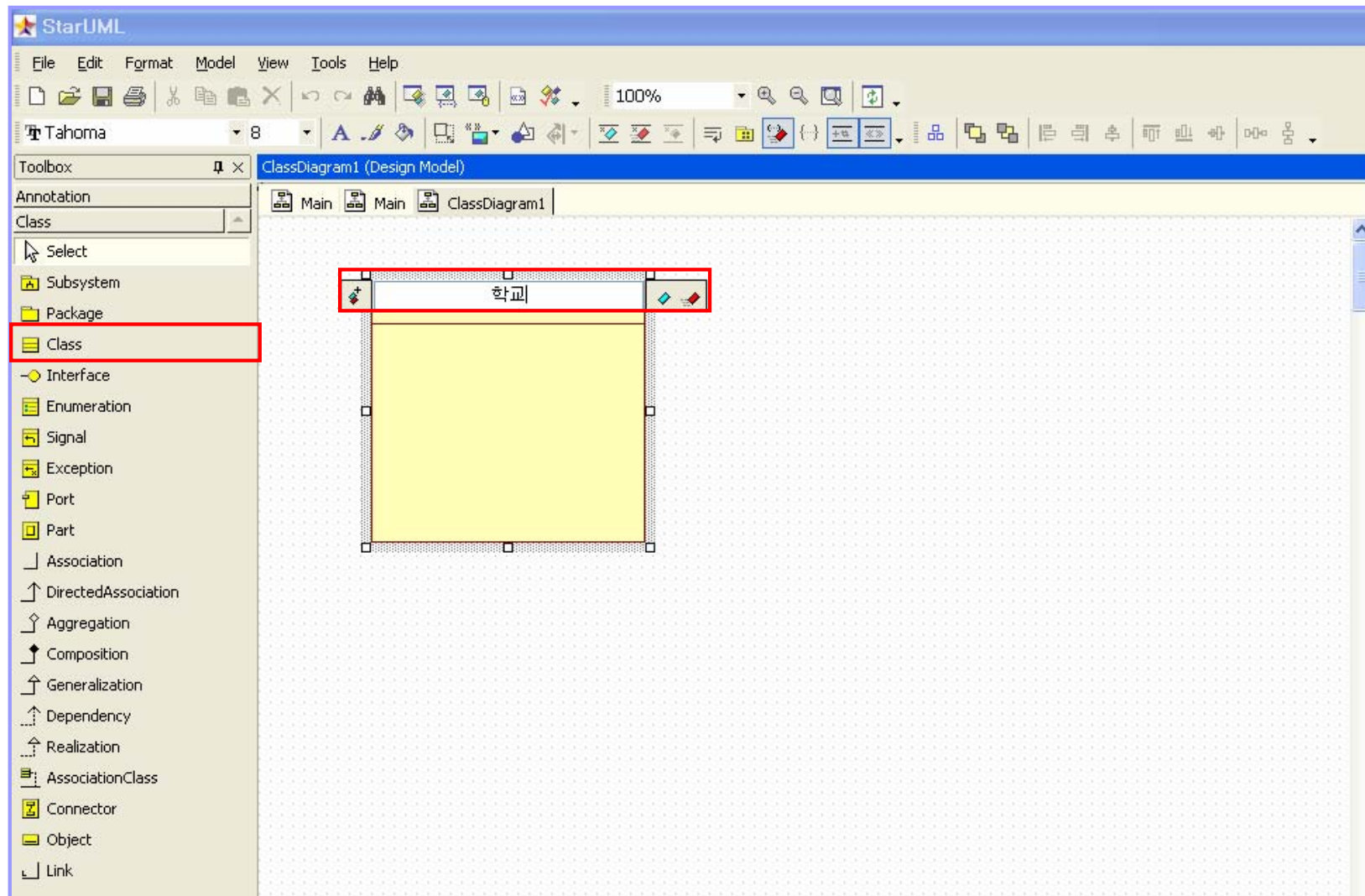
- Use Case Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 <<designModel>> Design Model 선택 -> 마우스 오른쪽 버튼 클릭 -> Add Diagram -> Class Diagram 선택



### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(5/10)

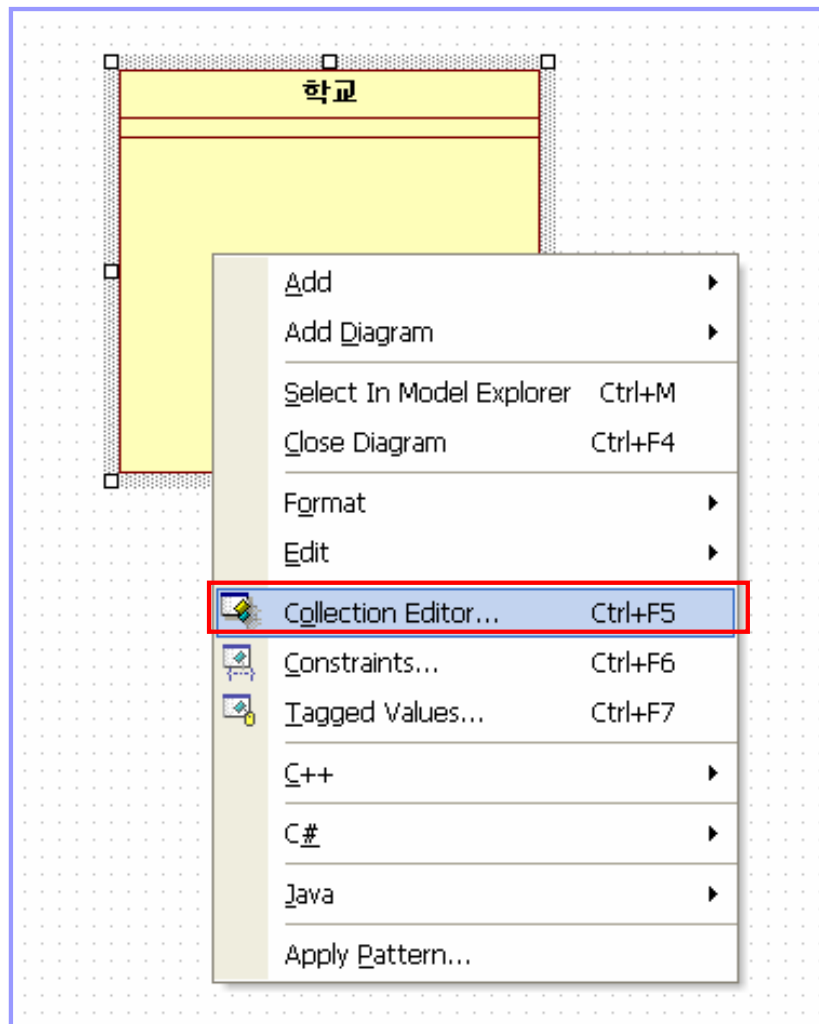
- 시스템을 구성하는 학교 클래스를 표현합니다.
  - Toolbox의 Class 에서 Class 아이콘 클릭 -> 마우스로 Class 크기 지정 -> Class 이름 입력 -> Enter



### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(6/10)

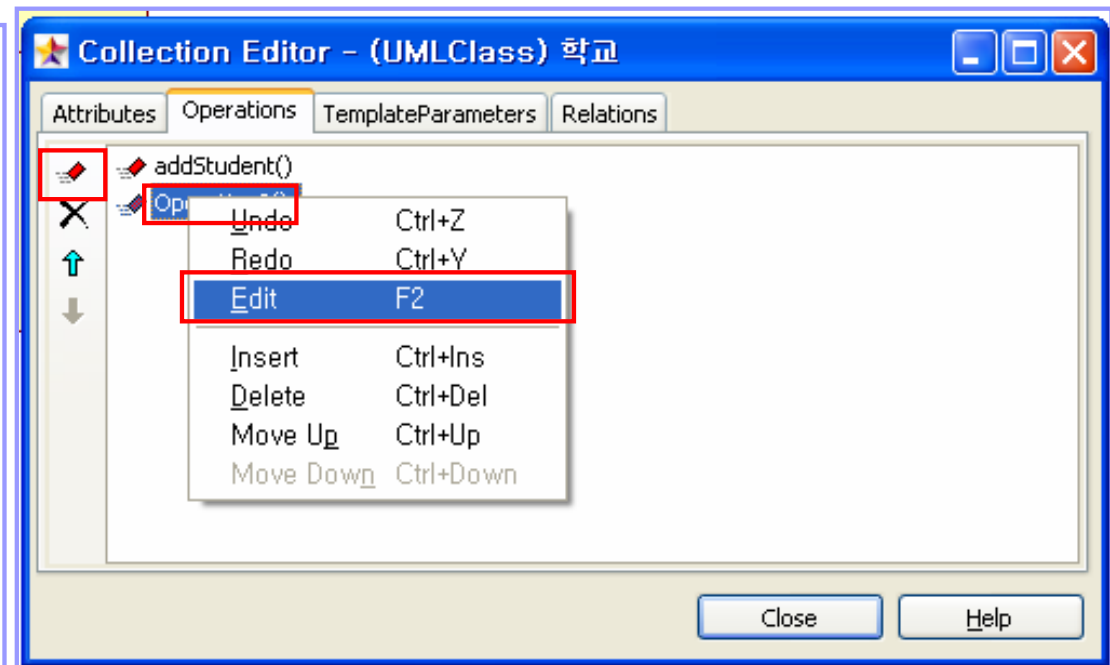
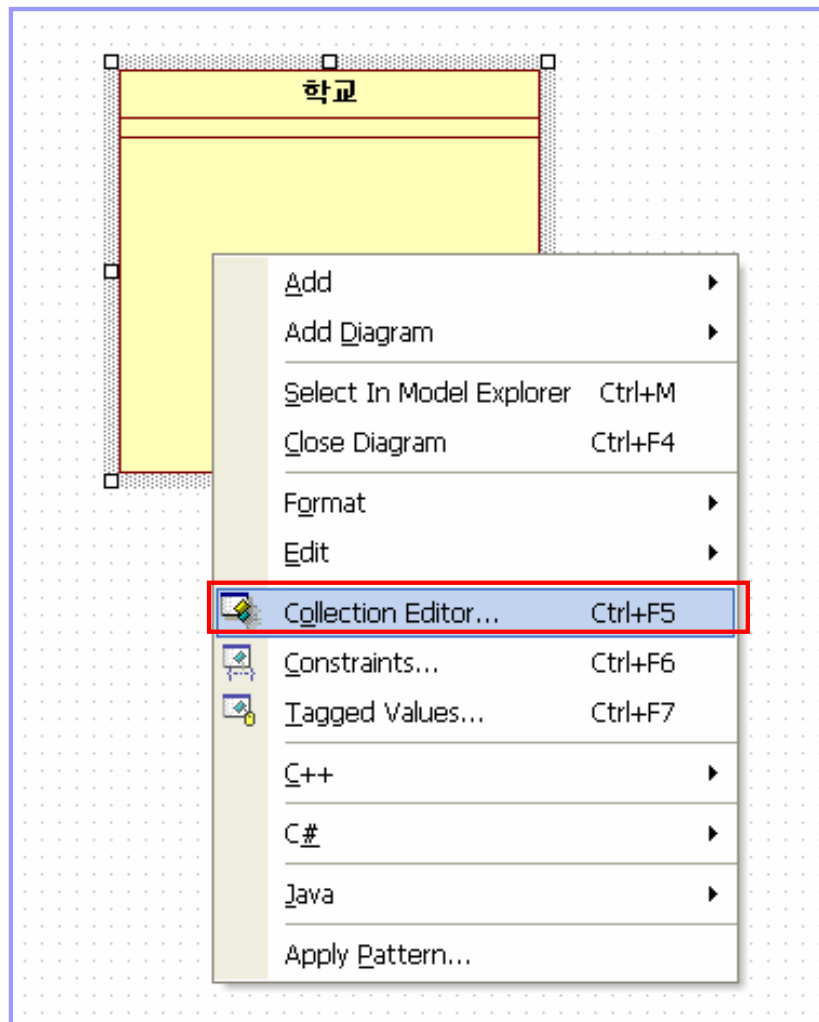
- 학교 Class에 Attributes를 입력합니다.
  - 학교 Class 선택 -> 마우스 오른쪽 버튼 클릭 -> Collection Editor... 클릭
  - Attributes 탭 -> Insert 아이콘 클릭 -> Edit 클릭 -> 마우스 오른쪽 버튼 클릭 -> Name 변경



### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(7/10)

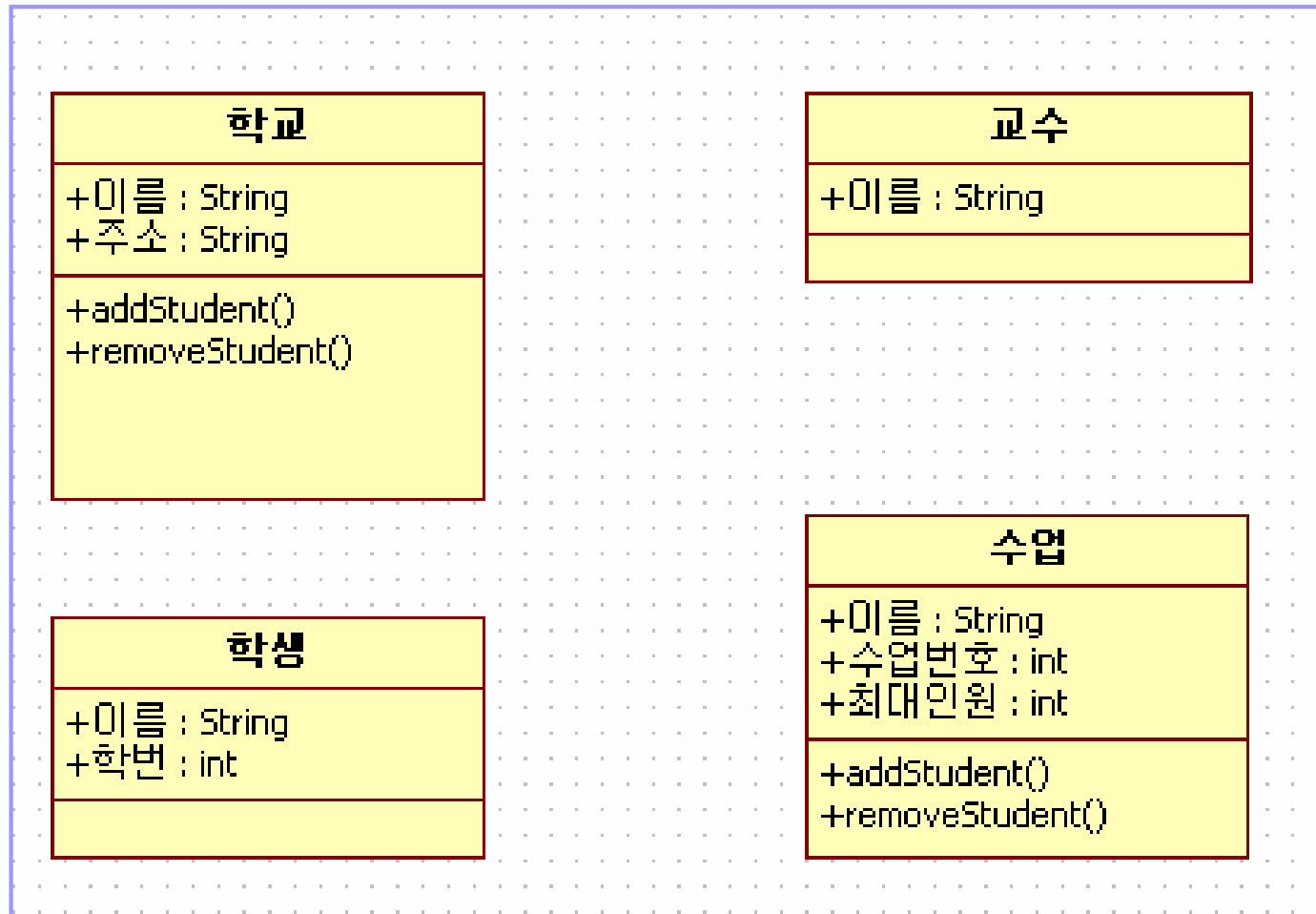
- 학교 Class에 Operations를 입력합니다.
  - 학교 Class 선택 -> 마우스 오른쪽 버튼 클릭 -> Collection Editor... 클릭
  - Operations 탭 -> Insert 아이콘 클릭 -> Edit 클릭 -> 마우스 오른쪽 버튼 클릭 -> Name 변경



### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(8/10)

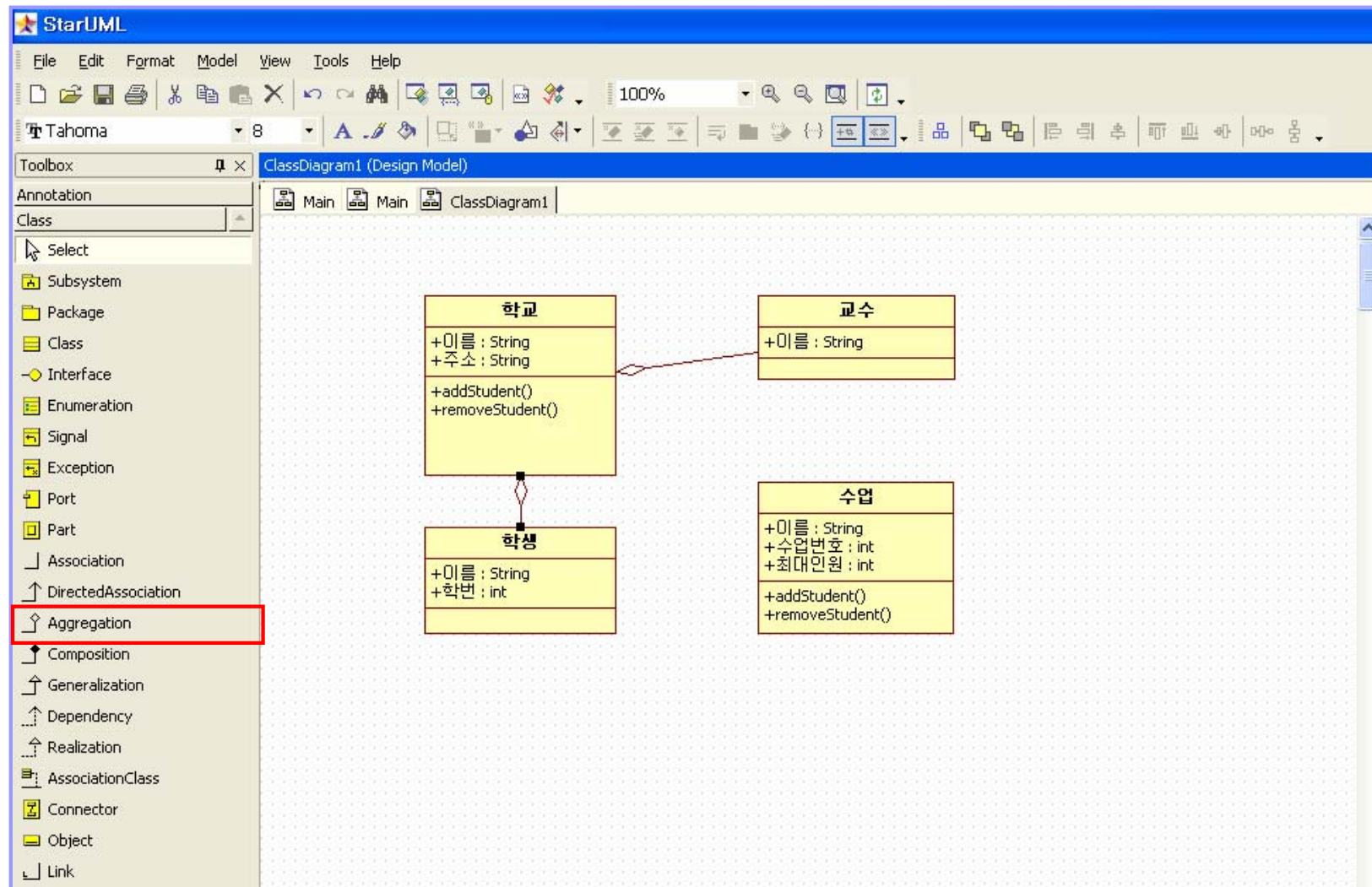
- 학교 Class 생성 방법을 참고하여 교수, 수업, 학생 Class도 다음과 같이 생성합니다.



# 3. 도구 기능 소개

## 3.8 Class Diagram 그리기(9/10)

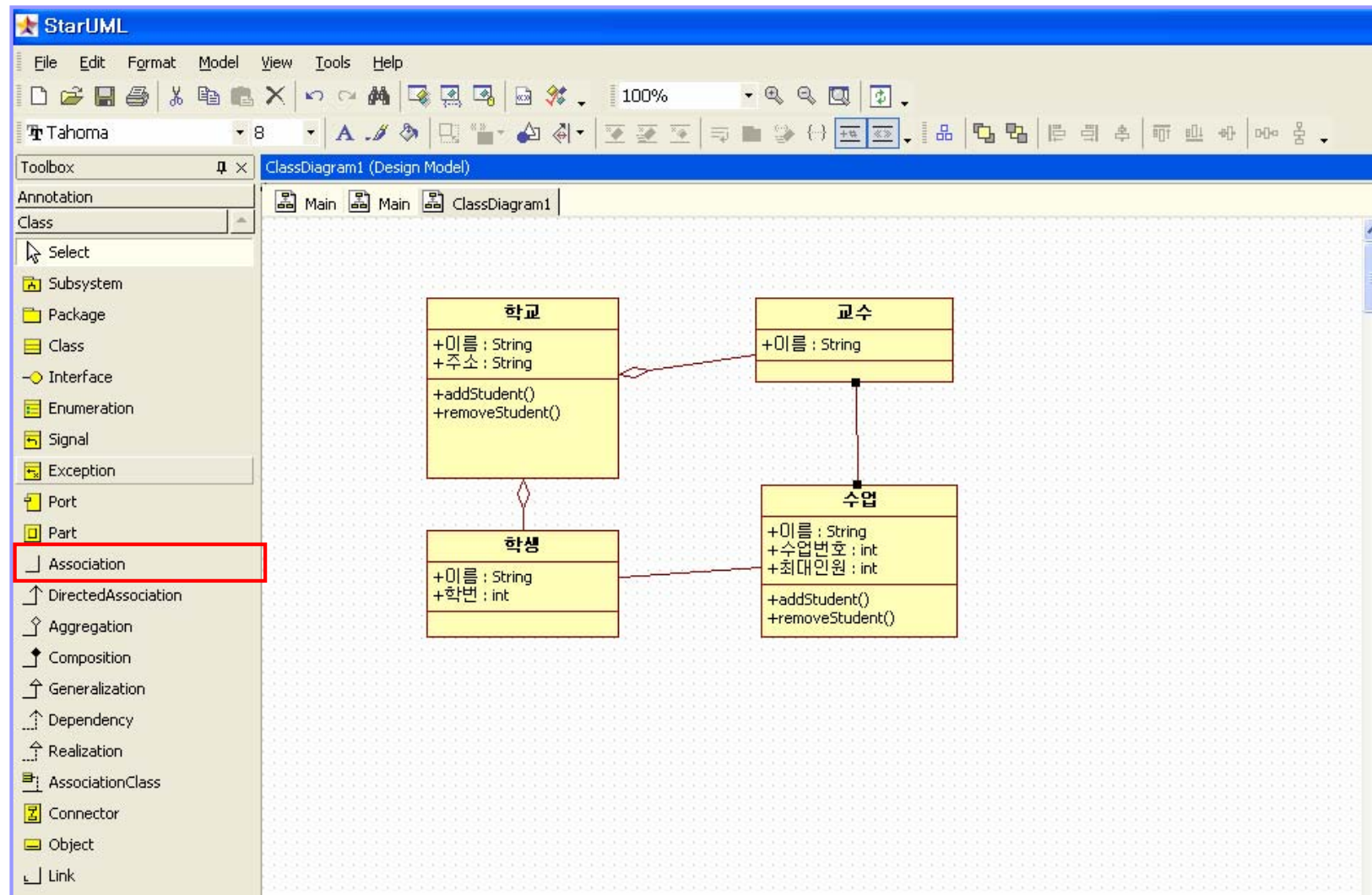
- 클래스간의 관계를 표현해 줍니다.
- 집합연관관계 (한 클래스가 다른 클래스를 포함하는 관계를 의미합니다.)
  - Toolbox의 Class 에서 Aggregation 클릭 -> Class와 Class 마우스로 연결



### 3. 도구 기능 소개

#### 3.8 Class Diagram 그리기(10/10)

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

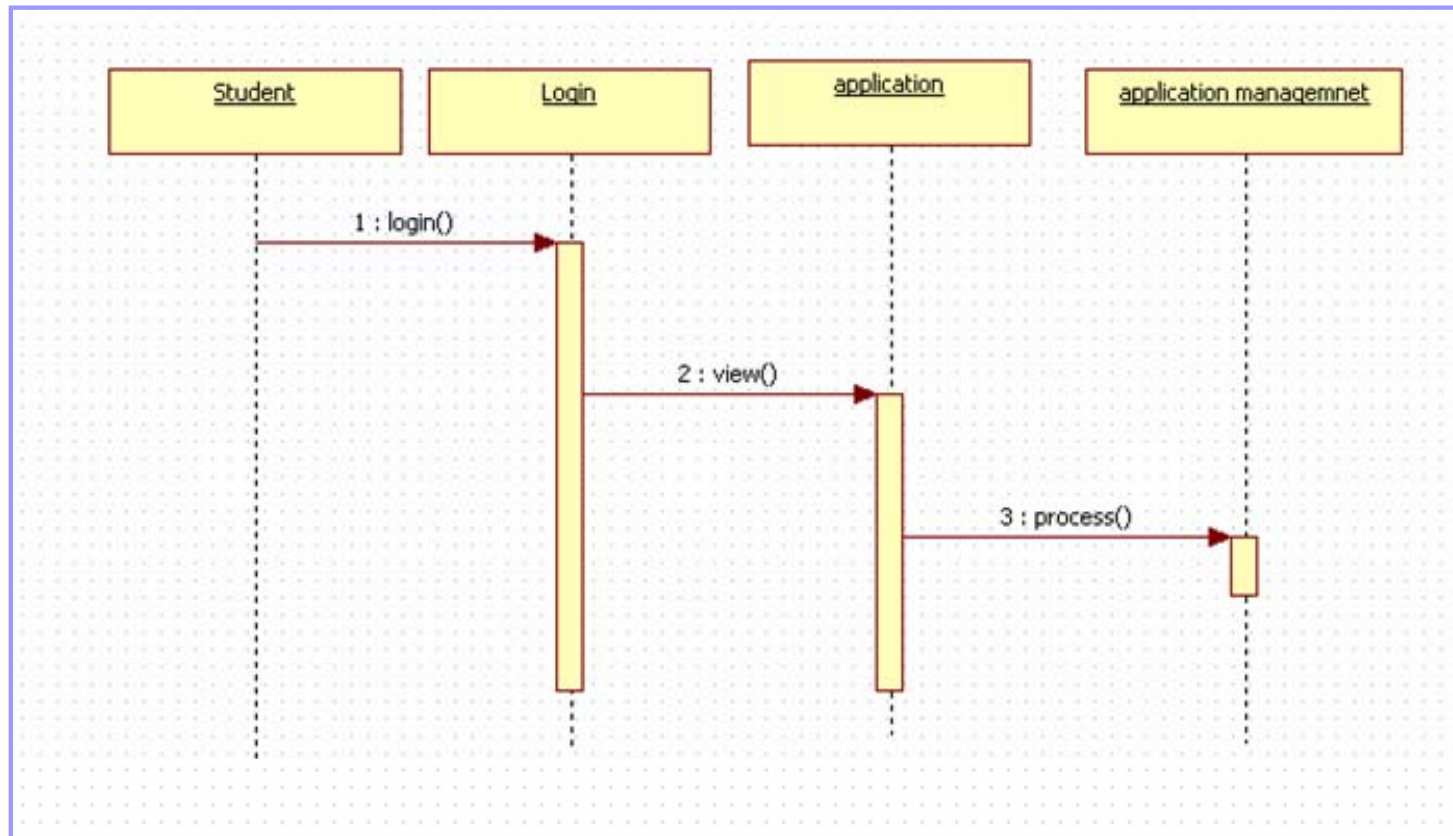


### 3. 도구 기능 소개

#### 3.9 Sequence Diagram 그리기(1/7)

- 31p의 수강신청 시스템의 요구사항을 Sequence Diagram으로 표현해 보겠습니다.

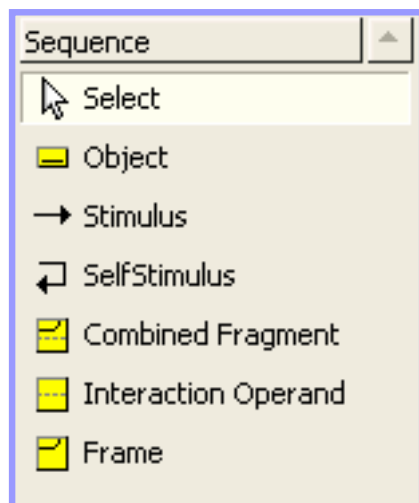
대기상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다. 로그인을 하고 과목목록을 확인한 뒤 과목을 선택한다. 선택한 과목의 수업을 선택하면 수강신청이 완료된다.



### 3. 도구 기능 소개

#### 3.9 Sequence Diagram 그리기 (2/7)

- Sequence Diagram Tool Bar

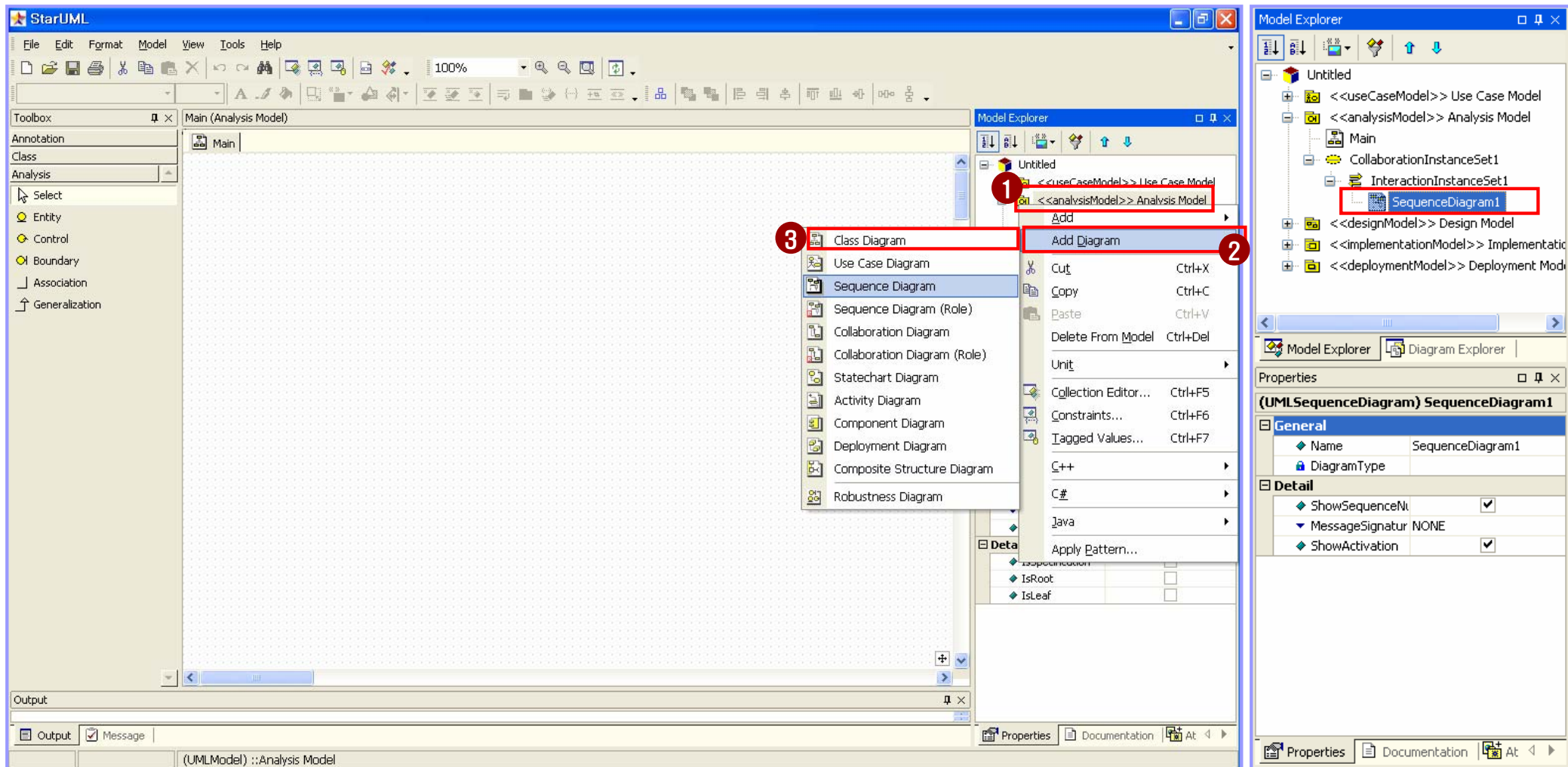


Select	Diagram 요소를 선택합니다.
Object	클래스로부터 생성된 객체를 의미합니다.
Stimulus	두 객체간의 커뮤니케이션을 의미합니다.
SelfStimulus	한 객체가 스스로 커뮤니케이션 함을 의미합니다.
Combined Fragment	Fragment 영역을 의미합니다.
Interaction Operand	조각 내부를 구분할 때 사용합니다.
Frame	시퀀스 다이어그램을 특정 영역으로 구분할 때 사용합니다.

# 3. 도구 기능 소개

## 3.9 Sequence Diagram 그리기(3/7)

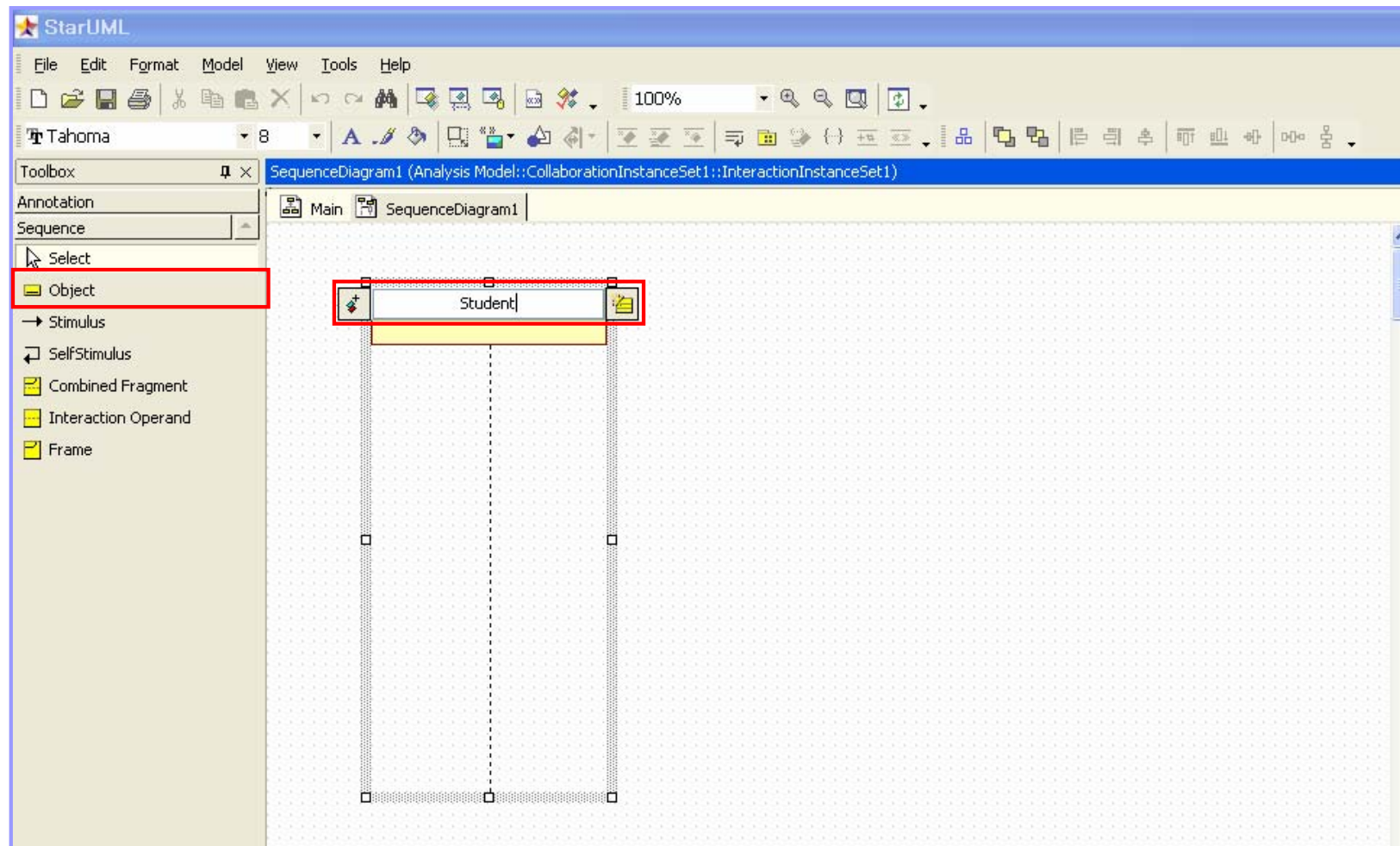
- Sequence Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 <<analysisModel>> Analysis Model 선택 -> 마우스 오른쪽 버튼 클릭 -> Add Diagram -> Sequence Diagram 선택



# 3. 도구 기능 소개

## 3.9 Sequence Diagram 그리기 (4/7)

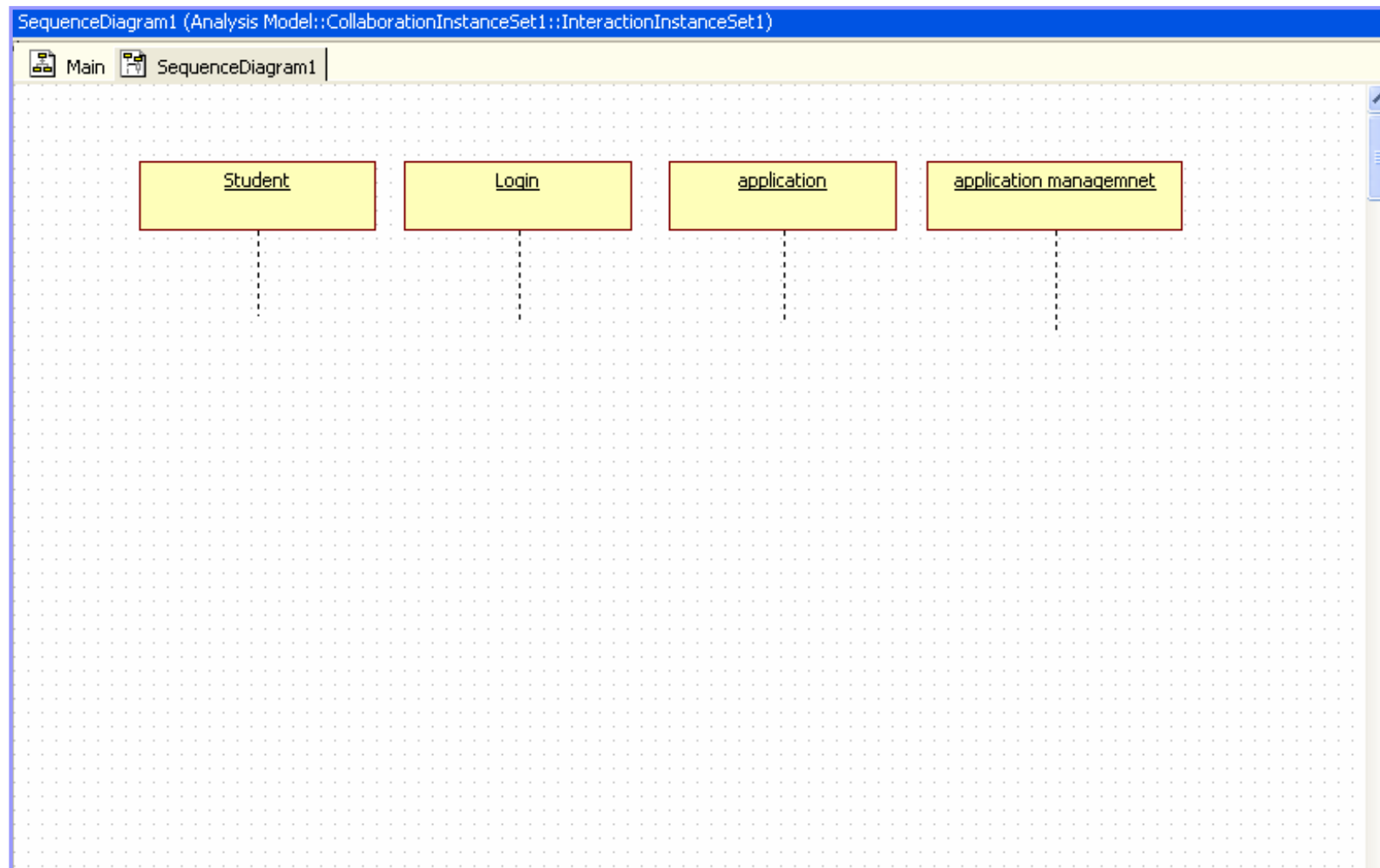
- 상호작용에 참여하는 개체들을 표시합니다.
  - Toolbox의 Sequence 에서 Object 아이콘 클릭 -> 마우스로 Object 크기 지정 -> Object 이름 입력 -> Enter



### 3. 도구 기능 소개

#### 3.9 Sequence Diagram 그리기 (5/7)

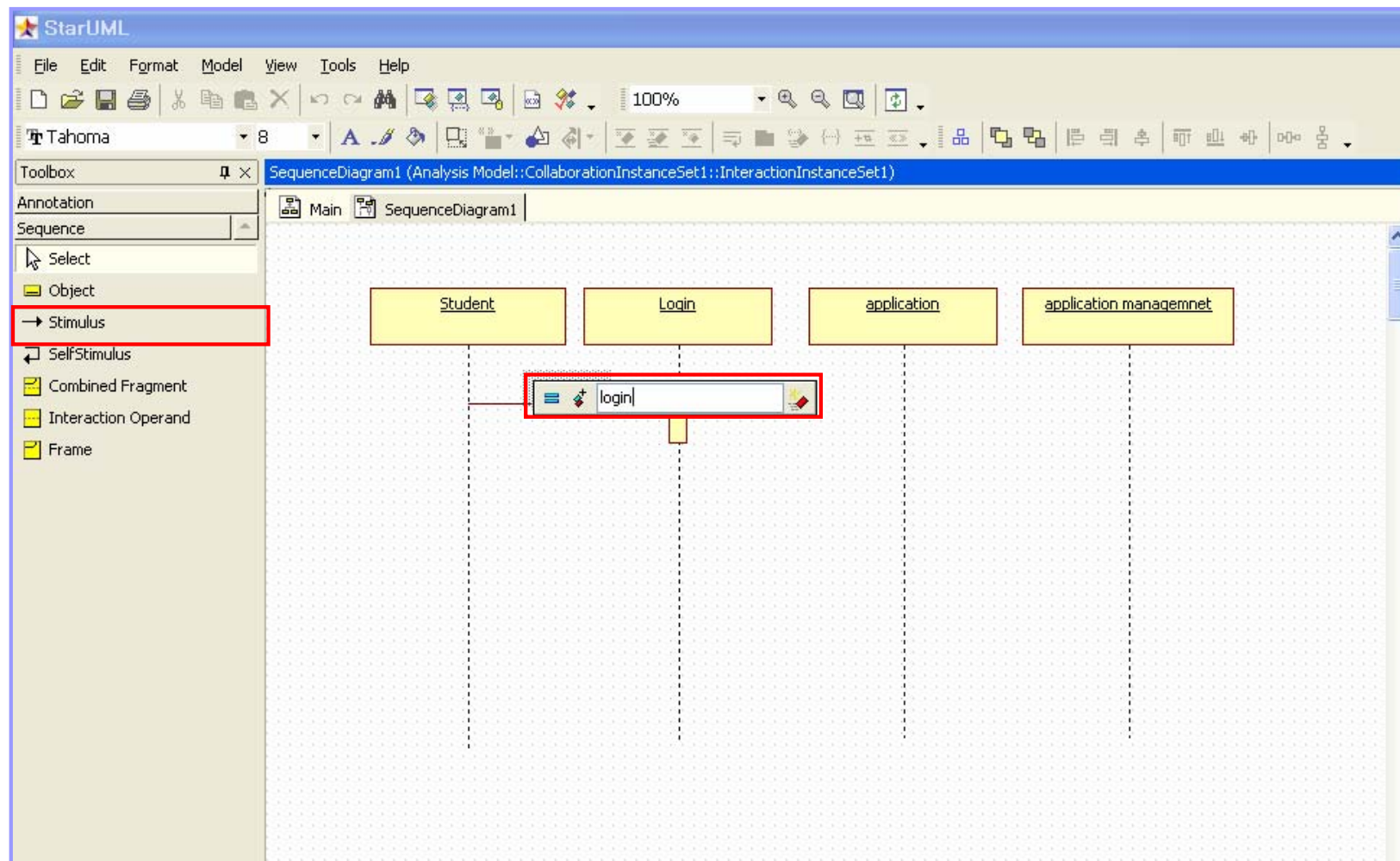
- 상호작용에 참여하는 개체들을 표시합니다.
  - Student Object 생성하는 방법을 참고하여 Login, application, application management 개체들을 표시합니다.



# 3. 도구 기능 소개

## 3.9 Sequence Diagram 그리기 (6/7)

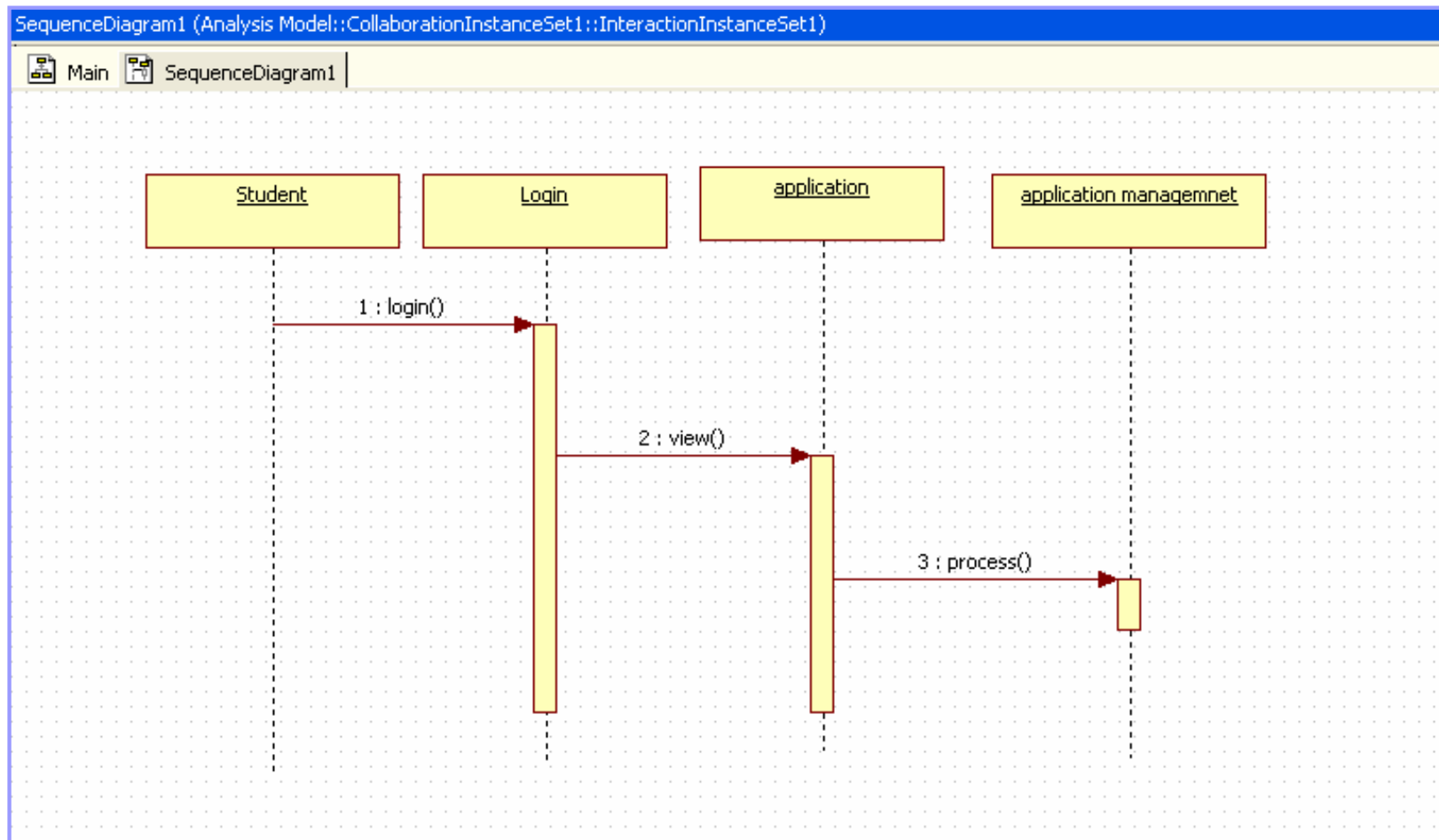
- 다른 객체로의 제어흐름의 이동과 호출 메소드를 표시합니다.
  - Toolbox의 Sequence 에서 Stimulus 아이콘 클릭 -> Object와 Object 선택 -> Stimulus 이름 입력 -> Enter



### 3. 도구 기능 소개

#### 3.9 Sequence Diagram 그리기 (7/7)

- 다른 객체로의 제어흐름의 이동과 호출 메소드를 표시합니다.
  - Toolbox의 Sequence 에서 Stimulus 아이콘 클릭 -> Object와 Object 선택 -> Stimulus 이름 입력 -> Enter

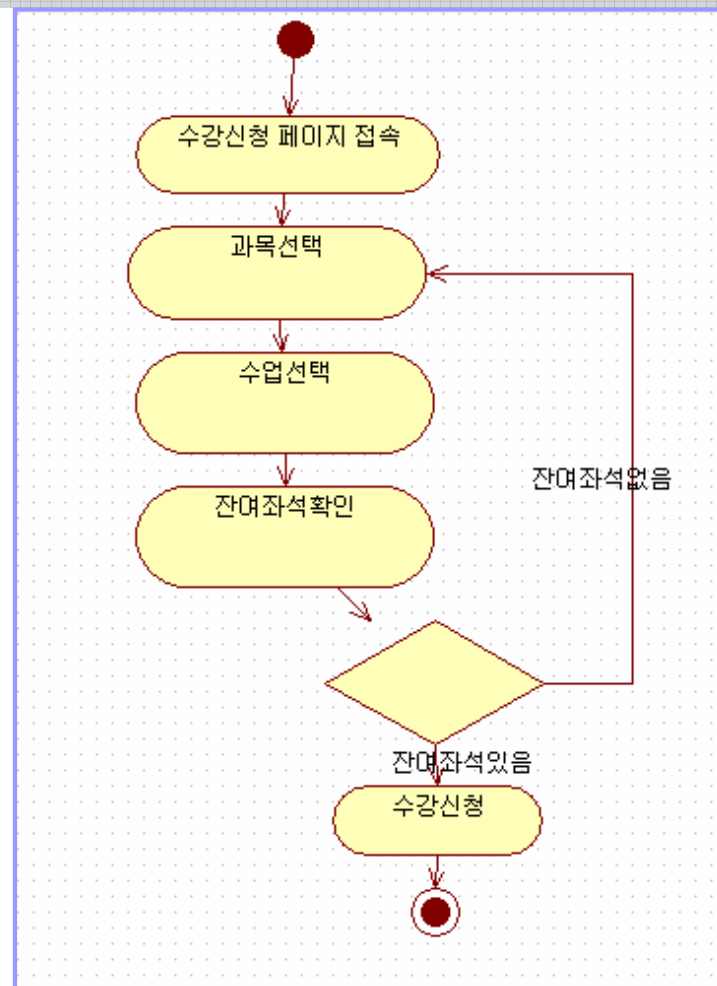


### 3. 도구 기능 소개

#### 3.10 Activity Diagram 그리기(1/9)

- 31p의 수강신청 시스템의 요구사항을 Activity Diagram으로 표현해 보겠습니다.

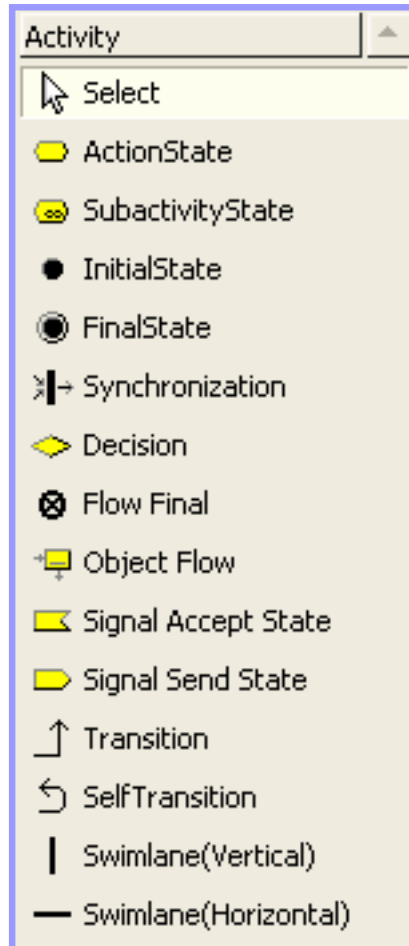
수강신청 페이지에 접속하여 과목을 선택 한 뒤 수업을 선택하여 잔여좌석을 확인한다.  
잔여좌석이 없을 경우 과목선택부터 다시 시작한다.  
잔여좌석이 있을 경우 수업등록을 하고 완료하게 된다.



### 3. 도구 기능 소개

#### 3.10 Activity Diagram 그리기 (2/9)

- Activity Diagram Tool Bar

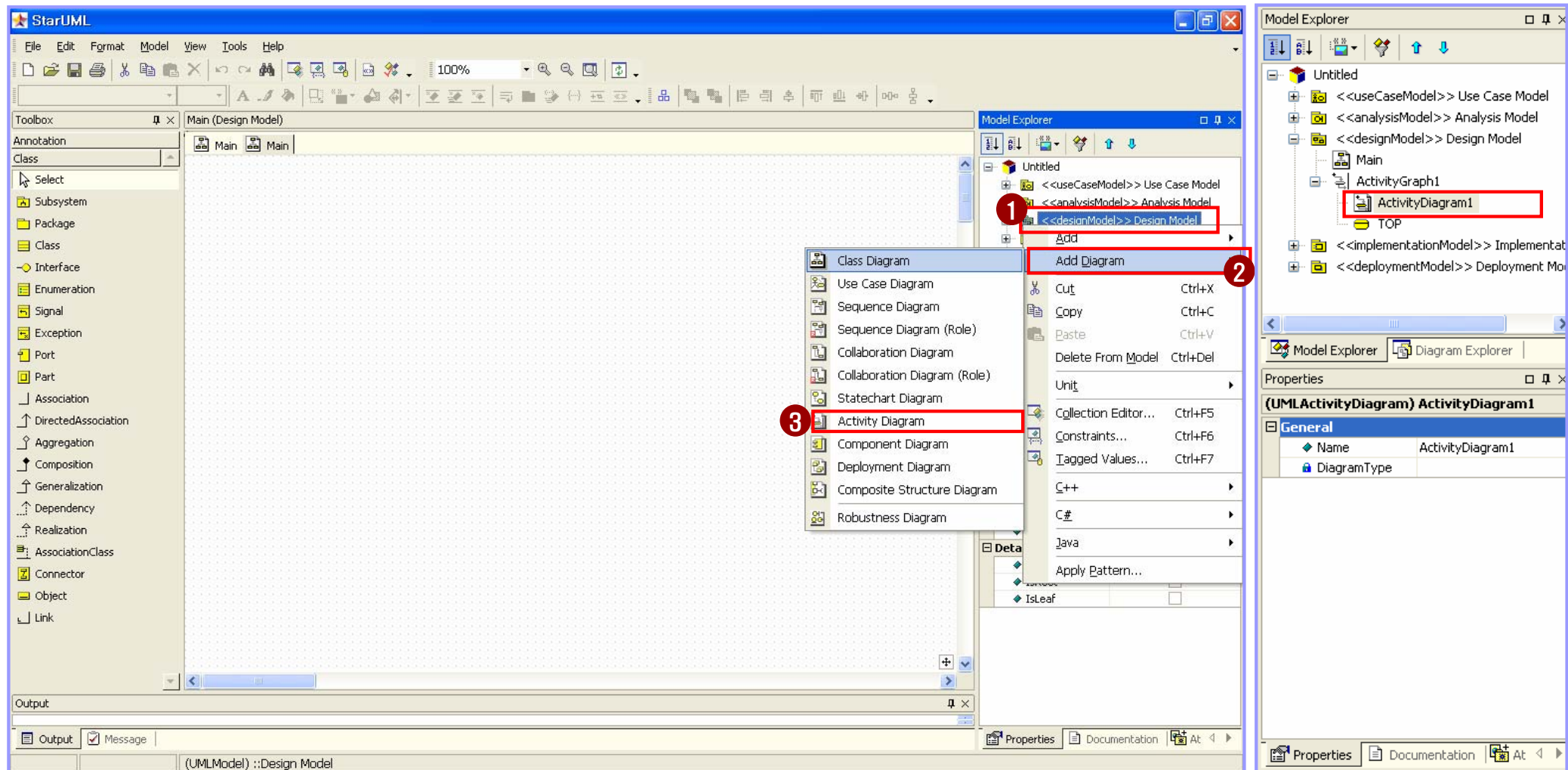


Select	Diagram 요소를 선택합니다.
ActionState	객체가 가질 수 있는 조건이나 상황을 의미합니다.
SubactivityState	하나의 하위 Activity Diagram을 의미합니다.
InitialState	객체의 활동이 시작됨을 의미합니다.
FinalState	객체의 활동이 종료됨을 의미합니다.
Synchronization	병렬 처리(동시 처리)가 시작되거나 끝나는 곳을 의미합니다.
Decision	객체의 조건에 따른 분기를 표시할 때 사용합니다.
Flow Final	흐름이 종료됨을 의미합니다.
Object Flow	
Signal Accept State	객체가 입력 값을 받을 때 사용합니다.
Signal Send State	입력 값을 다른 객체에 전달하여, 그 객체가 상태를 바꾸거나 다른 행동을 취하게 할 때 사용합니다.
Tracsion	하나의 상태에서 다른 상태로 전이됨을 의미합니다.
SelfTransition	활동 자체가 하나의 상태에서 다른 상태로 전이됨을 의미합니다.
Swimlane(Vertical)	Swimlane 의 세로축을 의미합니다.
Swimlane(Horizontal)	Swimlane 의 가로축을 의미합니다.

# 3. 도구 기능 소개

## 3.10 Activity Diagram 그리기(3/9)

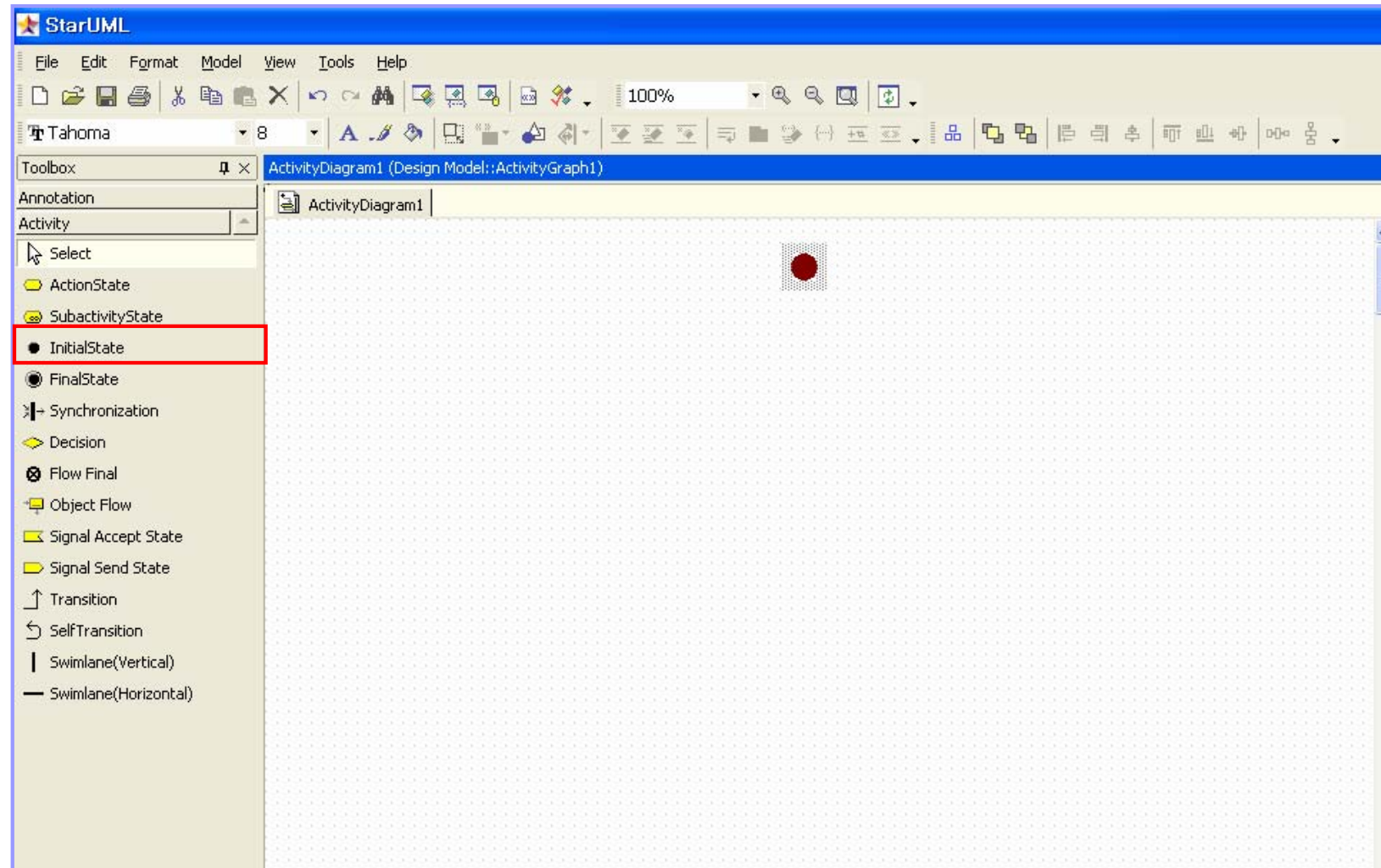
- Activity Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 각 Diagram에 적합한 Model 선정 -> 오른쪽 버튼 클릭 -> Add Diagram -> Activity Diagram 선택



## 3. 도구 기능 소개

### 3.10 Activity Diagram 그리기(4/9)

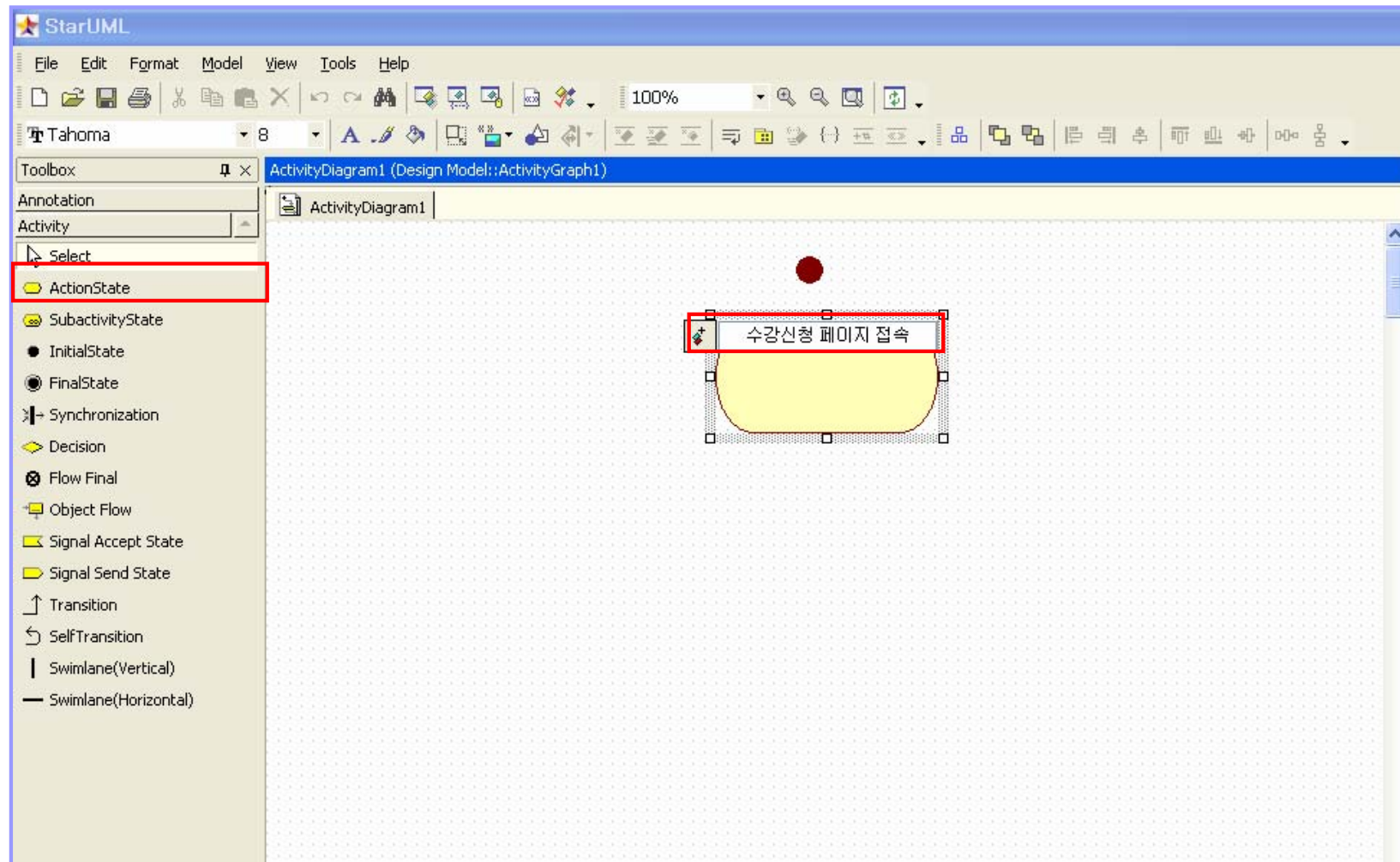
- 객체의 상태변화가 시작됨을 알려줍니다.
  - Toolbox의 Activity 에서 InitialState 아이콘 클릭 -> 마우스로 크기 지정



## 3. 도구 기능 소개

### 3.10 Activity Diagram 그리기(5/9)

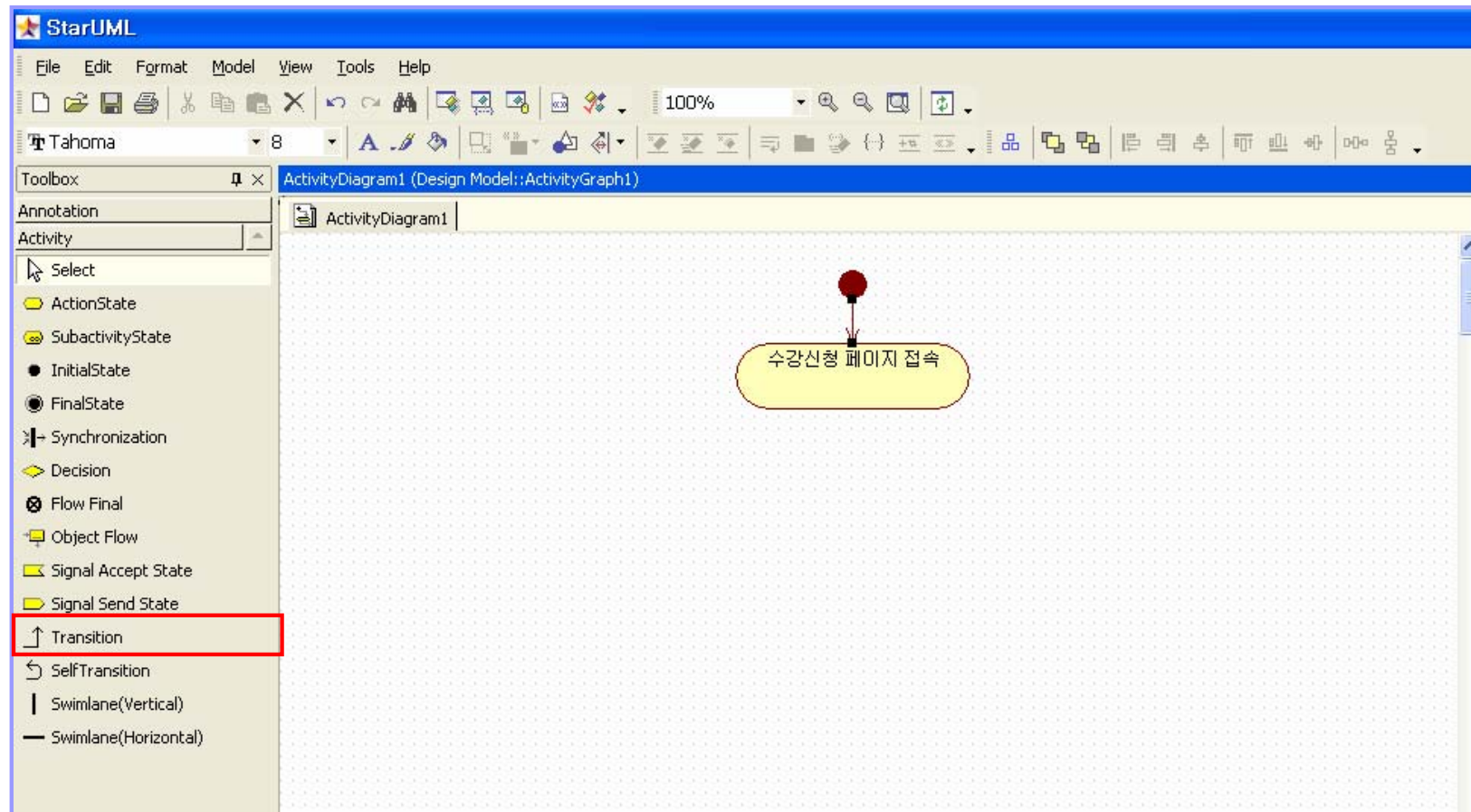
- 작업을 알려줍니다.
  - Toolbox의 Activity 에서 ActionState 아이콘 클릭 -> 마우스로 ActionState 크기 지정-> ActionState 내용 입력 -> Enter



## 3. 도구 기능 소개

### 3.10 Activity Diagram 그리기(6/9)

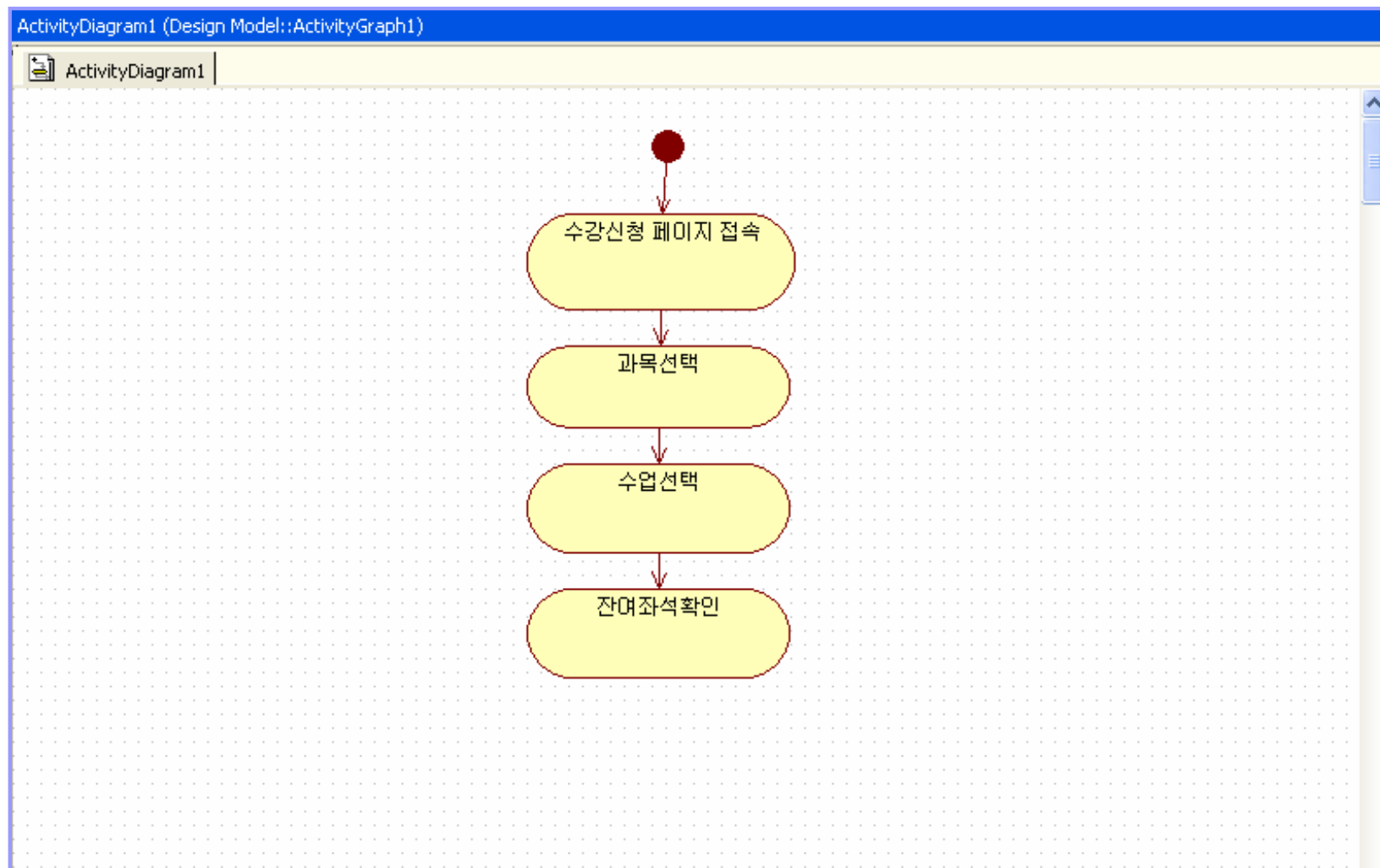
- 하나의 상태에서 다른 상태로 변화되는 것을 표현합니다.
  - Toolbox의 Activity 에서 Transition 아이콘 클릭 -> 위에서 아래로 드래그



### 3. 도구 기능 소개

#### 3.10 Activity Diagram 그리기(7/9)

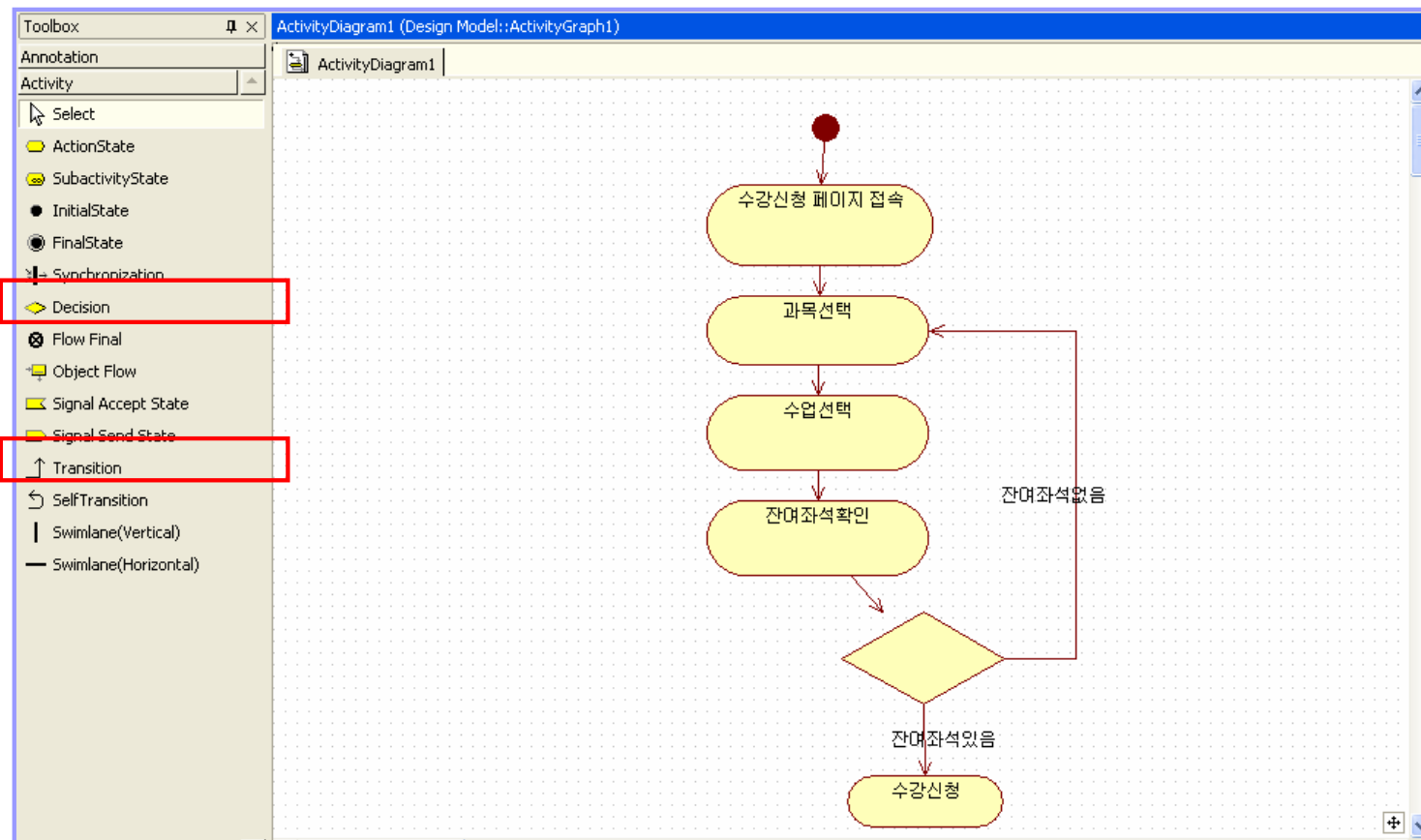
- ActionState과 Transition 생성 방법을 참고하여 다음과 같이 작성합니다.



# 3. 도구 기능 소개

## 3.10 Activity Diagram 그리기(8/9)

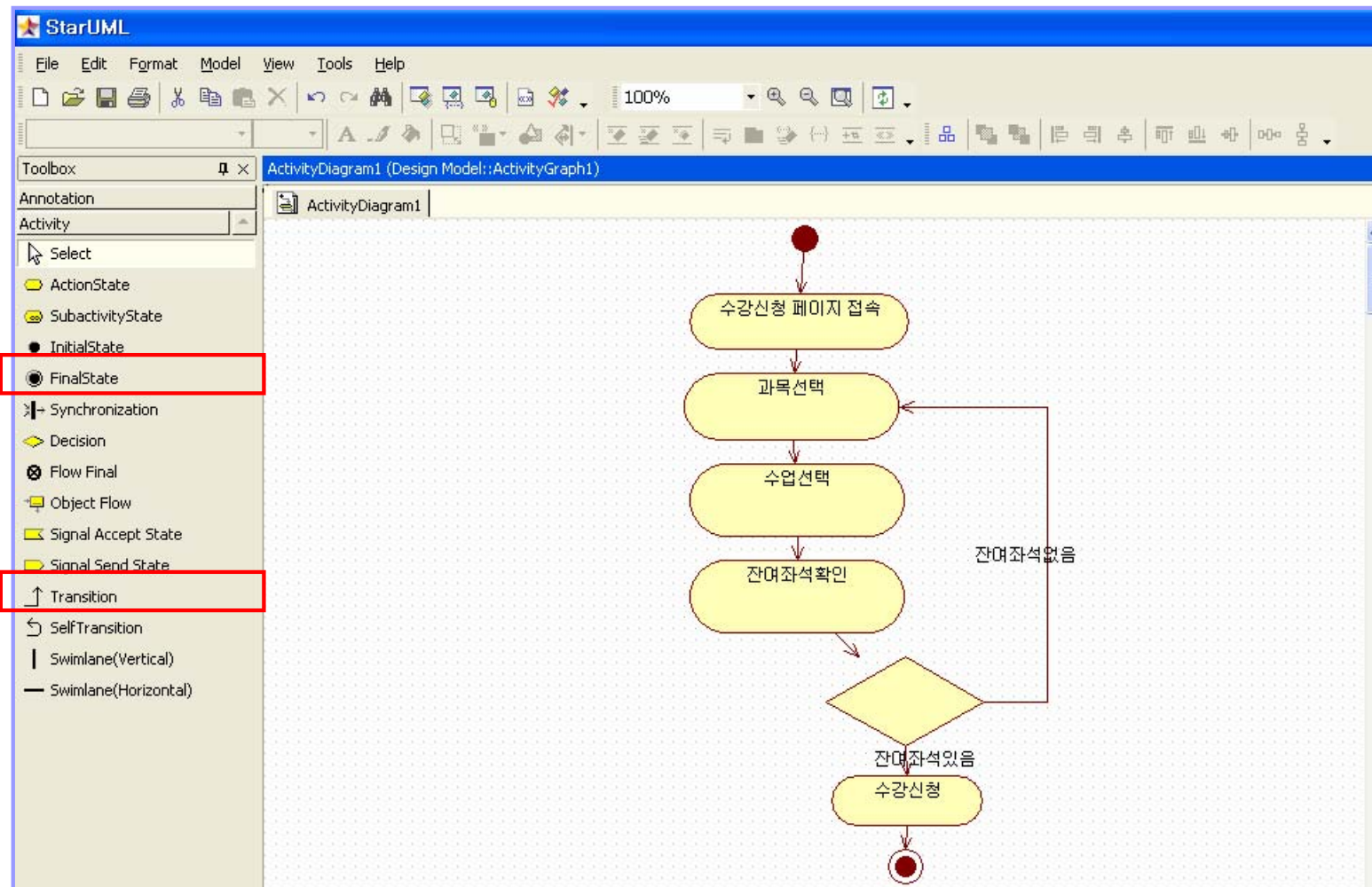
- 논리식의 결과 값에 따라 분기가 일어나는 곳을 표현합니다.
  - Toolbox의 Activity 에서 Decision 선택 -> 마우스로 크기 설정
  - Toolbox의 Activity 에서 Transition 선택 -> 분기가 일어나는 곳과 Decision부분을 이어줍니다. (본 예제에서는 과목선택)
  - 이어준 선을 더블클릭 하여 내용을 입력합니다.



# 3. 도구 기능 소개

## 3.10 Activity Diagram 그리기(9/9)

- 객체의 상태변화가 끝나게 됨을 알려줍니다.
  - Toolbox의 Activity 에서 FinalState 아이콘 클릭 -> 마우스로 크기 지정
  - Toolbox의 Activity 에서 Transition 아이콘 클릭 -> 위에서 아래로 드래그

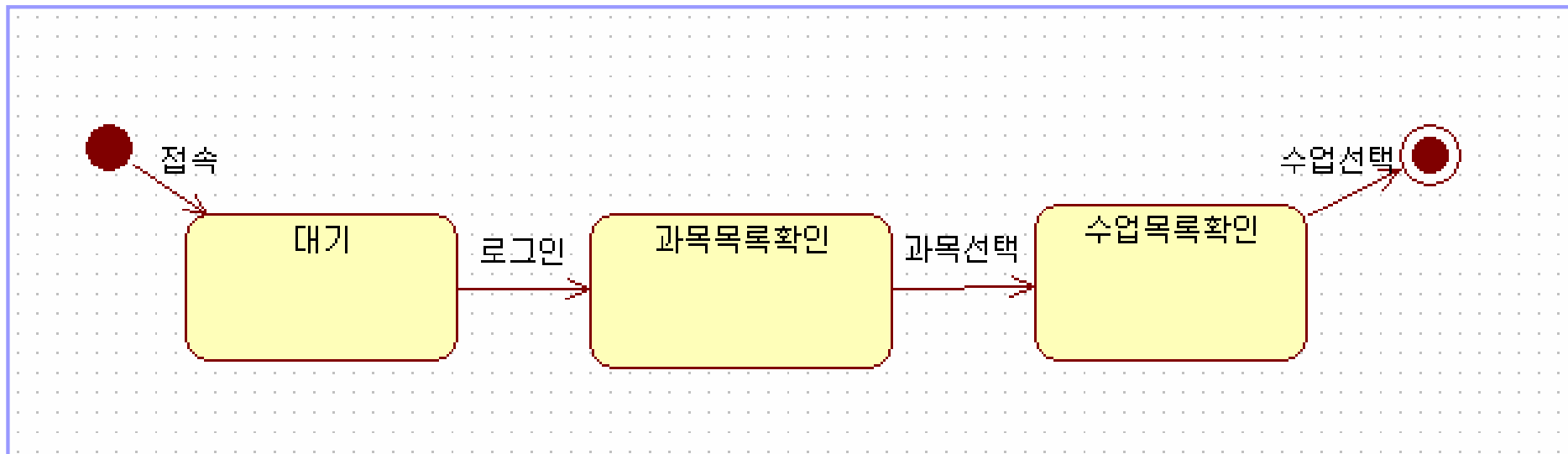


### 3. 도구 기능 소개

#### 3.11 Statechart Diagram 그리기(1/9)

- 31p의 수강신청 시스템의 요구사항을 Statechart Diagram으로 표현해 보겠습니다.

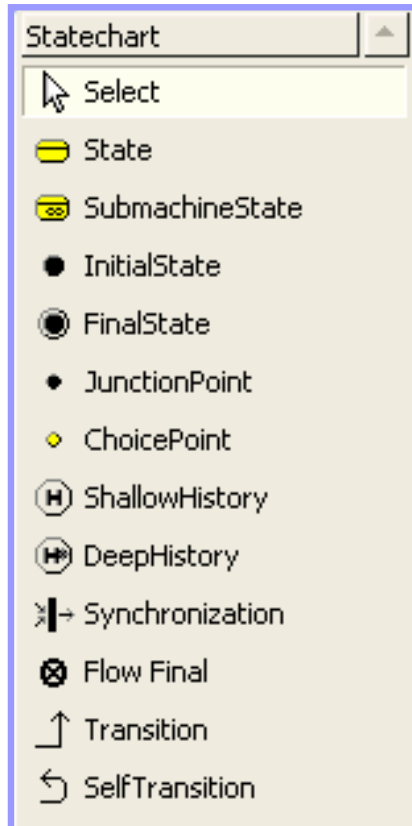
대기상태에서 수강신청을 하기 위해 수강신청 페이지에 접속한다. 로그인을 하고 과목목록을 확인한 뒤 과목을 선택한다. 선택한 과목의 수업을 선택하면 수강신청이 완료된다.



## 3. 도구 기능 소개

### 3.11 Statechart Diagram 그리기 (2/9)

#### • Statechart Diagram Tool Bar

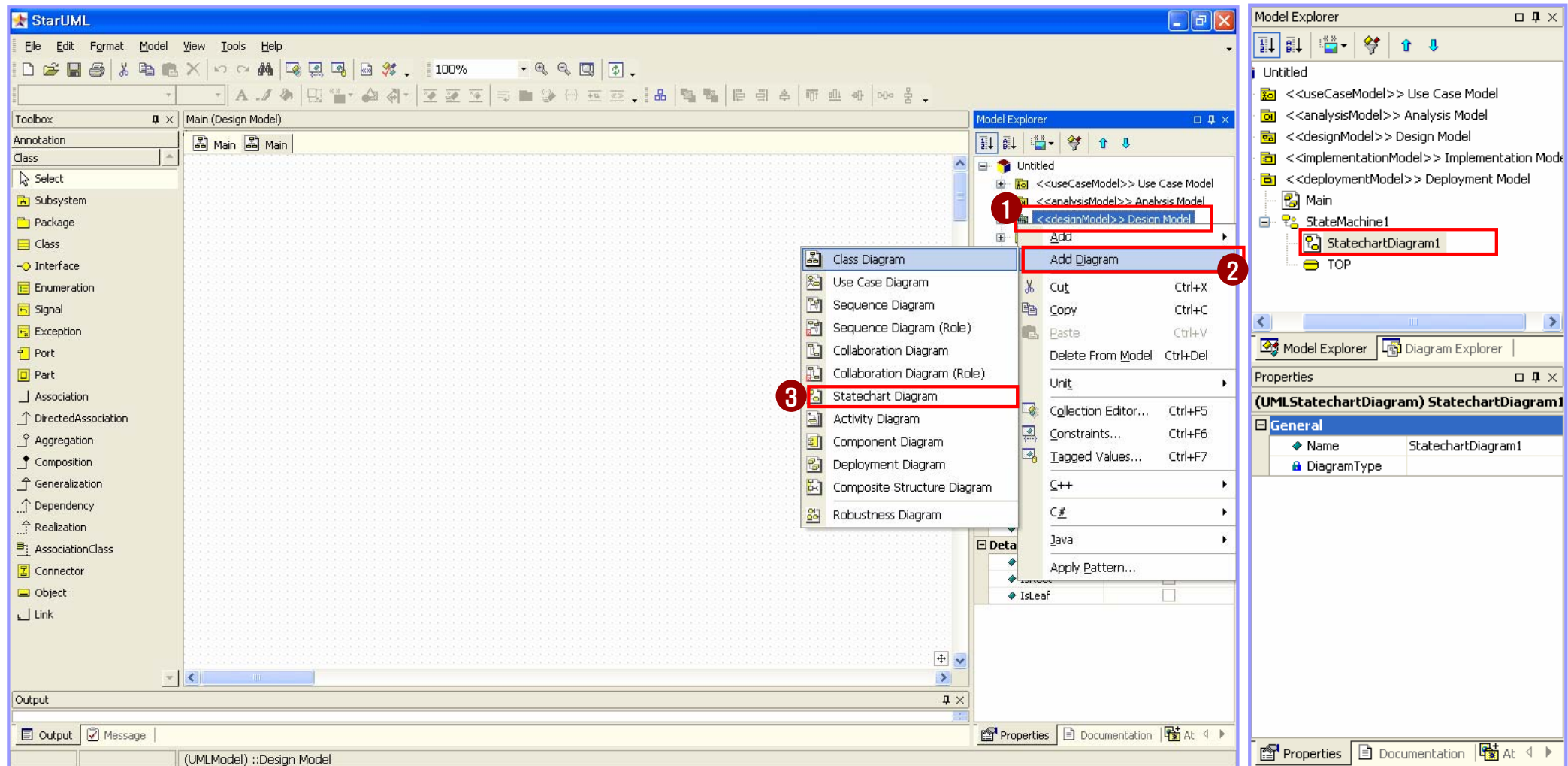


Select	Diagram 요소를 선택합니다.
State	객체의 상태를 나타내기 위해 사용합니다.
SubmachineState	하나의 상태 머신을 나타내기 위해 사용합니다.
InitialState	객체의 상태 변화가 시작되는 곳을 의미합니다.
FinalState	객체의 상태 변화가 종료되는 곳을 의미합니다.
JunctionPoint	분기의 시작 또는 종료 지점을 의미합니다.
ChoicePoint	선택 지점을 의미합니다.
ShallowHistory	객체의 직전의 상태 결과를 의미합니다.
DeepHistory	객체의 모든 상태 결과를 의미합니다.
Synchronization	상태들이 합쳐지는 바를 의미합니다.
Flow Final	흐름이 종료됨을 의미합니다.
Transition	한 상태에서 다른 상태로의 변환 상태를 의미합니다.
SelfTransition	한 상태가 스스로의 상태로 변환을 의미합니다.

# 3. 도구 기능 소개

## 3.11 Statechart Diagram 그리기(3/9)

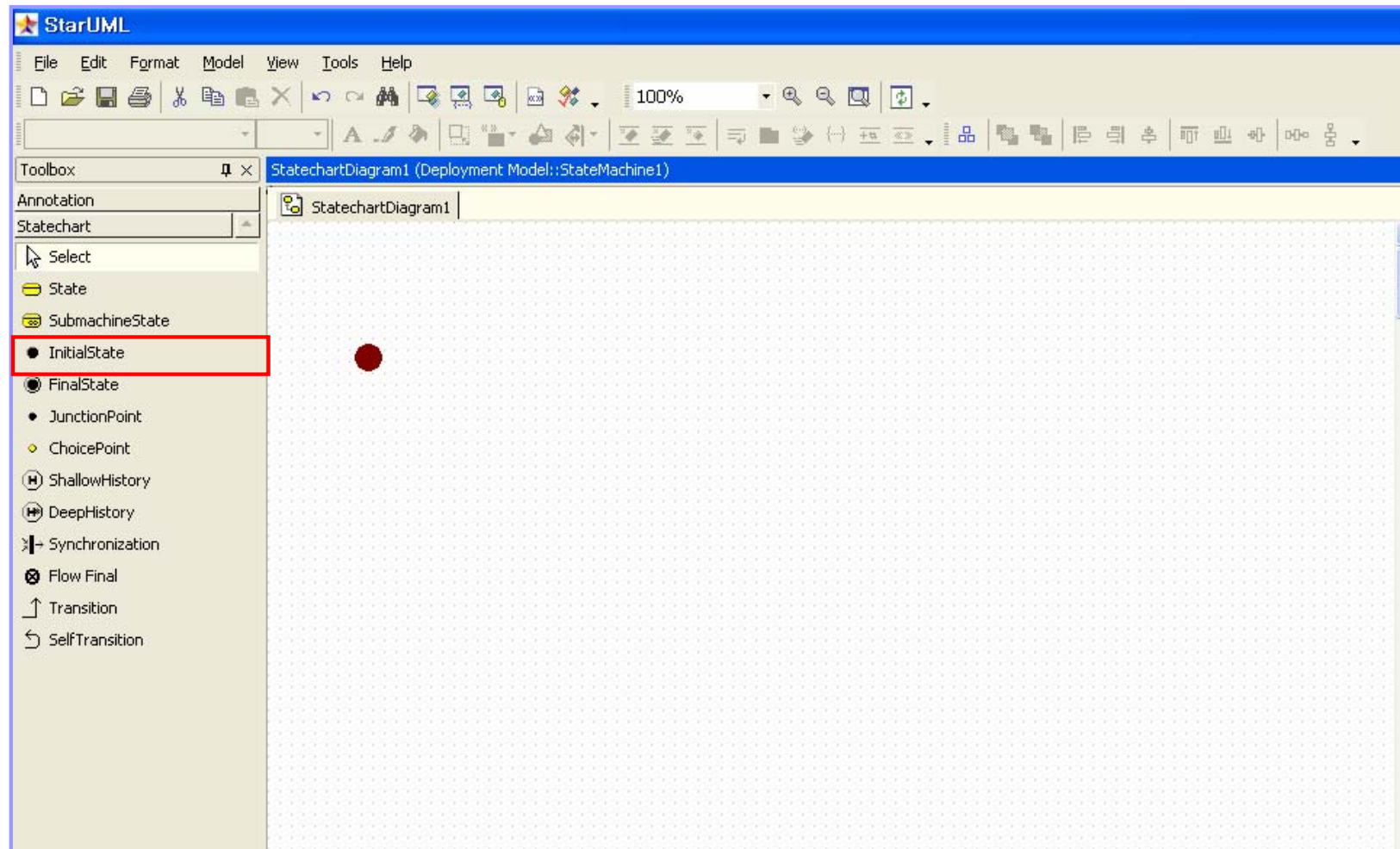
- Statechart Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 각 Diagram에 적합한 Model 선정 -> 오른쪽 버튼 클릭 -> Add Diagram -> Statechart Diagram 선택



## 3. 도구 기능 소개

### 3.11 Statechart Diagram 그리기(4/9)

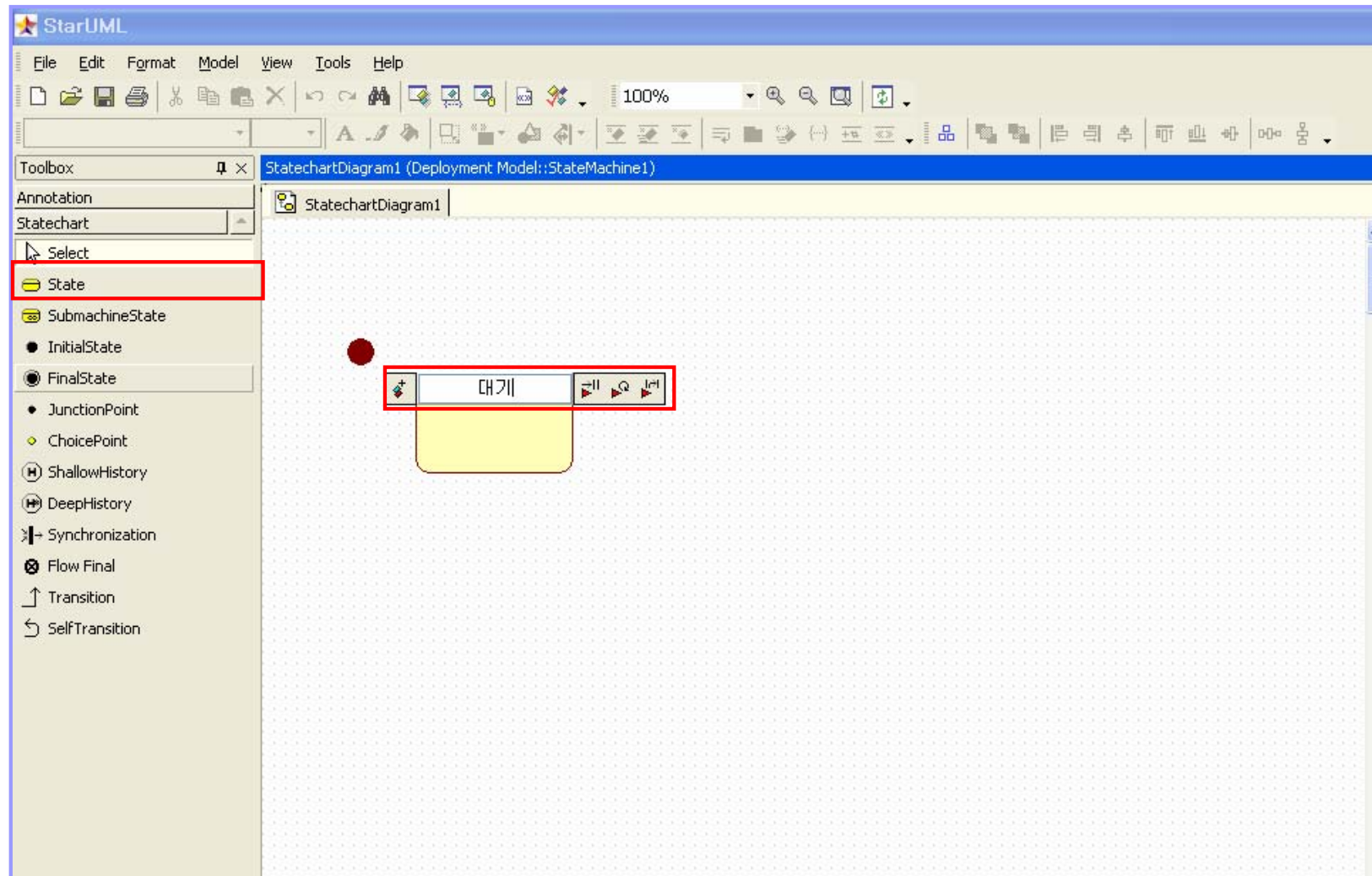
- 객체의 상태변화가 시작됨을 알려줍니다.
  - Toolbox의 Statechart 에서 InitialState 아이콘 클릭 -> 마우스로 크기 지정



## 3. 도구 기능 소개

### 3.11 Statechart Diagram 그리기(5/9)

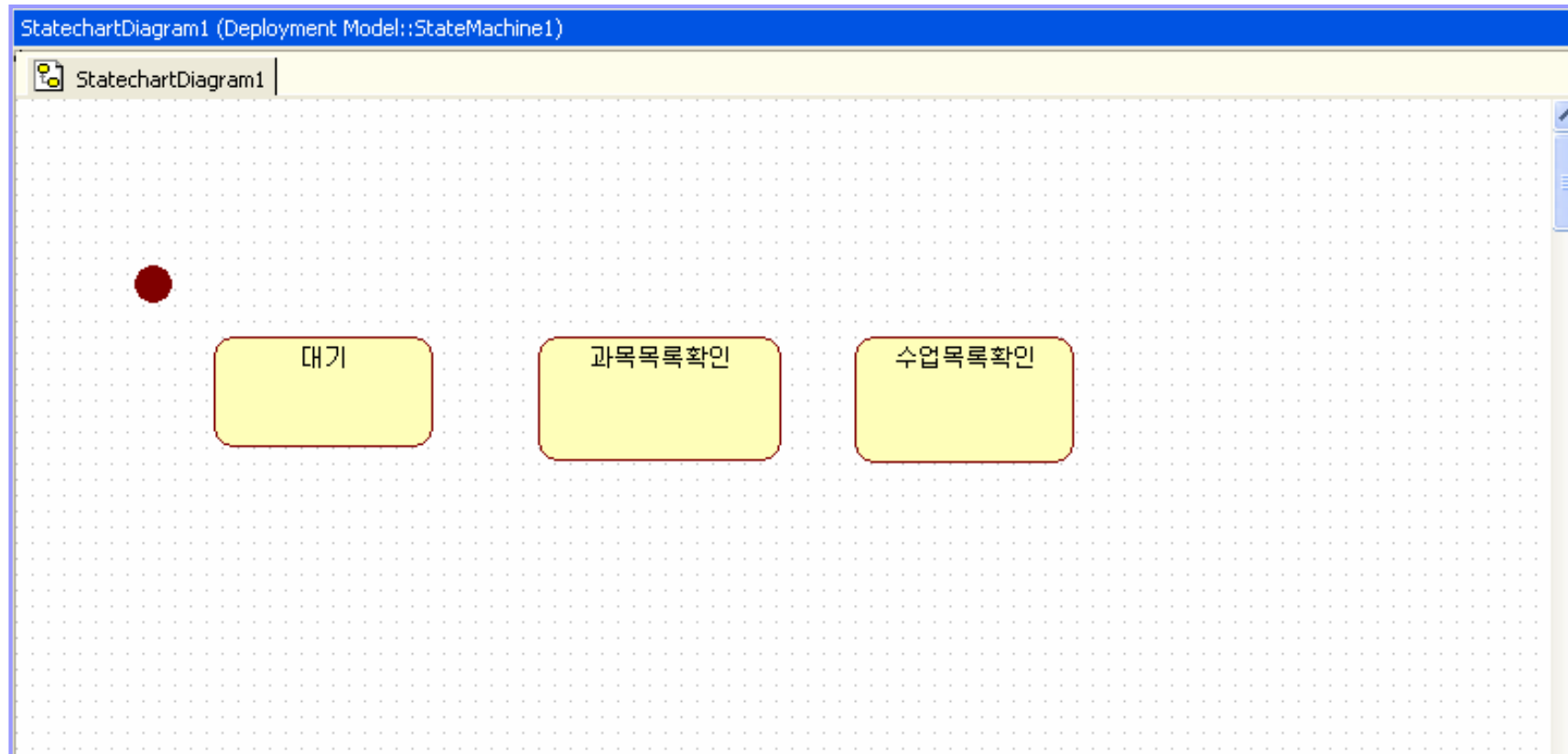
- 객체의 상태를 표현합니다.
  - Toolbox의 Statechart 에서 State 아이콘 클릭 -> 마우스로 State 크기 지정 -> State 내용 입력 -> Enter



### 3. 도구 기능 소개

#### 3.11 Statechart Diagram 그리기[6/9]

- State 생성 방법을 참고하여 과목목록확인, 수업목록확인 객체의 상태를 표현합니다.

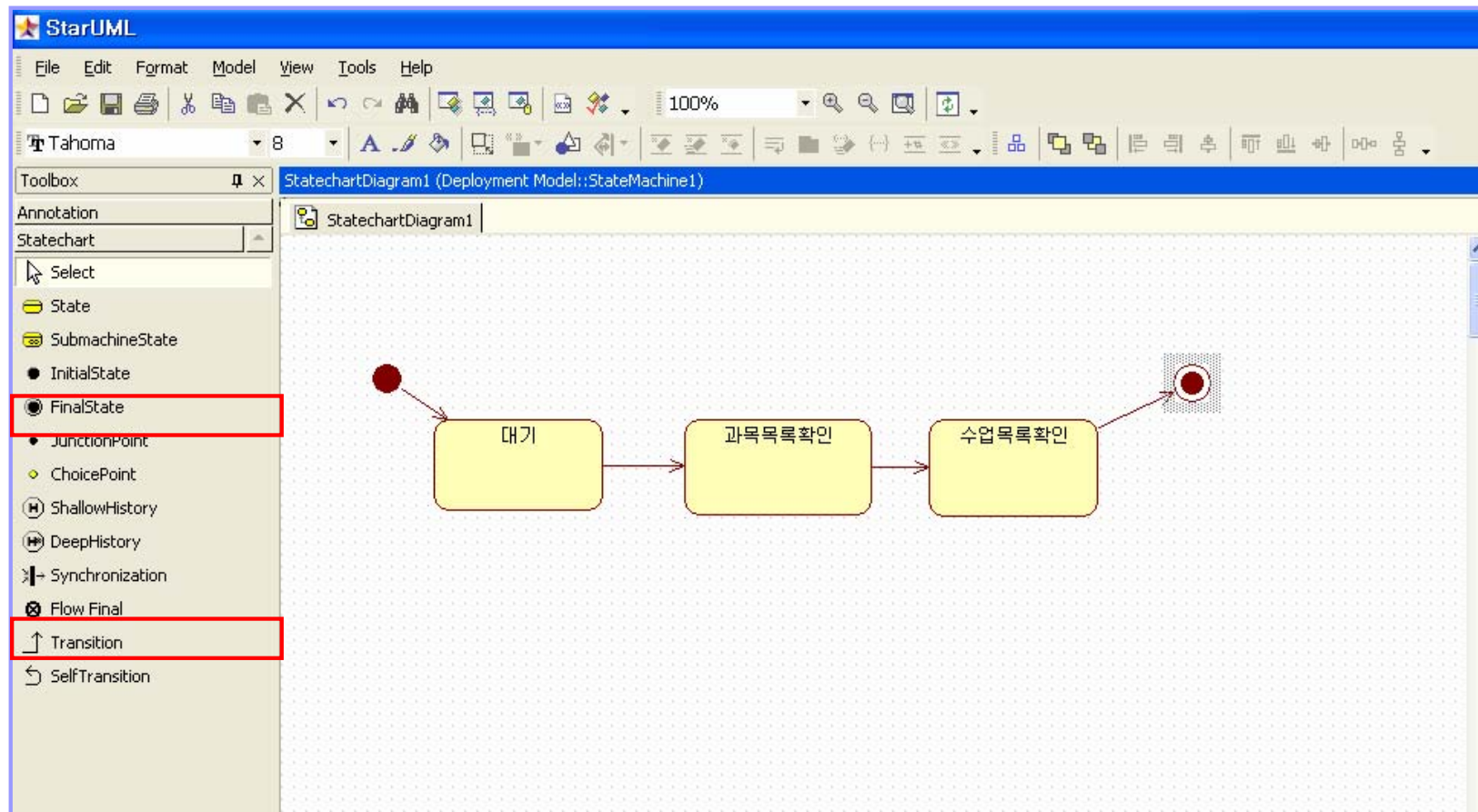




## 3. 도구 기능 소개

### 3.11 Statechart Diagram 그리기(8/9)

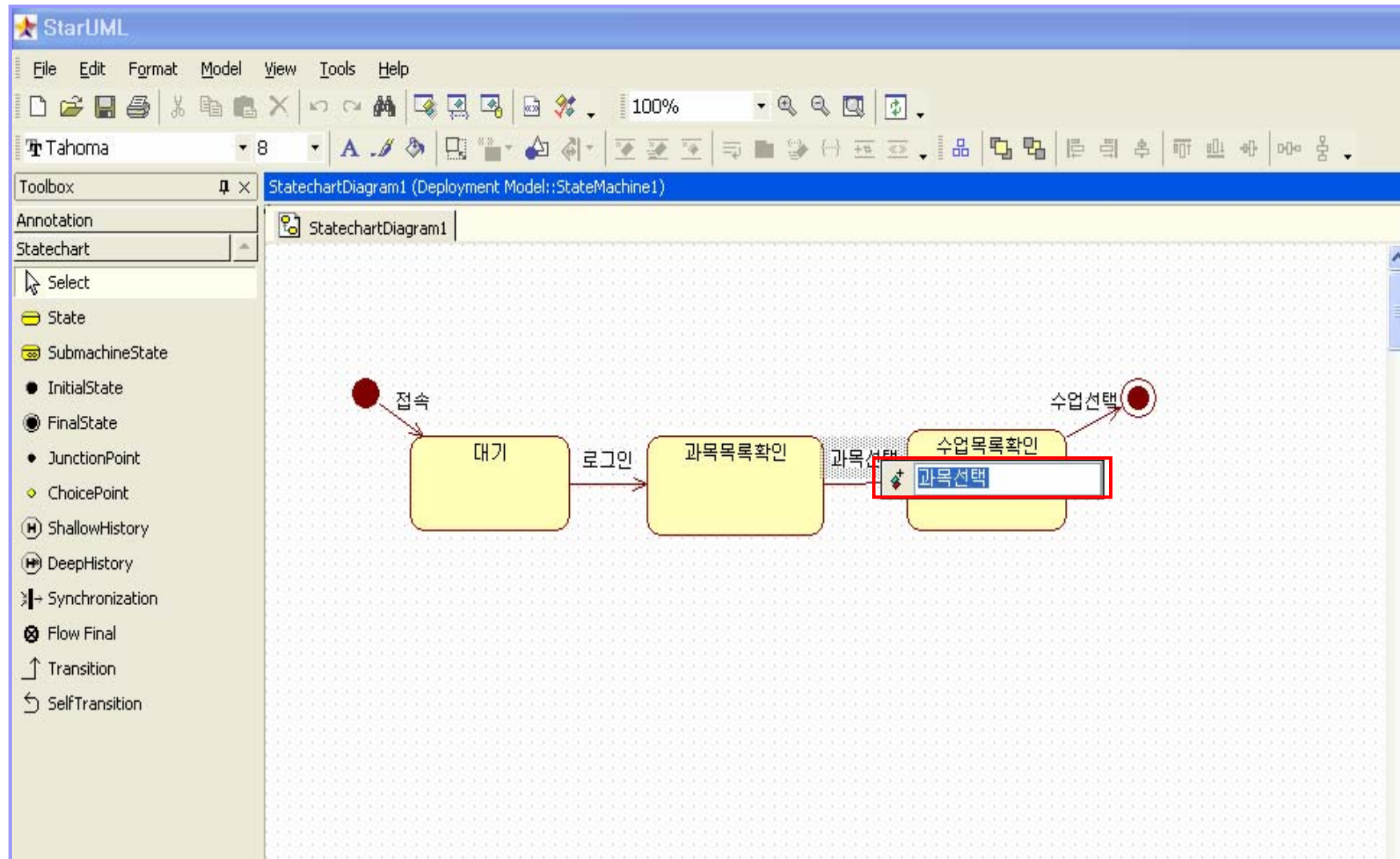
- 객체의 상태변화가 끝나게 됨을 알려줍니다.
  - Toolbox의 Statechart 에서 FinalState 아이콘 클릭 -> 마우스로 크기 지정
  - Toolbox의 Statechart 에서 Transition 아이콘 클릭 -> State와 FinalState 연결



## 3. 도구 기능 소개

### 3.11 Statechart Diagram 그리기(9/9)

- 객체의 상태 변화를 표현합니다.
  - 화살표 선택 더블클릭 -> 해당 내용 입력

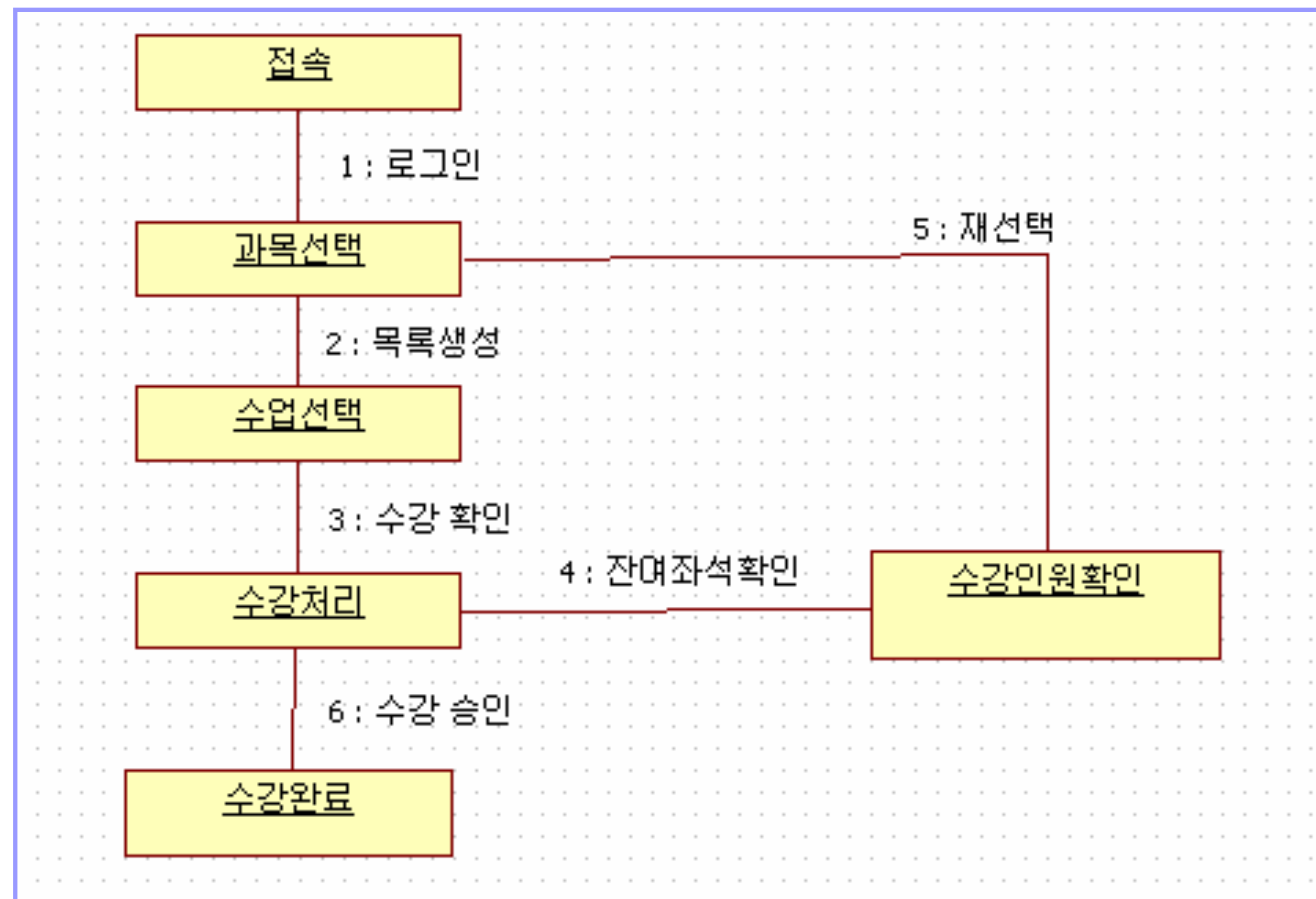


### 3. 도구 기능 소개

#### 3.12 Collaboration Diagram 그리기(1/7)

- 31p의 수강신청 시스템의 요구사항을 Collaboration Diagram으로 표현해 보겠습니다.

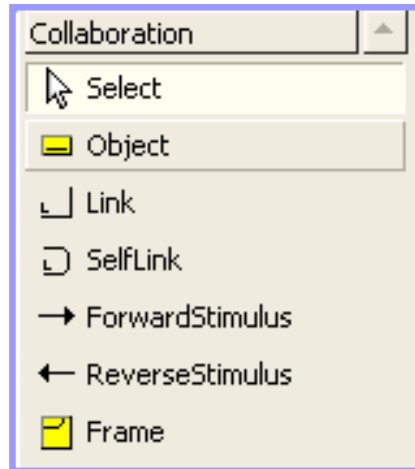
수강신청 페이지에 접속하여 과목을 선택 한 뒤 수업을 선택하여 잔여좌석을 확인한다.  
잔여좌석이 없을 경우 과목선택부터 다시 시작한다.  
잔여좌석이 있을 경우 수업등록을 하고 완료하게 된다.



## 3. 도구 기능 소개

### 3.12 Collaboration Diagram 그리기 (2/7)

- Collaboration Diagram Tool Bar

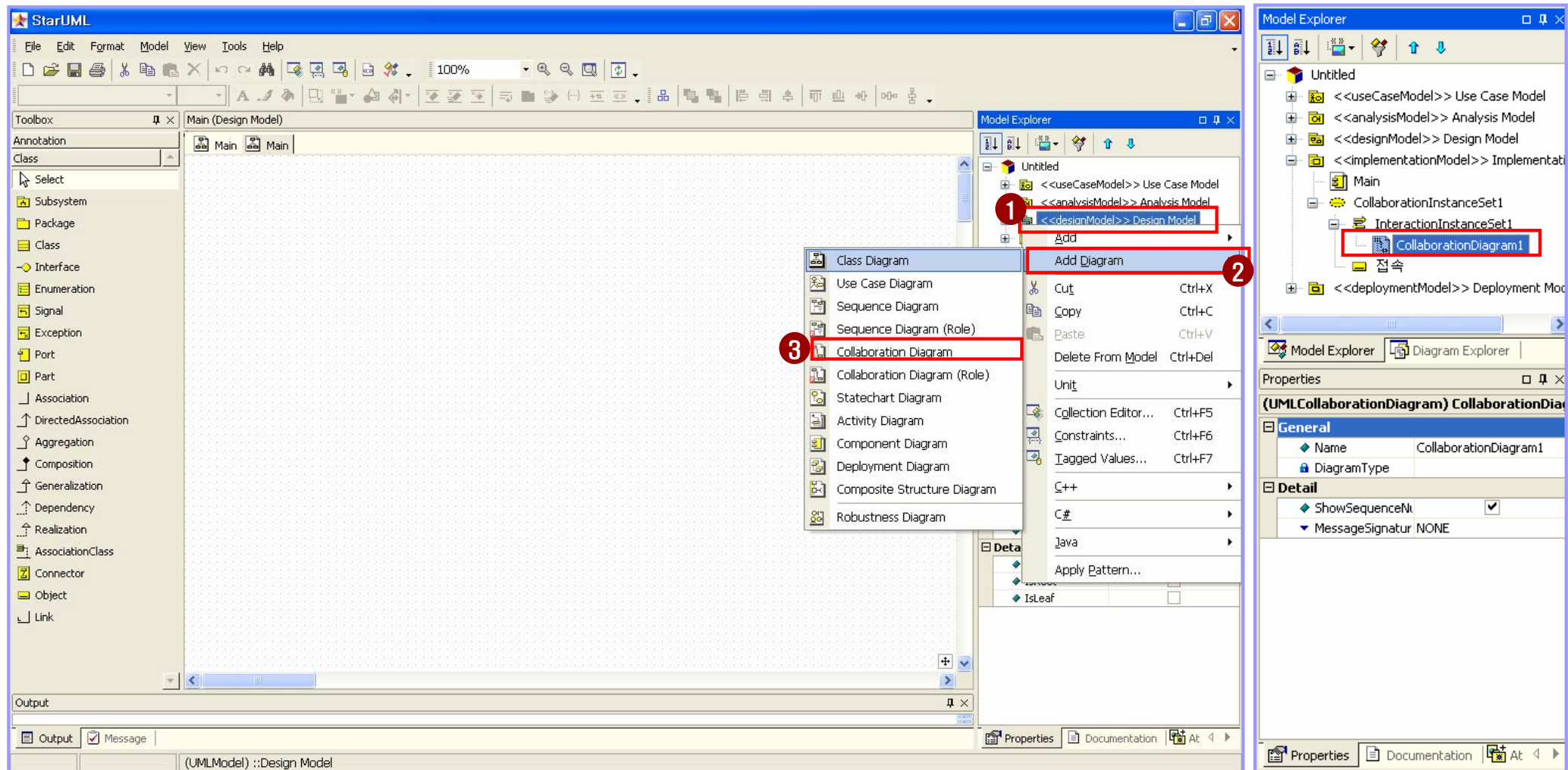


Select	Diagram 요소를 선택합니다.
Object	클래스의 객체를 의미합니다.
Link	객체들간의 관계를 의미합니다.
SelfLink	한 객체가 자신과의 연결을 의미합니다.
FowardStimulus	Link에 Forward Stimulus 설정을 추가합니다.
ReverseStimulus	Link에 Reverse Stimulus 설정을 추가합니다.
Frame	다이어그램을 특정 영역으로 구분할 때 사용합니다.

# 3. 도구 기능 소개

## 3.12 Collaboration Diagram 그리기 (3/7)

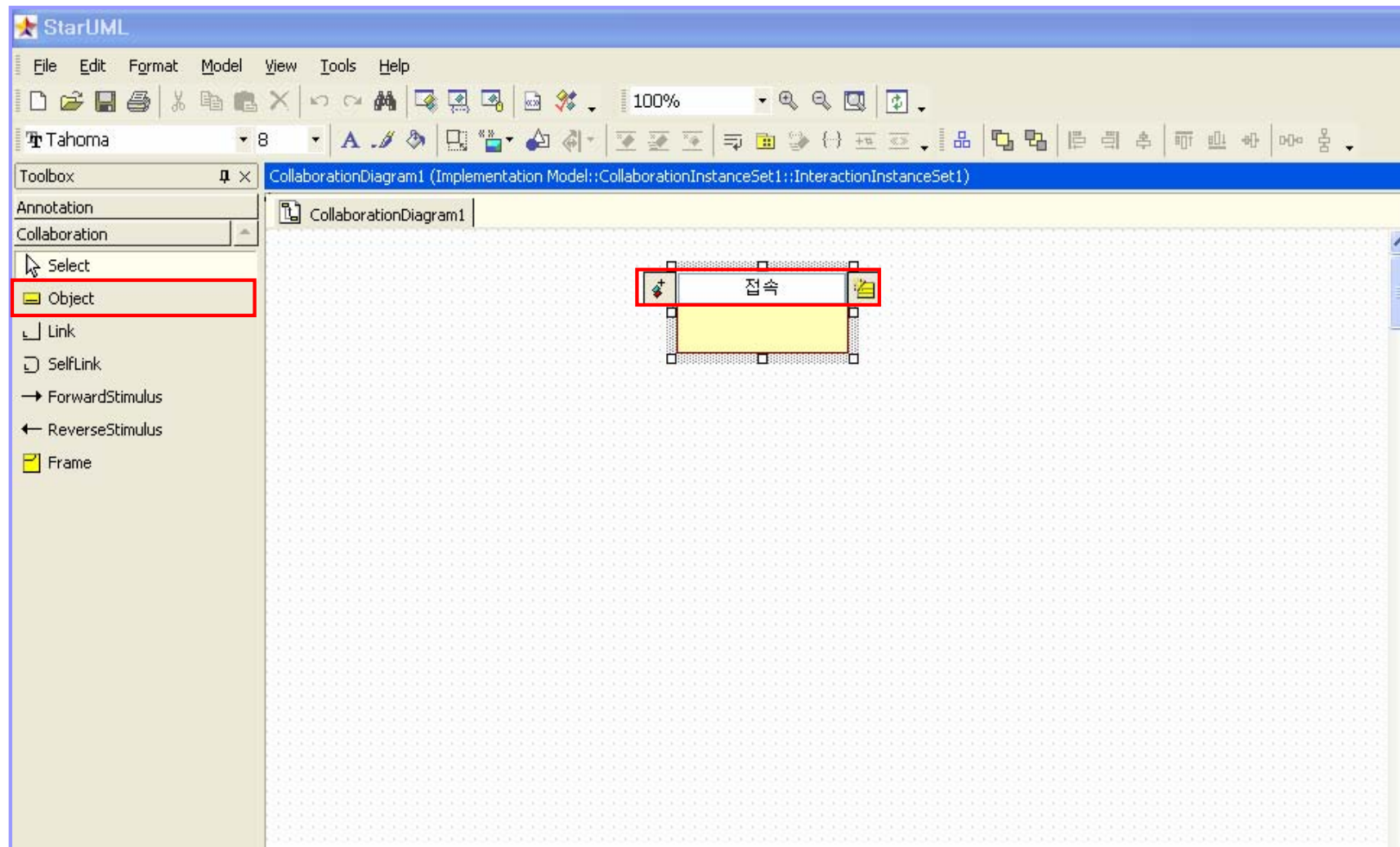
- Collaboration Diagram을 그리기 위해서 Diagram을 생성합니다.
  - Model Explorer의 각 Diagram에 적합한 Model 선정 -> 오른쪽 버튼 클릭 -> Add Diagram -> Collaboration Diagram 선택



### 3. 도구 기능 소개

#### 3.12 Collaboration Diagram 그리기 (4/7)

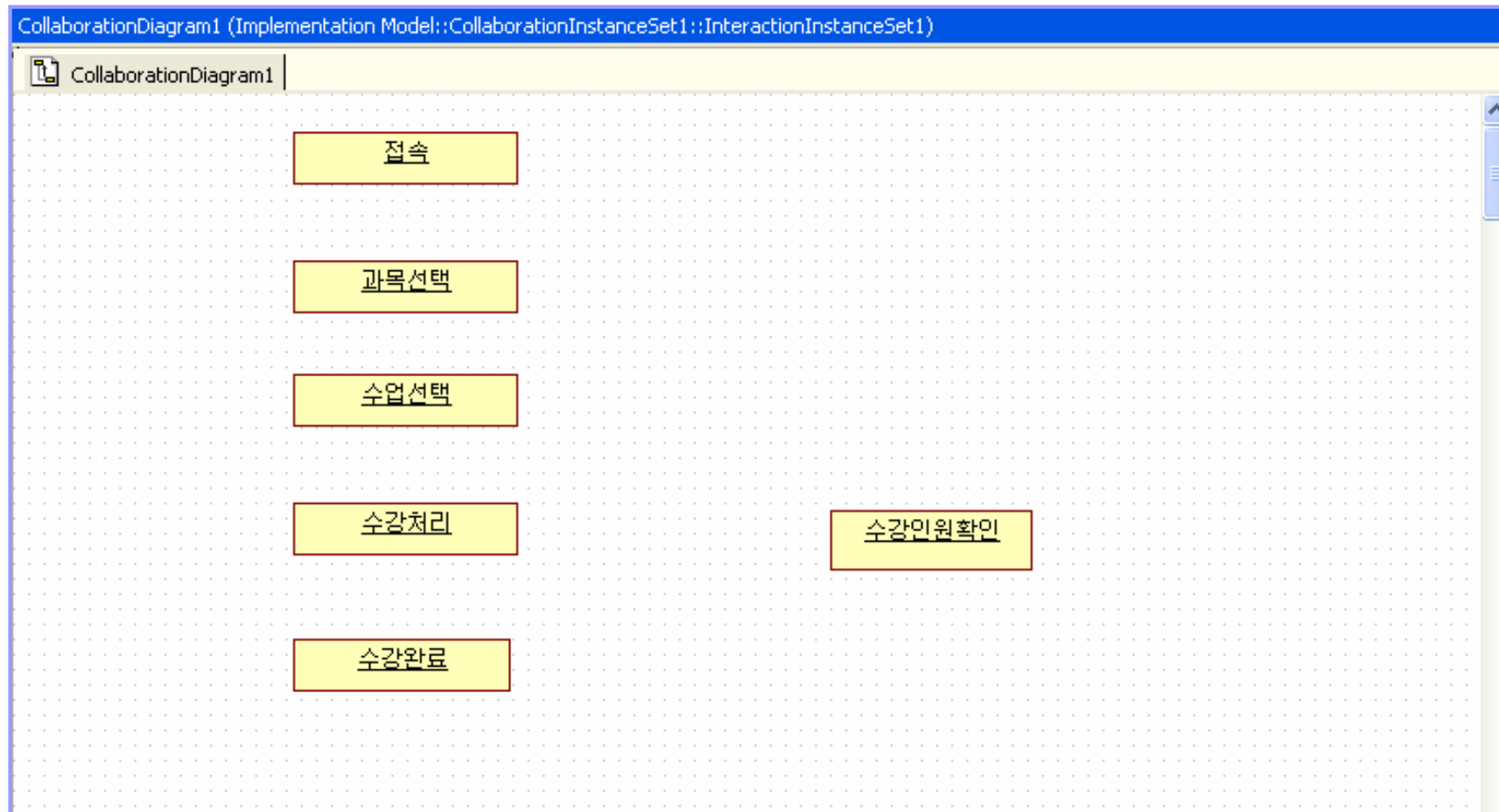
- 객체를 생성합니다.
  - Toolbox의 Collaboration 에서 Object 아이콘 클릭 -> 마우스로 Object 크기 지정-> Object 내용 입력 -> Enter



### 3. 도구 기능 소개

#### 3.12 Collaboration Diagram 그리기 (5/7)

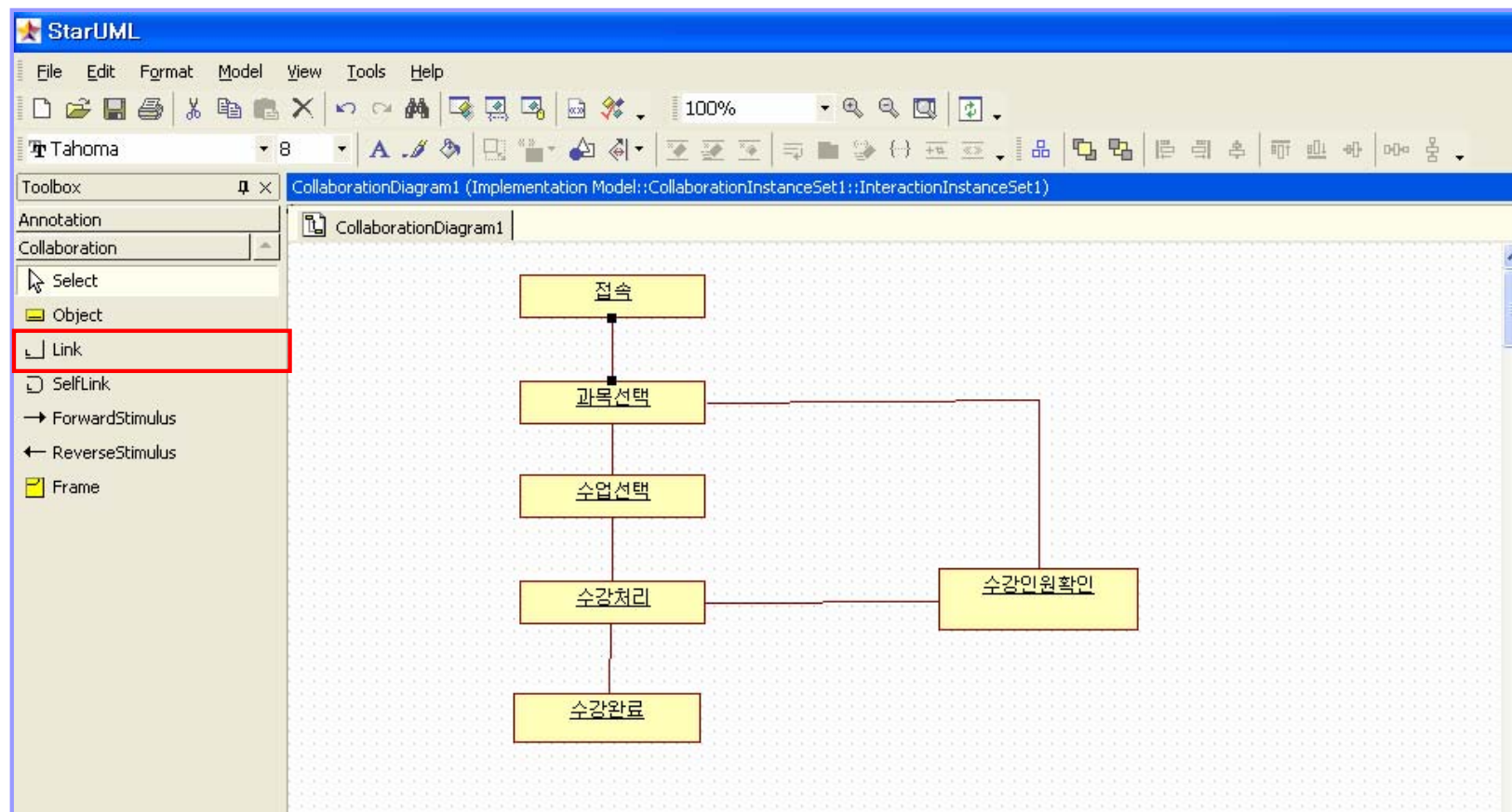
- 객체 생성하는 방법을 참고하여 과목선택, 수업선택, 수강처리, 수강인원확인, 수강완료 객체를 생성합니다.



### 3. 도구 기능 소개

#### 3.12 Collaboration Diagram 그리기 (6/7)

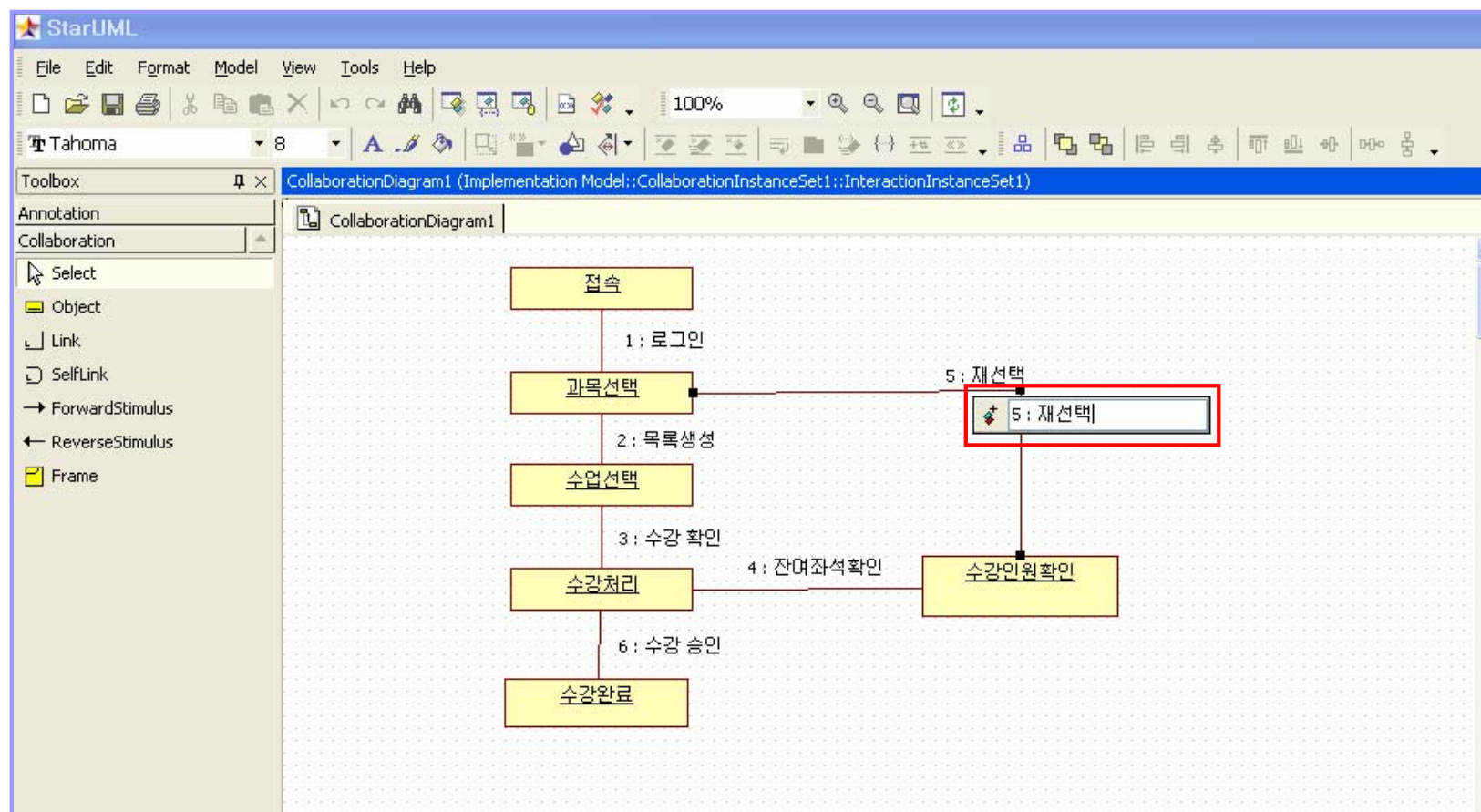
- 객체 사이의 링크를 생성합니다.
  - Toolbox의 Collaboration 에서 Link 아이콘 클릭 -> 연결할 Link와 Link 사이 드래그



### 3. 도구 기능 소개

#### 3.12 Collaboration Diagram 그리기 (7/7)

- Link의 상태 변화를 표현합니다.
  - Link 선택 더블클릭 -> 해당 내용 입력



## 4. 도구 활용 예제

## 4. 도구 활용 예제

### 세부 목차

- 4.1 예제소개
- 4.2 Use Case Diagram
- 4.3 Class Diagram
- 4.4 Sequence Diagram
- 4.5 Collaboration Diagram
- 4.6 Activity Diagram
- 4.7 Statechart Diagram

## 4. 도구 활용 예제

### 4.1 예제소개

- StarUML의 도구 활용 예제를 위해 생산 관리 시스템을 예로 표현하겠습니다.
  - 생산 관리 요구사항 중 일부입니다.

#### ▶ 요구사항

업무 담당자는 시스템에 접속하여 발주, 매입, 매출, 회계, 재고, 기준정보 관리가 가능하다. 승인자는 발주와 매입에 대한 승인을 하며, 시스템 관리자는 매출, 회계, 재고, 기준정보에 대한 시스템 관리를 수행한다.

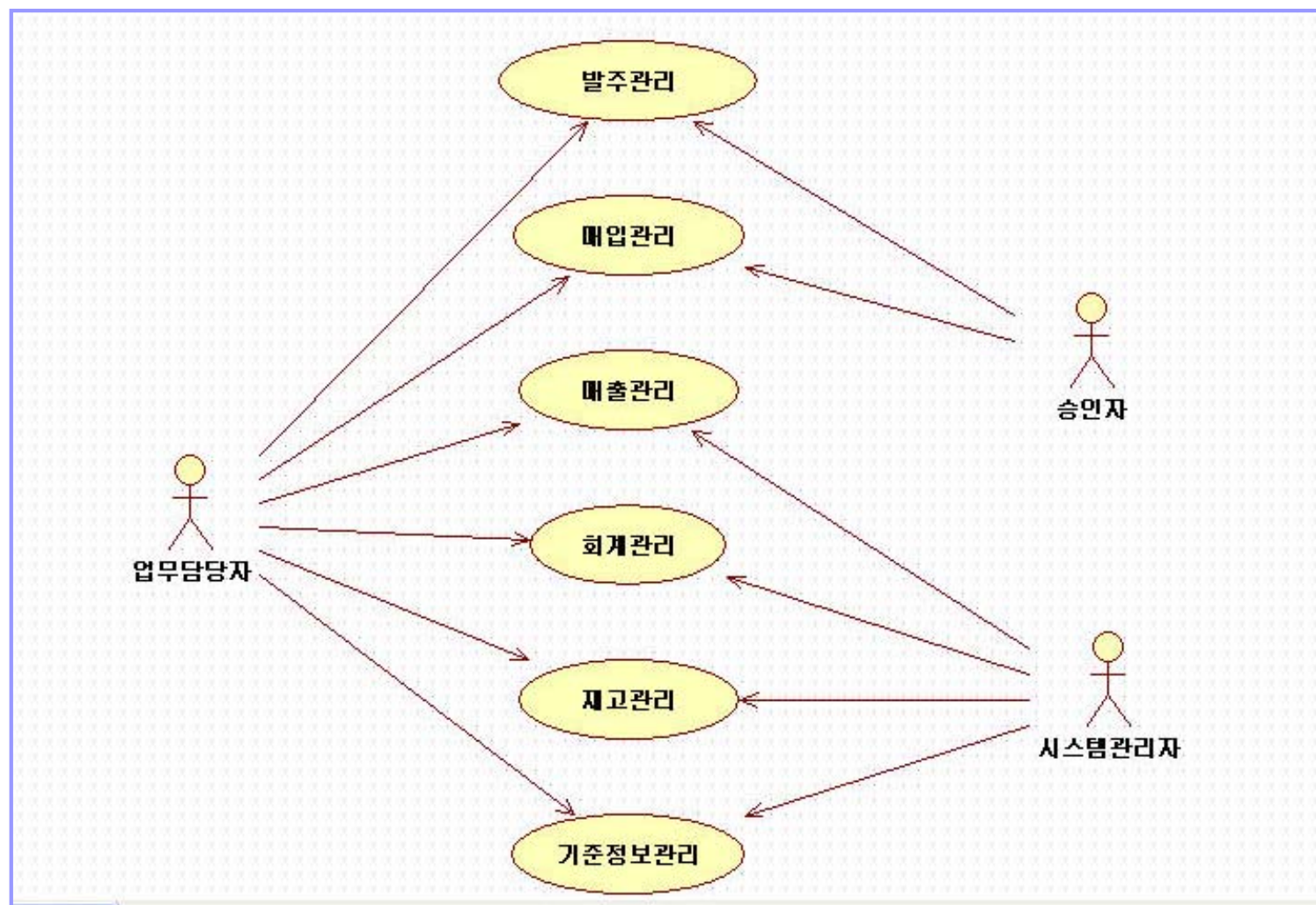
발주가 등록되면 발주를 위한 준비를 하여 재고 여부 조사 후 상품을 배송한다.

- 생산 관리 시스템의 구현 환경은 다음과 같습니다.
  - 운영체제 환경
    - » 사용자는 PC 기반의 Windows XP를 운영체제로 사용하고 있습니다.
    - » Web Server, DB Server, Web Application Server는 Windows NT 기반의 운영체제를 사용하고 있습니다.
  - 네트워크 환경
    - » 사용자와 Web Server간의 통신은 HTTP 방식을 사용하고 있습니다.
    - » 사용자와 Web Application Server간의 통신은 TCP/IP 방식을 사용하고 있습니다.
    - » Web Server와 DB Server, DB Server와 Web Application Server간의 통신은 TCP/IP 방식을 사용하고 있습니다.

## 4. 도구 활용 예제

### 4.2 Use Case Diagram (1/6)

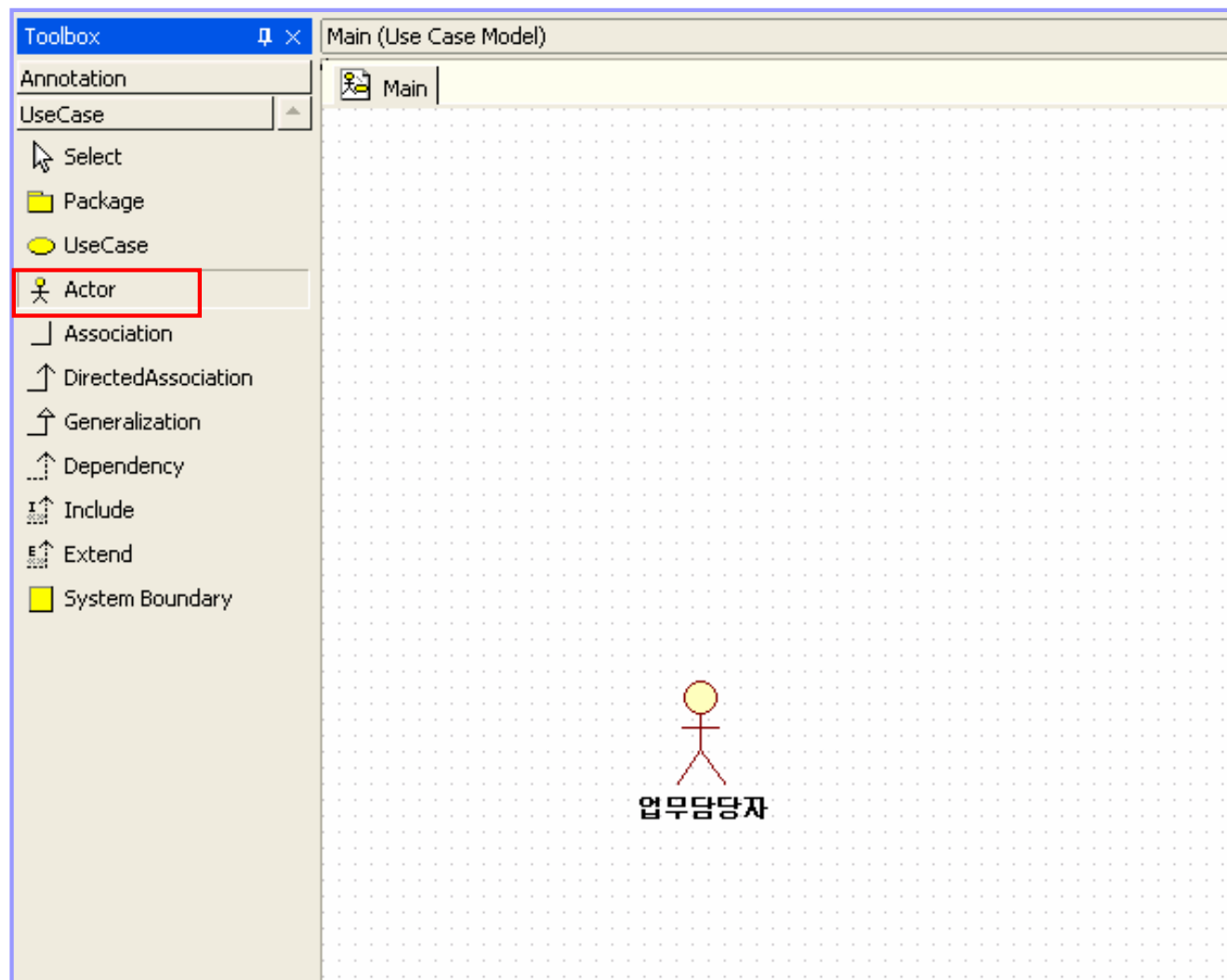
- 생산 관리 시스템의 요구사항을 다음과 같이 UseCase로 표현해 보겠습니다.



## 4. 도구 활용 예제

### 4.2 Use Case Diagram [2/6]

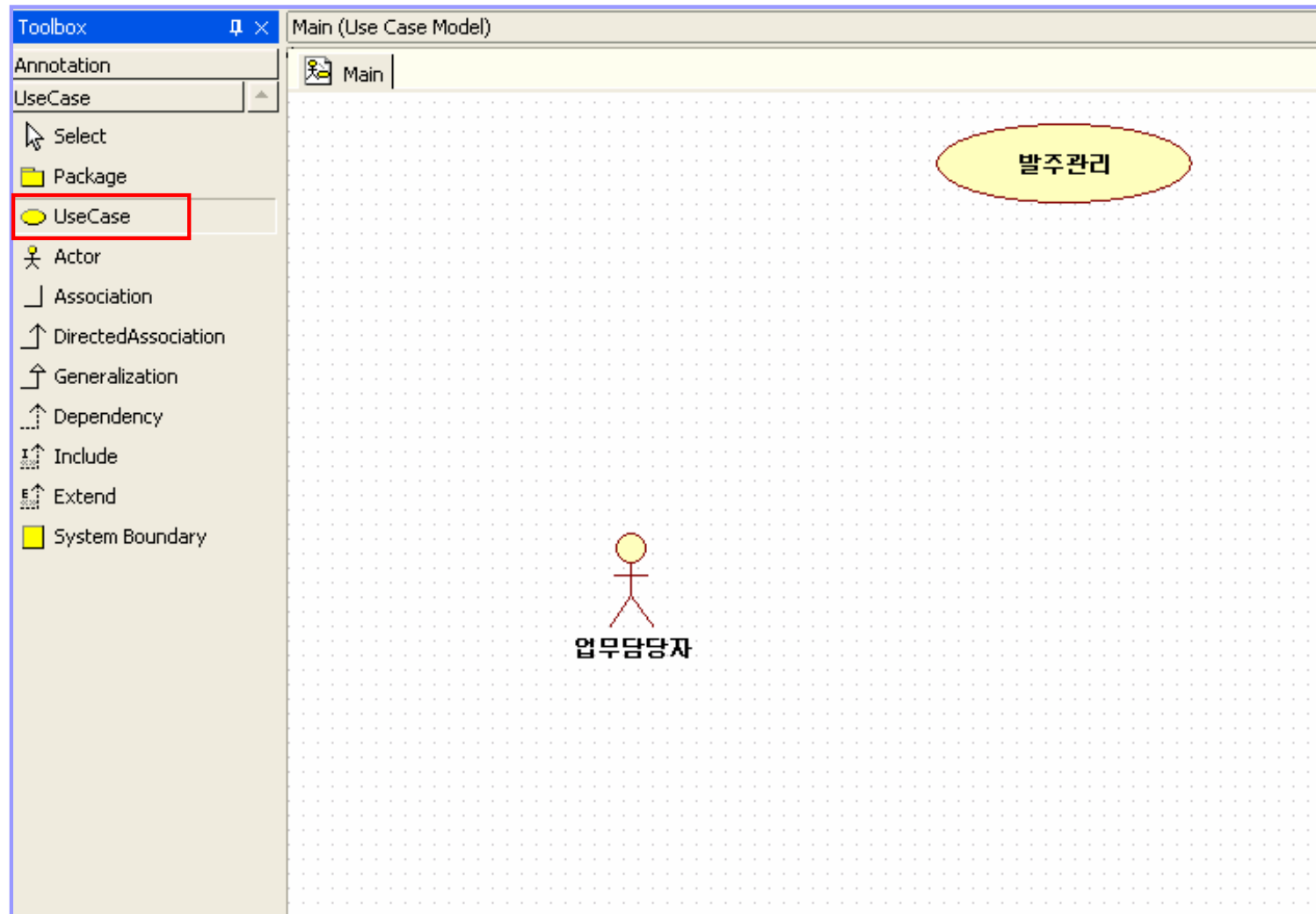
- 사용자인 업무담당자를 Actor로 표현합니다.
  - Tool Bar에서 Actor 선택 -> Main Window 영역에서 마우스 왼쪽 버튼 클릭 -> “업무담당자” 입력



## 4. 도구 활용 예제

### 4.2 Use Case Diagram (3/6)

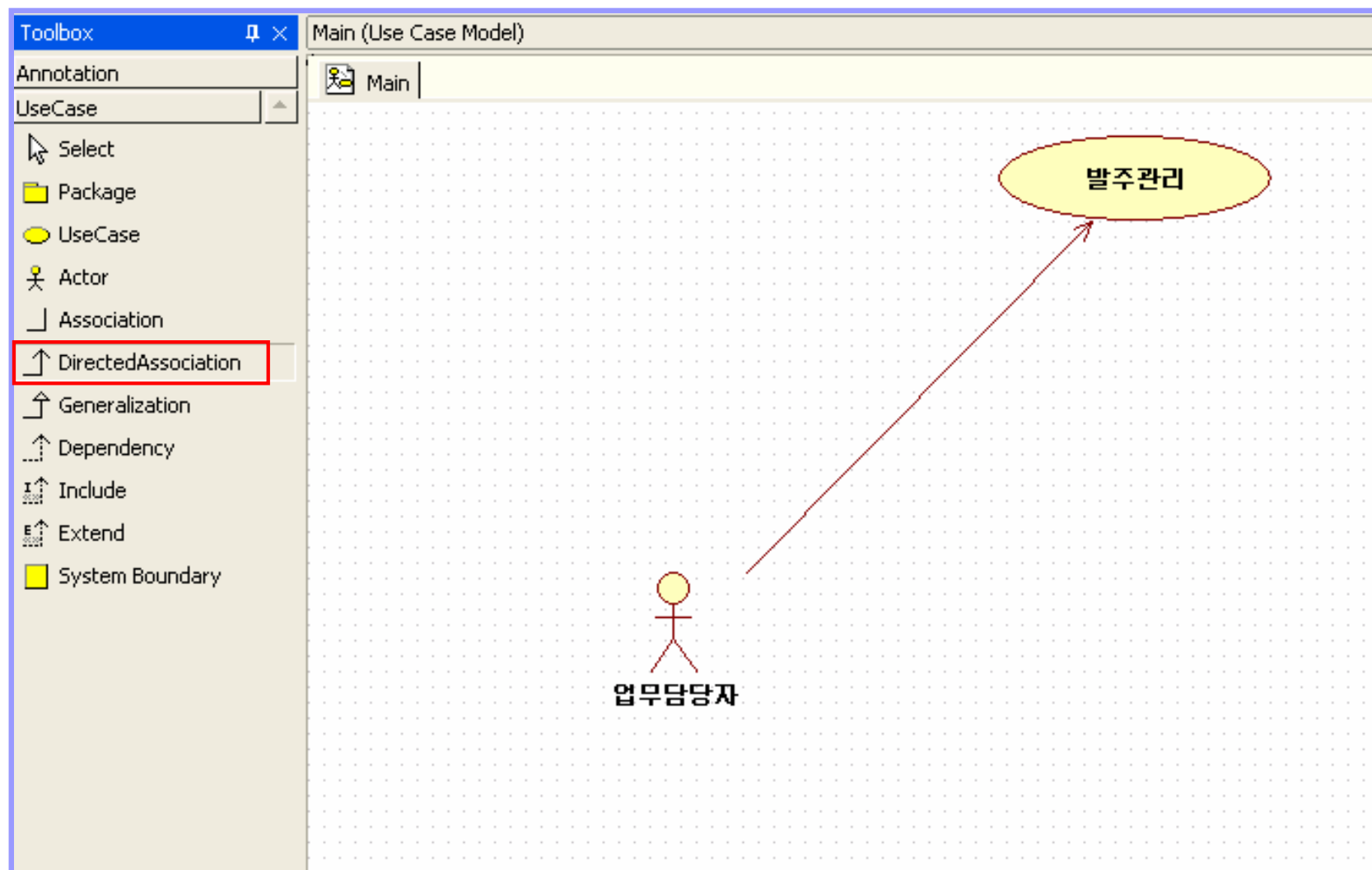
- 발주관리 기능을 UseCase로 표현합니다.
  - Tool Bar에서 UseCase 선택 -> Main Window 영역에서 마우스 왼쪽 버튼 클릭 -> “발주관리” 입력



## 4. 도구 활용 예제

### 4.2 Use Case Diagram [4/6]

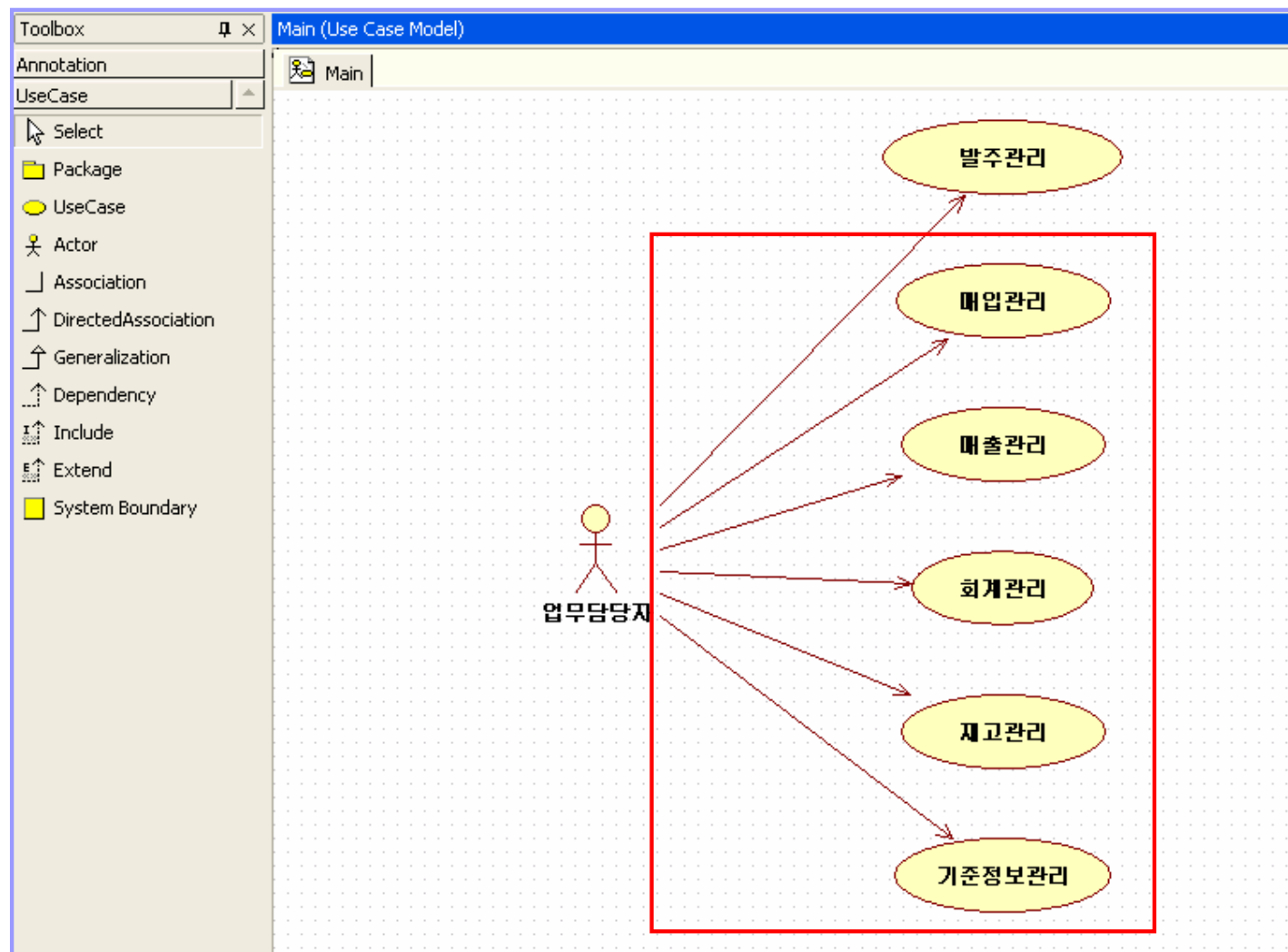
- 업무담당자 Actor와 발주관리 UseCase의 관계를 표현합니다.
  - Tool Bar에서 DirectedAssociation 선택 -> Main Window 영역에서 마우스 왼쪽 버튼 클릭



## 4. 도구 활용 예제

### 4.2 Use Case Diagram (5/6)

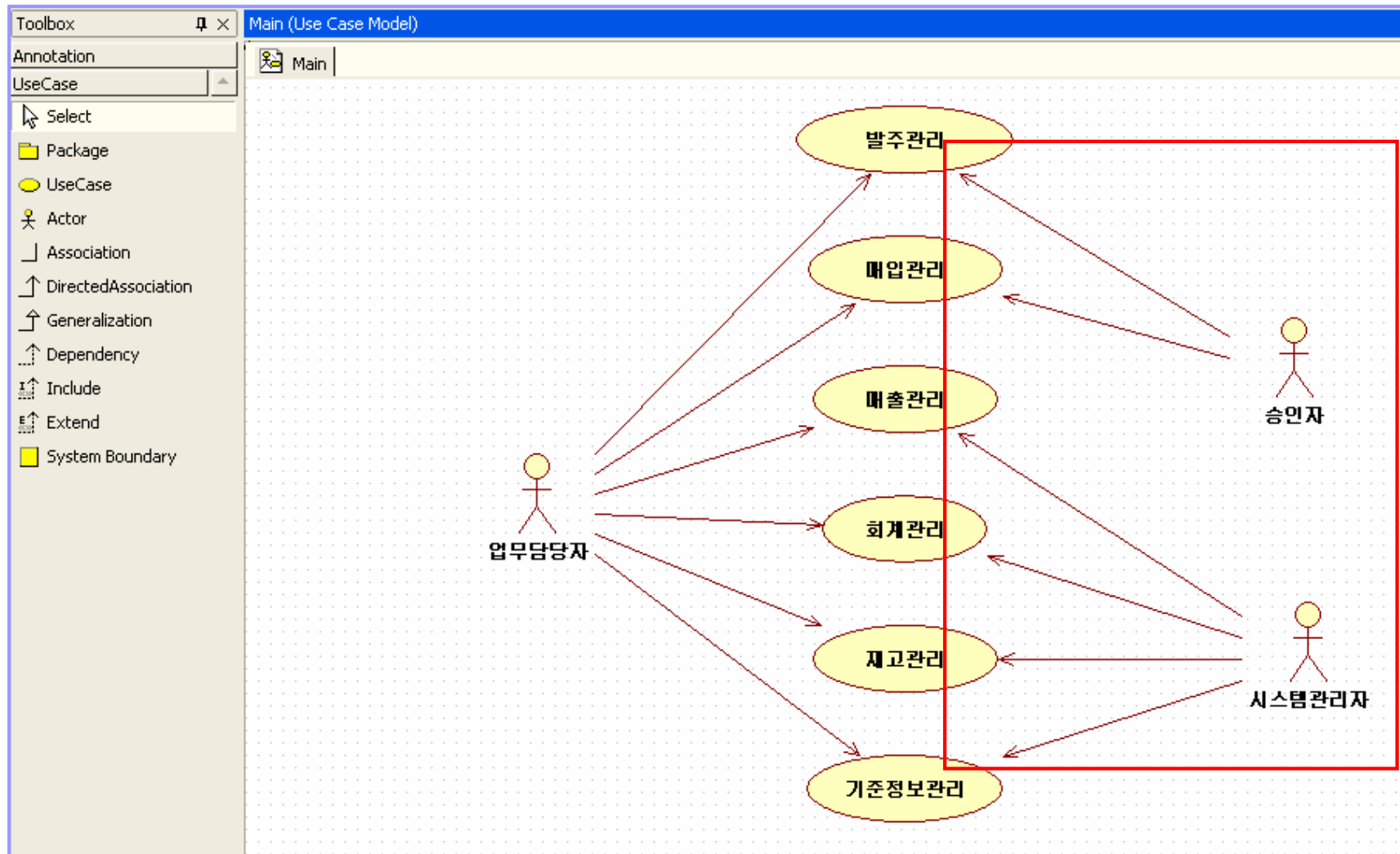
- 매입관리, 매출관리 기능들에 대해서도 앞서 확인한 바와 같이 UseCase를 추가하고, 관계를 표현합니다.



## 4. 도구 활용 예제

### 4.2 Use Case Diagram (6/6)

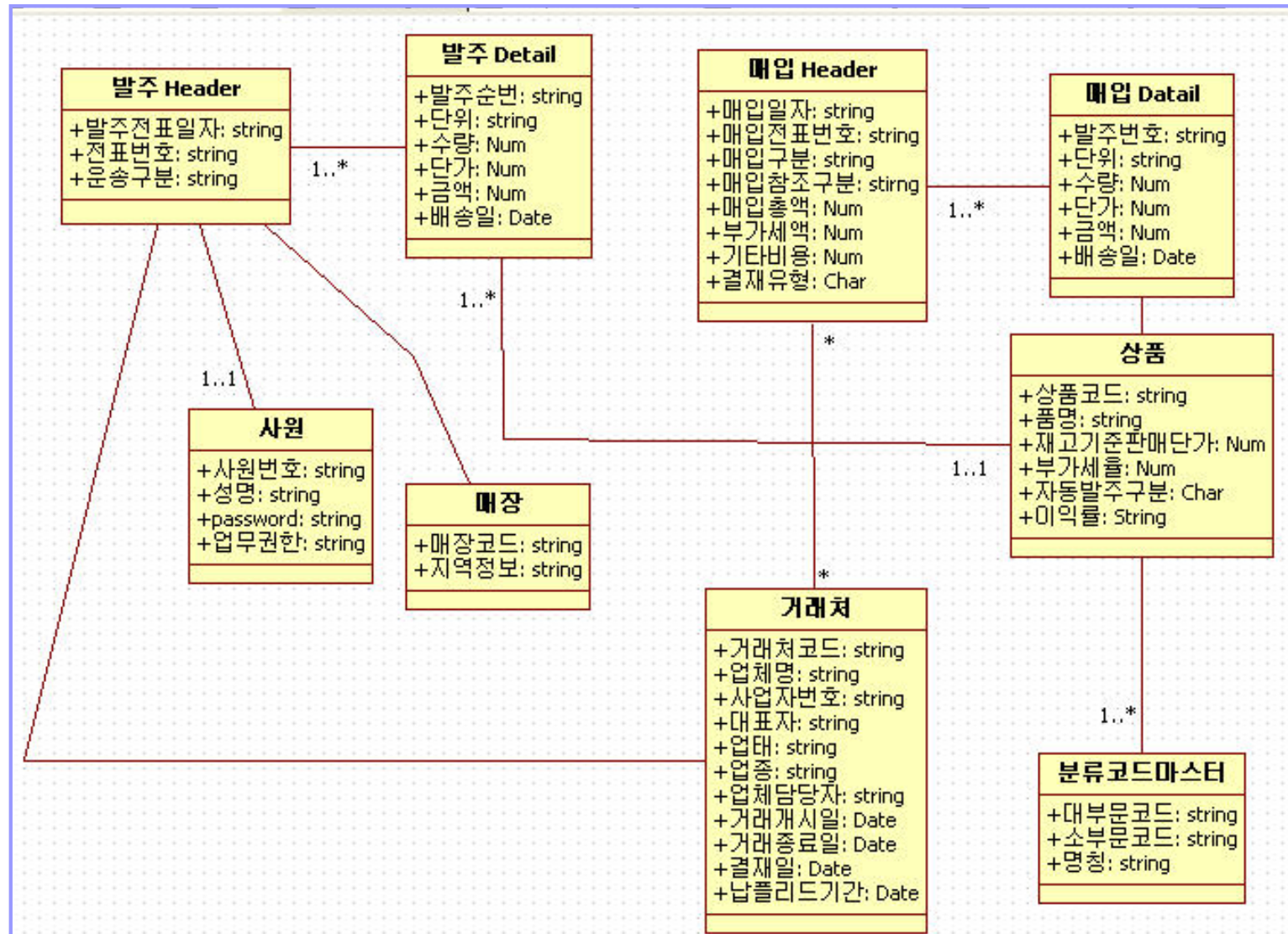
- 승인자와 시스템관리자 Actor를 생성하고, 관련 UseCase와의 관계를 통해 UseCase Diagram을 완성합니다.



## 4. 도구 활용 예제

### 4.3 Class Diagram

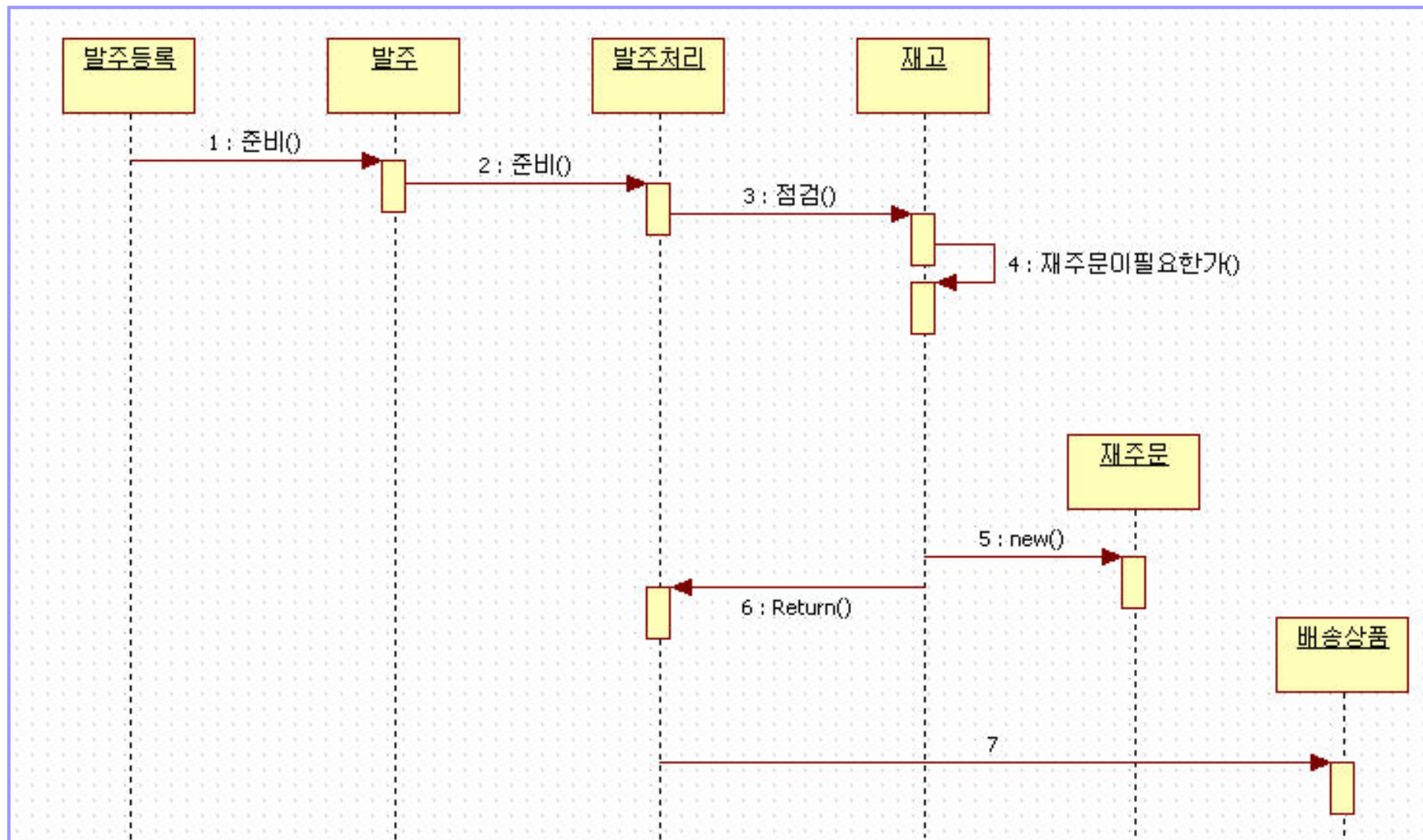
- 생산 관리 시스템 요구사항을 Class Diagram으로 표현하겠습니다.
  - 45 ~ 53p를 참조합니다.



## 4. 도구 활용 예제

### 4.4 Sequence Diagram

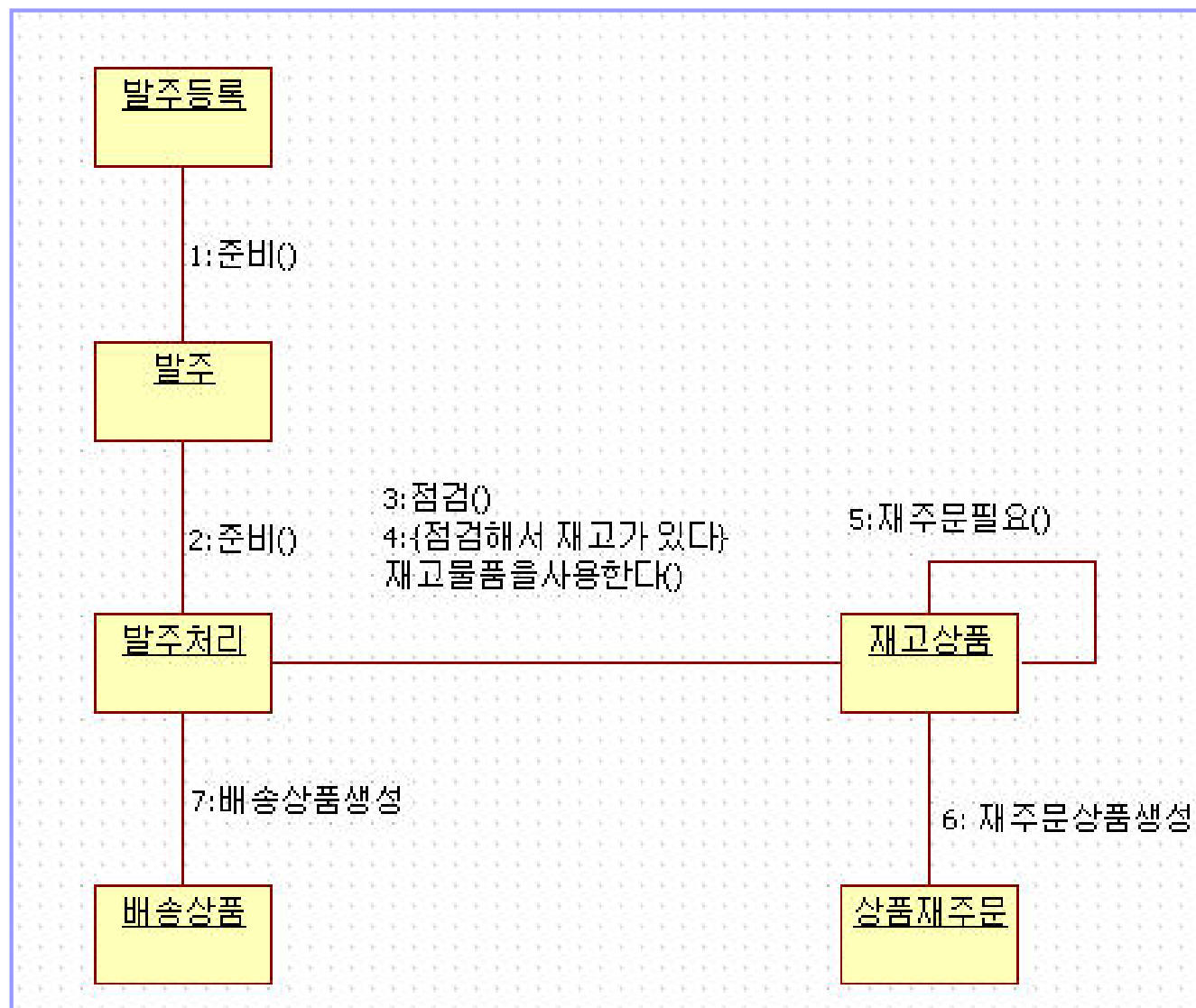
- 생산 관리 시스템 요구사항을 Sequence Diagram으로 표현하겠습니다.
  - 55 ~ 60p를 참조합니다.



## 4. 도구 활용 예제

### 4.5 Collaboration Diagram

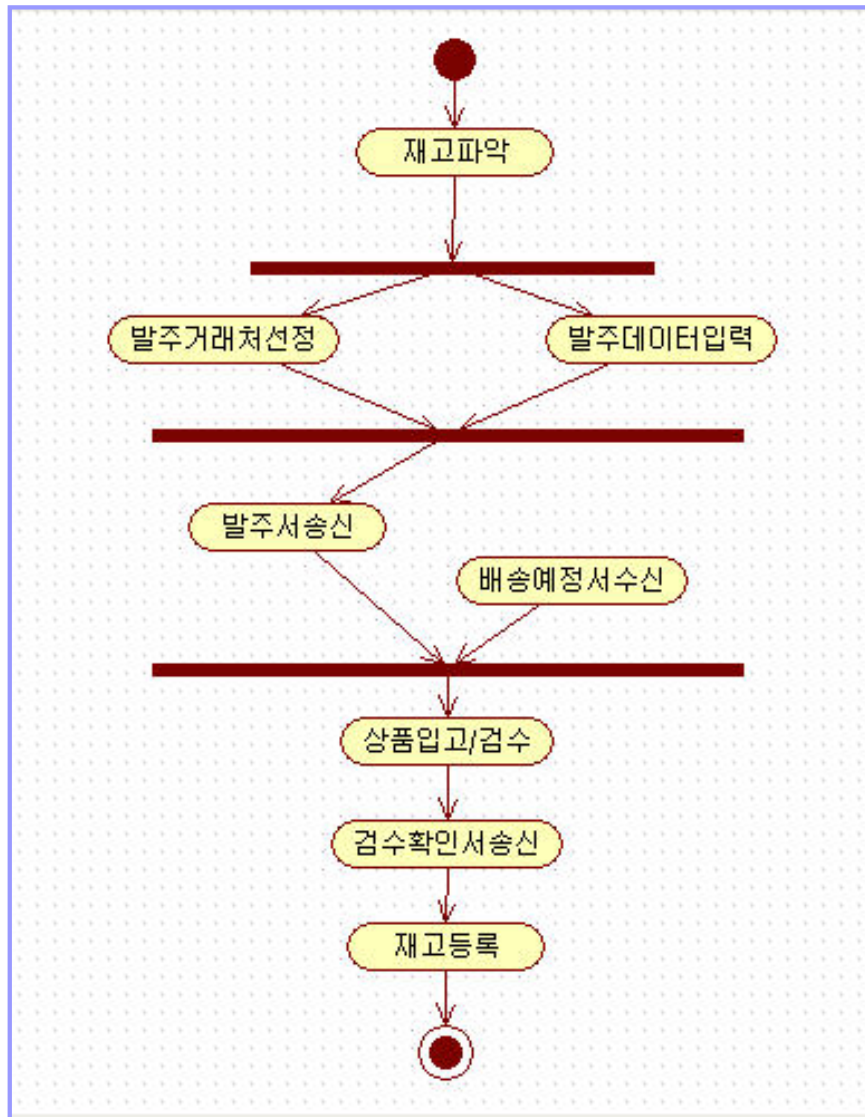
- 생산 관리 시스템 요구사항을 Collaboration Diagram으로 표현하겠습니다.
  - 80 ~ 85p를 참조합니다.



## 4. 도구 활용 예제

### 4.6 Activity Diagram

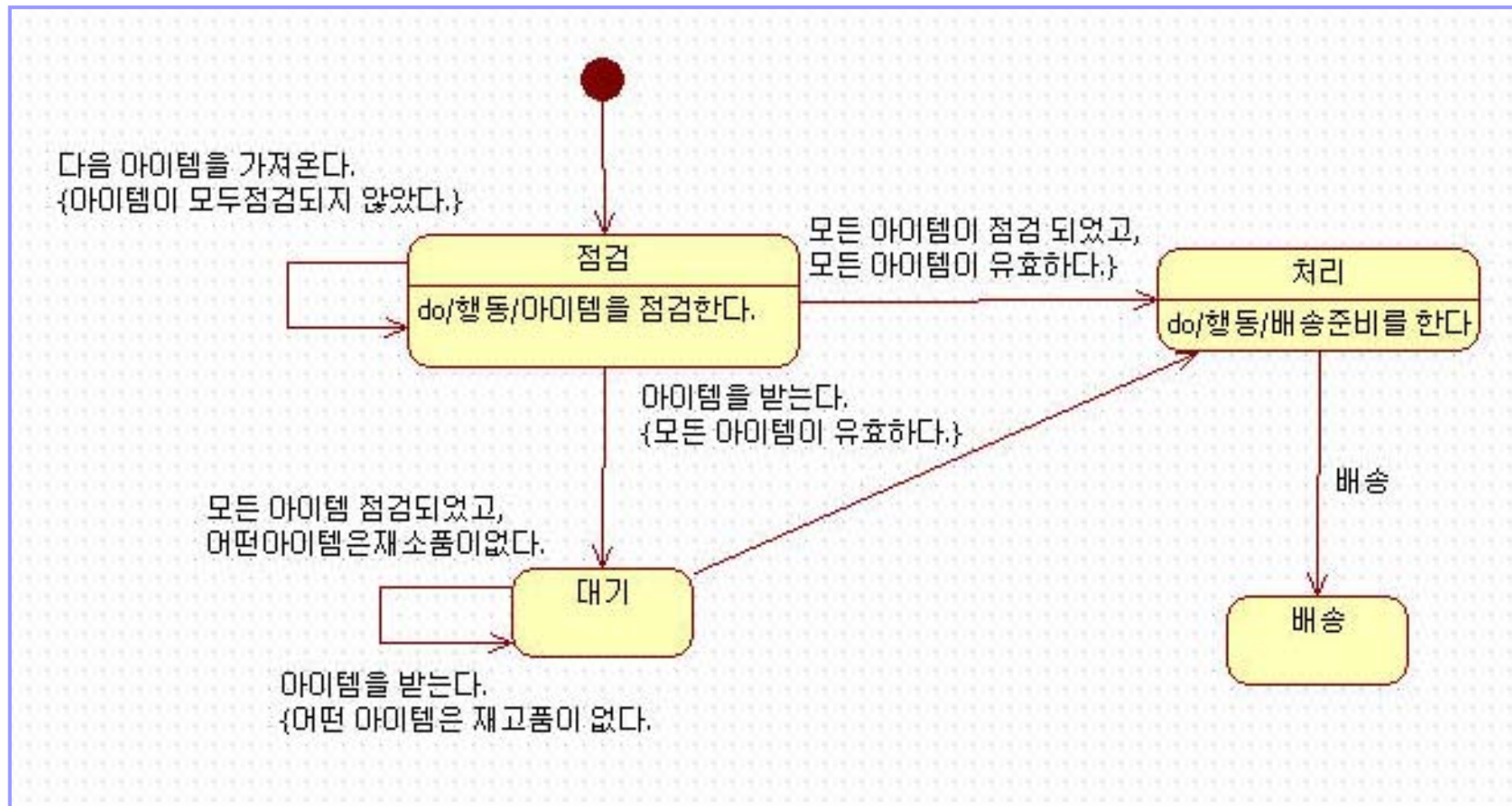
- 생산 관리 시스템 요구사항을 Activity Diagram으로 표현하겠습니다.
  - 62 ~ 69p 를 참조합니다.



## 4. 도구 활용 예제

### 4.7 Statechart Diagram

- 생산 관리 시스템 요구사항을 Statechart Diagram으로 표현하겠습니다.
  - 71 ~ 78p를 참조합니다.



## 5. FAQ

## 5. FAQ

질문1) StarUML은 eclipse와 연동되어 구동되는 프로그램인가요?



답1) 아닙니다. StarUML은 독립적으로 구동되는 유용한 프로그램입니다.

## 6. 도구 평가

## 6. 도구 평가

도구를 사용하기 위한 학습시간이 짧아 초보자도 쉽게 사용할 수 있습니다.

유료소프트웨어 못지 않은 다양한 UML 작성 기능을 가지고 있습니다.

Reverse Engineering을 지원합니다.

## 7. 용어집

## 7. 용어집

- 본 매뉴얼에서 사용하고 있는 용어가 정리되어 있습니다.

용어	설명
UML	Unified Modeling Language의 약자이며 요구분석, 시스템설계, 시스템 구현 등의 시스템 개발 과정에서, 개발자간의 의사소통을 원활하게 이루어지게 하기 위하여 표준화한 모델링 언어입니다.
Activity Diagram	처리 로직이나 조건에 따른 처리흐름을 순서에 따라 정의한 모델 입니다.
Class Diagram	시스템에서 사용되는 객체 타입(클래스)을 정의하고 그들 간에 존재하는 정적인 관계를 표현한 다이어그램입니다.
Object Diagram	특정 조건하에서 주요 객체들의 속성과 객체관계를 분석함으로써 클래스 모델의 완전성을 검증하는 모델입니다.
Sequence Diagram	문제해결에 필요한 객체를 정의하고 객체간 동적 상호관계를 시간순서에 따라 정의하는 모델입니다.
State Diagram	하나의 객체가 생성되어 소멸 될 때까지의 모든 상태를 분석하고 표현하는 모델입니다.
Use Case Diagram	사용자 관점에서 SW 시스템의 범위와 기능 정의하고, 시스템이 해야 할 무엇을 작성하는 모델입니다.