

AmaterasUML

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 AmaterasUML 다운로드
- 3. 도구 기능 소개
 - 3.1 AmaterasUML 주요 기능
 - 3.2 AmaterasUML 메뉴 소개
 - 3.3 예제 소개
 - 3.4 Eclipse
 - 3.5 Class Diagram
 - 3.6 Use Case Diagram
 - 3.7 Activity Diagram
- 4. 도구 활용 예제
 - 4.1 예제 설명
 - 4.2 Class Diagram
 - 4.3 Use Case Diagram
 - 4.4 Activity Diagram
- 5. FAQ
- 6. 도구 평가
- 7. 용어집

1. 도구개요

1. 도구 개요






소개	AmaterasUML 도구는 Class Diagram, Sequence Diagram 등의 UML Diagram 작성을 도와줍니다.		
주요 기능	Class Diagram, Sequence Diagram, Use Case Diagram, Activity Diagram 작성		
카테고리	Design-Modeling	세부카테고리	설 계 & 모델링
커버리지	Design-Modeling	도구 난이도	초급
라이선스 형태 / 비용	EPL (Eclipse Public License) / 무료	사전 설치 도구	EMF, GEF, UML2
사용환경	운영체제	Platform Independent	
	Eclipse 환경	Eclipse 3.3	
특징	• Eclipse의 plug-in으로 설치가 용이합니다. • UML Diagram 작성이 용이하도록 도와줍니다.		
적용 회사 / 프로젝트	-		
관련 도구	ArgoUML, Umlet, Violet UML Editor, StarUML, Papyrus		
제작사	-		
공식 홈페이지	http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasUML		
개발자	Takezoe, rinoa		

2. 도구 설치 방법

2. 도구 설치 방법

2.1 AmaterasUML 다운로드 (1/4)

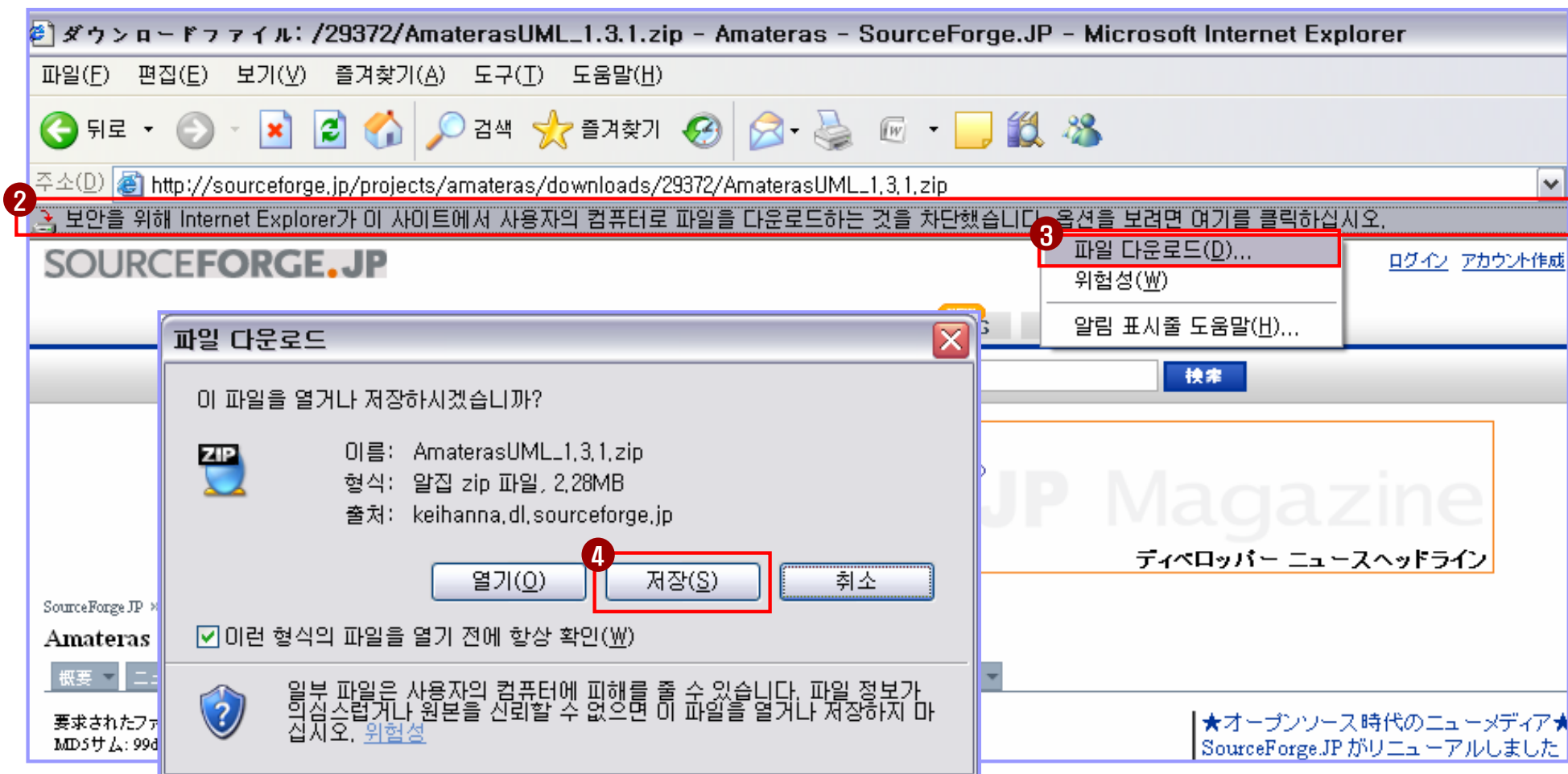
- 다음의 사이트에서 AmaterasUML을 다운 받습니다.
 - <http://sourceforge.jp/projects/amateras/releases>
 - 본 매뉴얼은 AmaterasUML에 대한 매뉴얼이므로 AmaterasUML 1.3.1을 다운로드 하겠습니다.
 - 사이트 접속 -> AmaterasUML_1.3.1.zip 클릭

AmaterasUML		
▼ 1.3.1 - [リリースノート / 変更履歴]		
2008-02-14 10:50		
1  AmaterasUML_1.3.1.zip	2.3 MB	36730
▼ 1.3.0 - [リリースノート / 変更履歴]		
2007-10-07 23:12		
 AmaterasUML_1.3.0.1.zip	2.3 MB	22898
▼ 1.2.2 - [リリースノート / 変更履歴]		
2007-04-22 21:35		
 AmaterasUML_1.2.2.zip	2.3 MB	31033
▼ 1.2.1 - [リリースノート / 変更履歴]		
2006-08-31 17:59		
 AmaterasUML_1.2.1.zip	2.2 MB	40006
▼ 1.2.0 - [リリースノート / 変更履歴]		
2006-06-10 15:20		
 AmaterasUML1_2_0.zip	2.2 MB	20956

2. 도구 설치 방법

2.1 AmaterasUML 다운로드 (2/4)

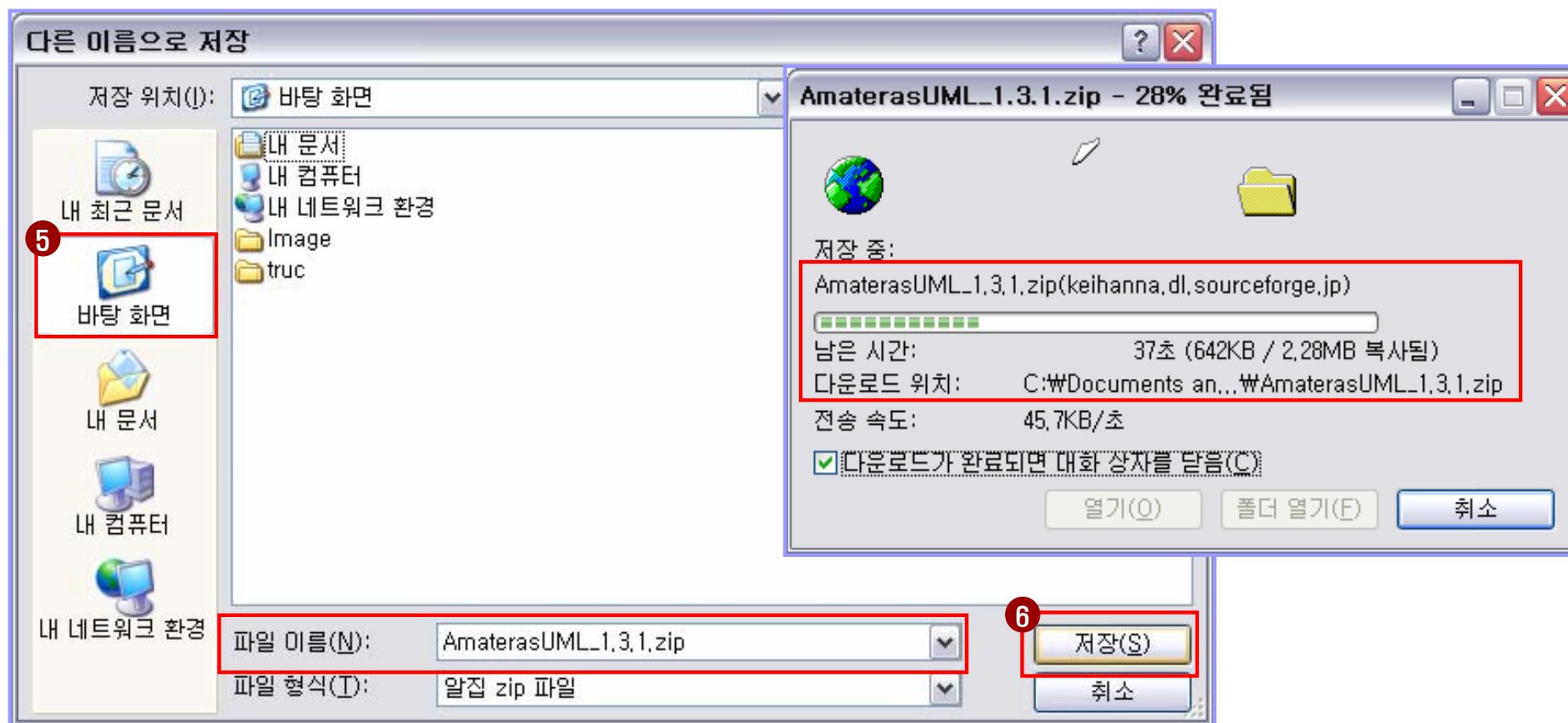
- 보안을 위해 Internet Explorer 에서 다음처럼 다운로드를 차단하는 경우가 있습니다. 차단을 해제하고 파일 다운로드를 클릭하여 파일을 다운받습니다.
 - 다운로드 차단 메시지 클릭 -> 파일 다운로드 -> 파일 다운로드 창에서 저장 버튼 클릭



2. 도구 설치 방법

2.1 AmaterasUML 다운로드 (3/4)

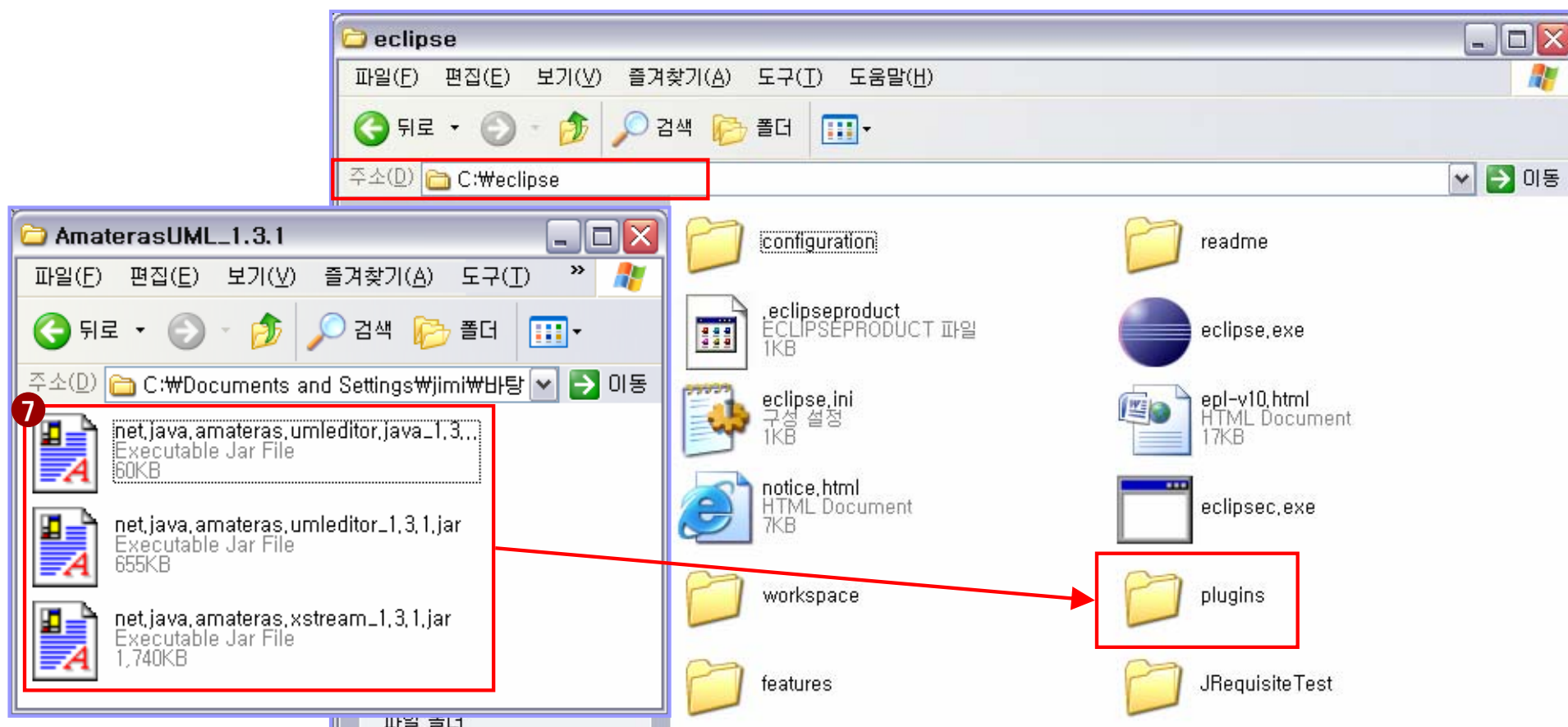
- 다운로드를 진행합니다.
 - AmaterasUML_1.3.1.zip 파일을 다운로드합니다.
 - 본 매뉴얼에서는 바탕화면에 다운받도록 하겠습니다.
 - 저장 위치 선택 -> 저장 버튼 클릭 -> 다운로드 진행



2. 도구 설치 방법

2.1 AmaterasUML 다운로드 (4/4)

- 다운로드한 AmaterasUML_1.3.1.zip 파일의 압축을 풀고 폴더 안의 .jar 파일을 Eclipse가 설치되어 있는 폴더 내의 plugins 폴더로 이동합니다.
 - 본 매뉴얼에서는 Eclipse가 C:\Weclipse/ 에 설치되어 있습니다.
 - AmaterasUML_1.3.1 폴더 안의 .jar 파일 선택 -> C:\Weclipse\plugins 폴더로 드래그



3. 도구 기능 소개

3. 도구 기능 소개

세부 목차

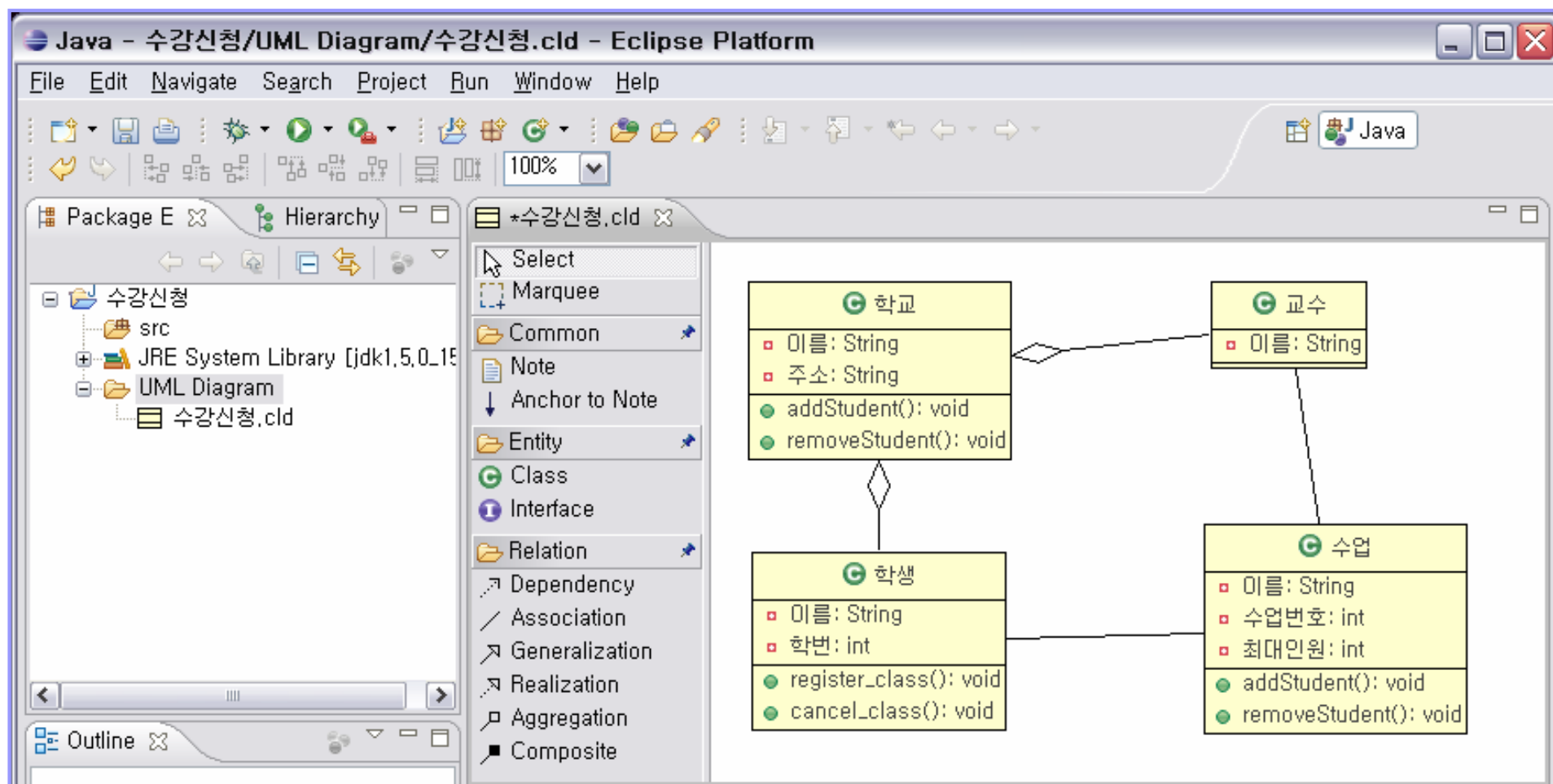
- 3.1 AmaterasUML 주요 기능
- 3.2 AmaterasUML 메뉴 소개
- 3.3 예제 소개
- 3.4 Eclipse
- 3.5 Class Diagram
- 3.6 Use Case Diagram
- 3.7 Activity Diagram

3. 도구 기능 소개

3.1 AmaterasUML 주요 기능 (1/3)

- Class Diagram

- 클래스와 인터페이스 사이의 관계를 나타낸 Class Diagram을 작성할 수 있습니다.

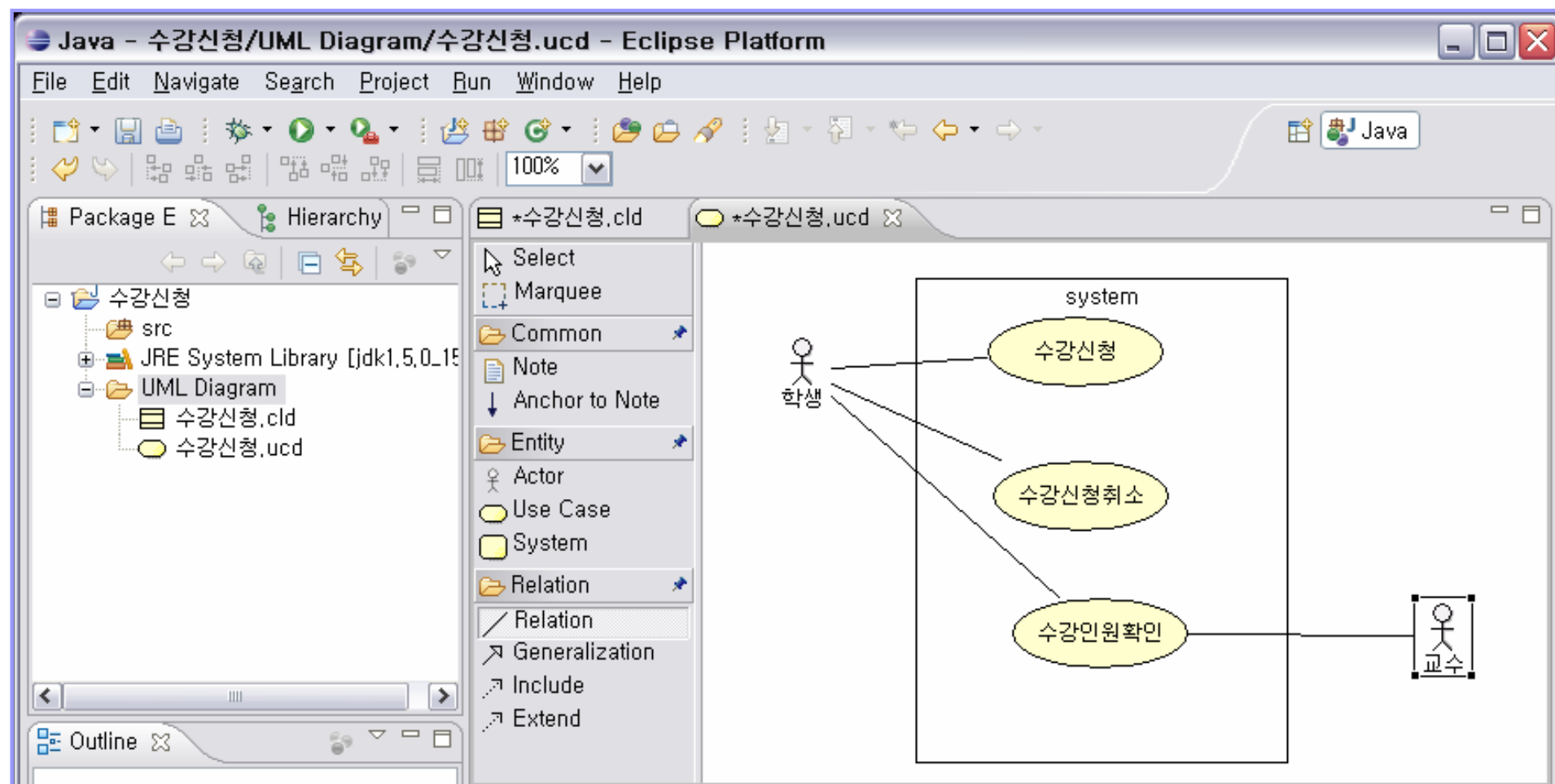


3. 도구 기능 소개

3.1 AmaterasUML 주요 기능 (2/3)

- Use Case Diagram

- 사용자와 행위의 관계를 표현해 주는 Use Case Diagram을 작성할 수 있습니다.

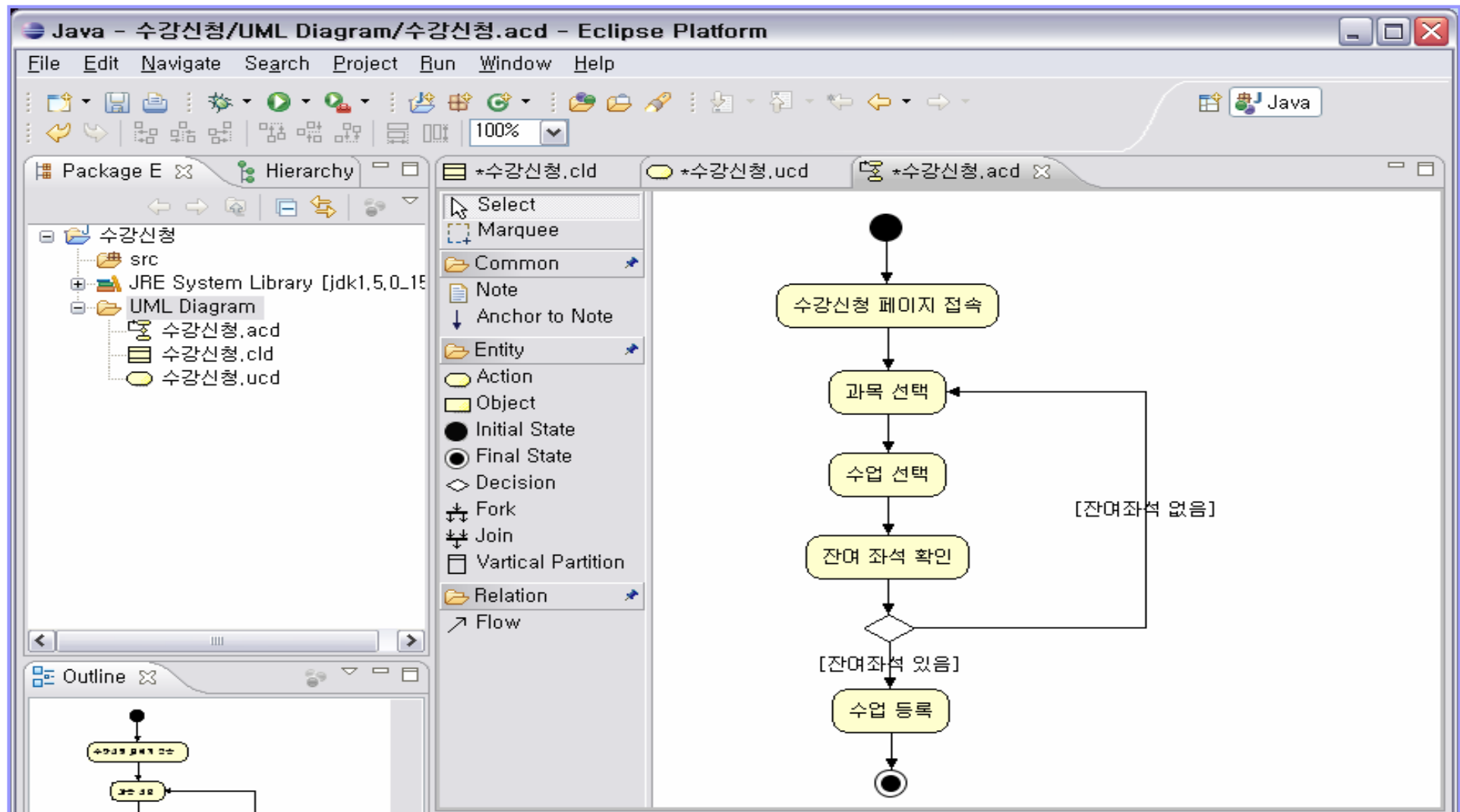


3. 도구 기능 소개

3.1 AmaterasUML 주요 기능 (3/3)

- Activity Diagram

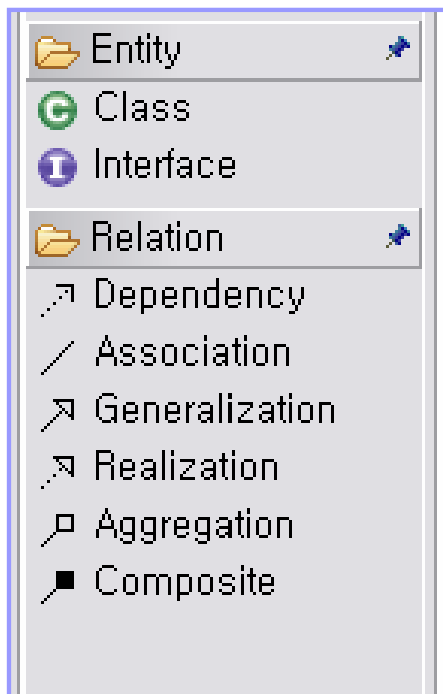
- 프로세스의 업무 절차를 표현하기 위한 Activity Diagram을 작성할 수 있습니다.



3. 도구 기능 소개

3.2 AmaterasUML 메뉴 소개 - 팔레트 (1/3)

- AmaterasUML의 Class Diagram을 실행하면 오른쪽에 다음과 같은 팔레트가 보입니다.

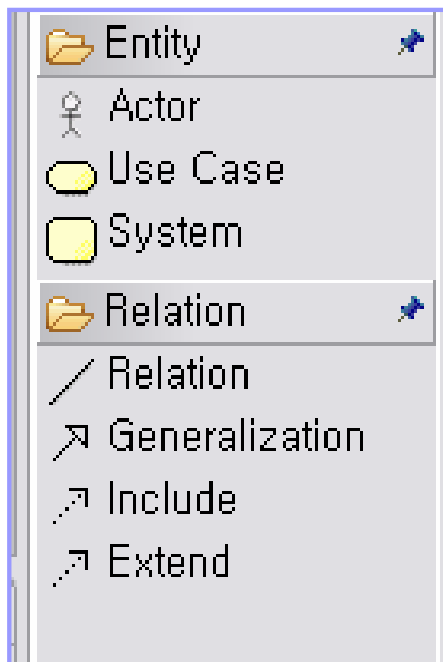


Entity	
Class	속성과 함수의 집합체
Interface	추상 메소드와 상수만으로 구성된 추상 클래스
Relation	
Dependency	의존관계. 두 사물간의 의미적 관계
Association	연관관계. 지속적으로 유지되는 구조적 관계
Generalization	상속관계. 일반화된 사물간의 관계
Realization	실체화관계. 분류자간의 의미적 관계
Aggregation	집합연관. 전체와 부분간의 구조적 관계
Composite	합성연관. 사물간의 포함관계

3. 도구 기능 소개

3.2 AmaterasUML 메뉴 소개 - 팔레트 (2/3)

- AmaterasUML의 Use Case Diagram을 실행하면 오른쪽에 다음과 같은 팔레트가 보입니다.



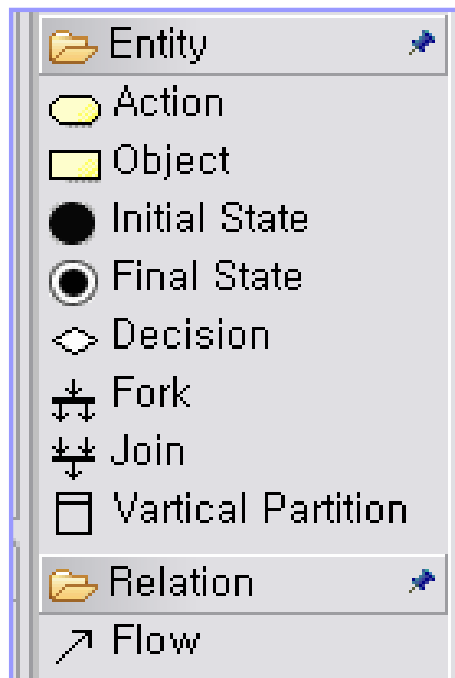
Entity	
Actor	사용자. 시스템 외부에 존재하는 개체
Use Case	사용사례. 액터가 수행해야 하는 일련의 행동들
System	시스템. 사용사례를 포함하는 시스템

Relation	
Relation	사용자와 사용사례간의 관계
Generalization	사용사례간의 상속관계
Include	여러 사용사례에게 중복적으로 나타나는 기능들을 별도의 사용사례로 추출하여 관련된 사용사례와 연결
Extend	특정한 사용사례 내에서 사용사례 내의 흐름이 특정 시점에서 여러가지 형태로 분류될 경우

3. 도구 기능 소개

3.2 AmaterasUML 메뉴 소개 - 팔레트 (3/3)

- AmaterasUML의 Activity Diagram을 실행하면 오른쪽에 다음과 같은 팔레트가 보입니다.



Entity	
Action	행위나 작업
Object	객체가 가질 수 있는 조건이나 상황
Initial State	객체의 상태변화가 시작되는 곳
Final State	객체의 상태변화가 종료되는 곳
Decision	논리식의 결과 값에 따라 분기가 일어남
Fork	병행처리가 시작되는 곳
Join	병행처리가 모이는 곳
Vertical Partition	역할에 따른 구분이나 조직에 따른 구분

Relation	
Flow	하나의 상태에서 다른 상태로 변화

3. 도구 기능 소개

3.3 예제 소개

- AmaterasUML의 Diagram 작성 기능을 설명하기 위해 A대학의 수강신청 예제를 사용합니다.
 - 수강신청 프로그램의 요구사항은 다음과 같습니다.

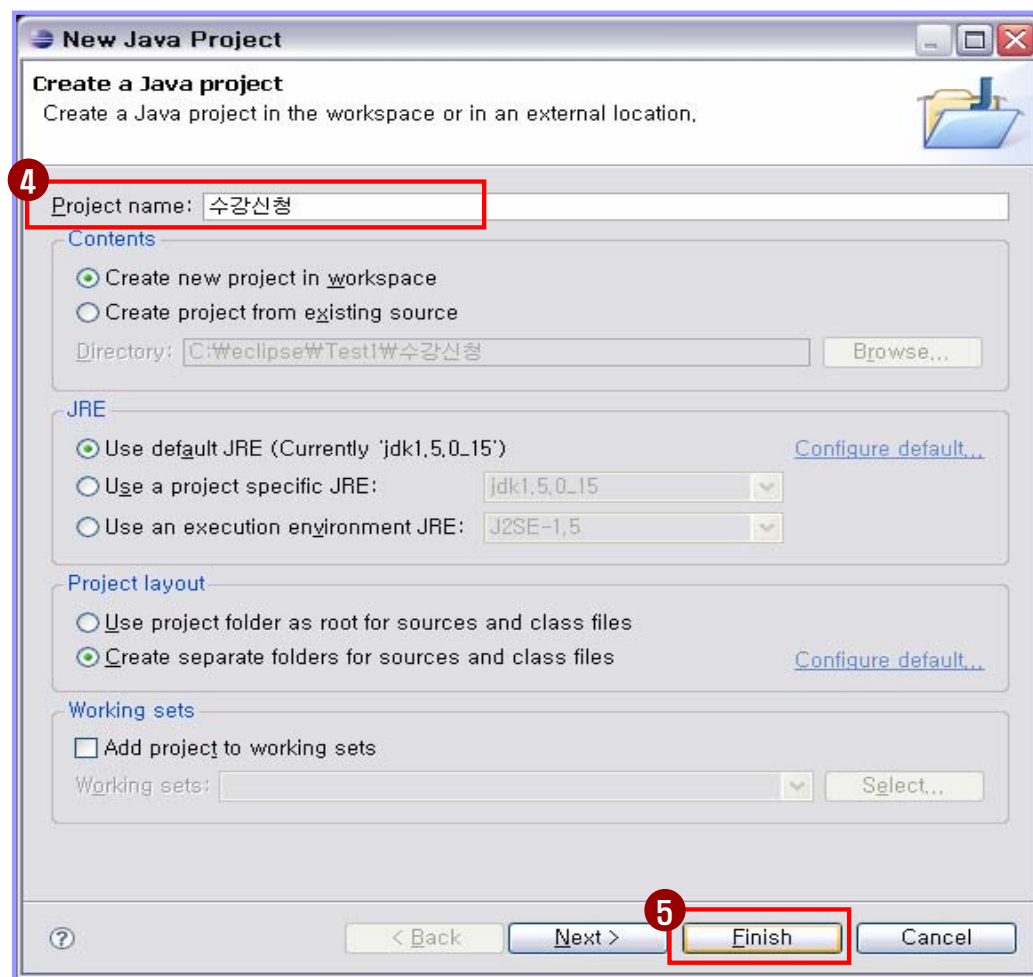
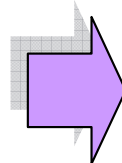
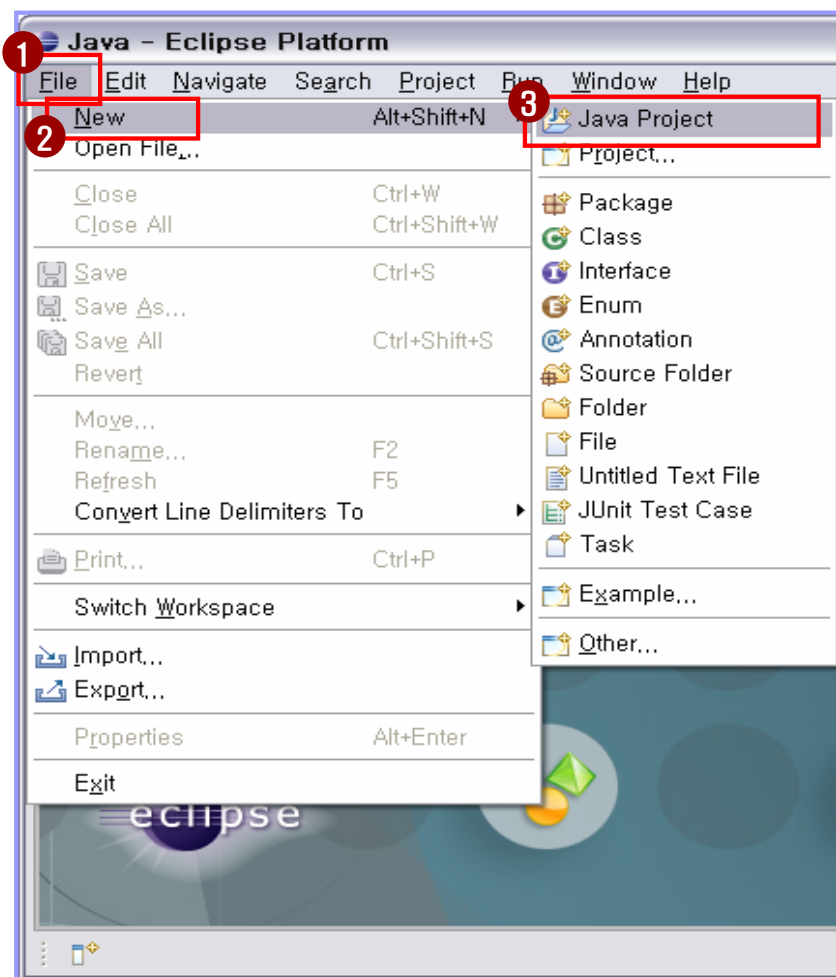
▶ 요구사항

- A대학은 다수의 학생 및 교수로 구성되어 있습니다.
- 한 학생은 여러수업을 수강할 수 있고 한 수업은 여러명의 학생을 수용할 수 있습니다.
- 수강신청 기간에 학생들은 정해진 학점을 수강신청 할 수 있습니다.
- 수강신청 프로그램에 접속하여 과목을 선택하면 그 과목의 수업시간표를 확인할 수 있습니다.
- 수업시간표에서 수강할 수업을 선택하고 잔여좌석을 확인합니다.
- 잔여좌석이 있는 경우 수업에 등록할 수 있고, 잔여좌석이 없는 경우에는 과목 선택 화면으로 되돌아갑니다.
- 수강신청한 과목은 기간내에 취소할 수 있습니다.
- 학생과 교수는 수강인원을 확인할 수 있는 권한이 있습니다.

3. 도구 기능 소개

3.4 Eclipse - 프로젝트 생성

- Eclipse에서 예제로 사용할 프로젝트를 생성합니다.
 - AmaterasUML을 실행하기 전 예제로 사용할 수강신청 프로그램에 대한 프로젝트를 생성합니다.
 - File -> New -> Java Project 선택
 - New Java Project 창에서 Project name 입력 -> Finish 버튼 클릭

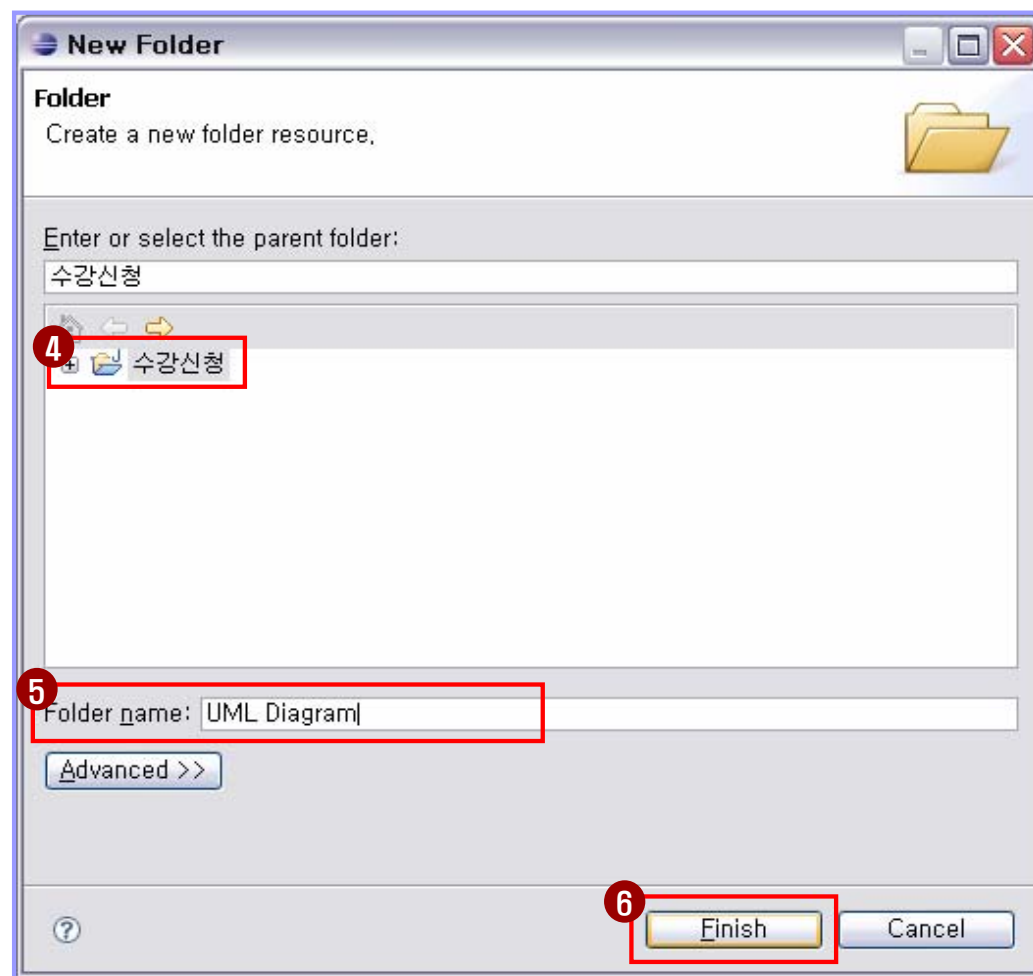
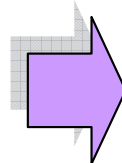
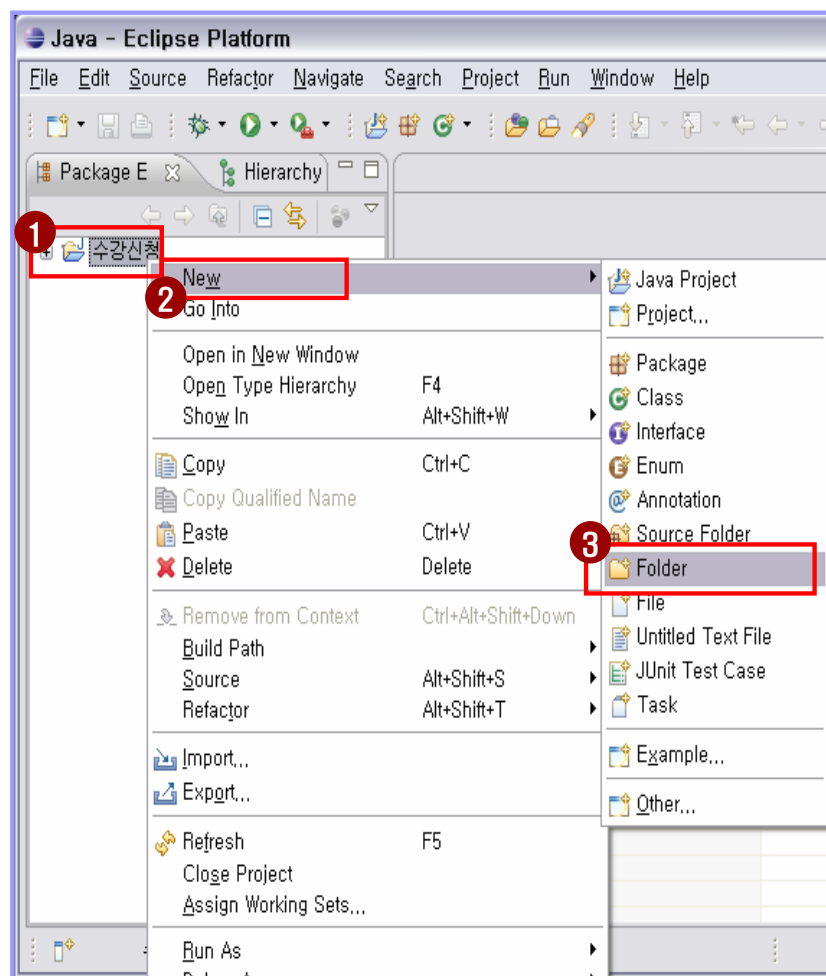


3. 도구 기능 소개

3.4 Eclipse - 폴더 생성

- 프로젝트에 새로운 폴더를 생성합니다.

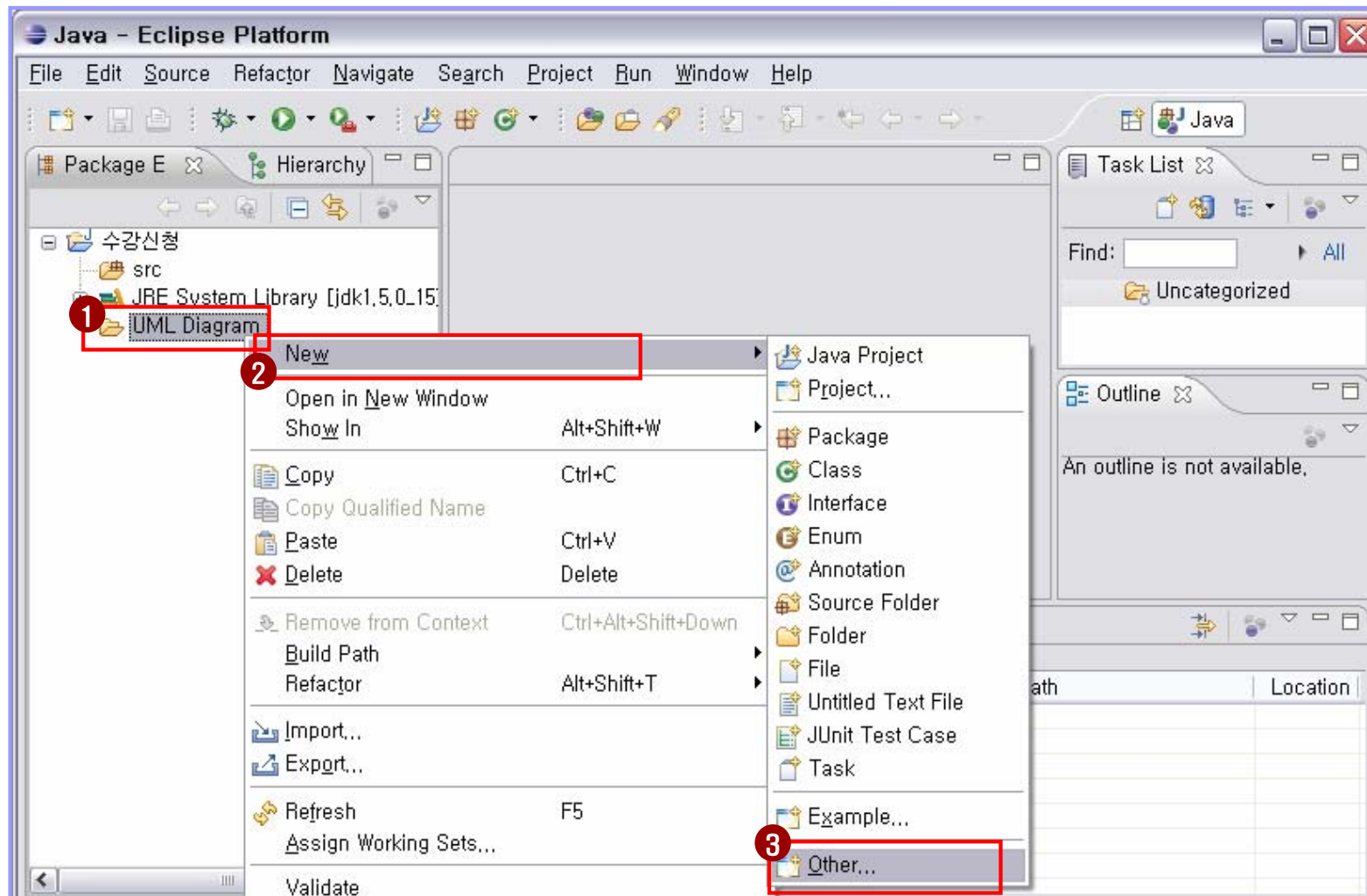
- 프로젝트 내에 다이어그램을 저장하기 위한 폴더를 새로 생성하겠습니다.
 - Package Explorer의 수강신청 project 에서 오른쪽 버튼 클릭 -> New -> Folder 선택
 - New Folder 창에서 수강신청 project 선택 -> Folder name을 UML Diagram으로 설정 -> Finish 버튼 클릭



3. 도구 기능 소개

3.5 Class Diagram - 다이어그램 생성 (1/3)

- UML Diagram 파일을 생성합니다.
 - 수강신청 프로젝트의 UML Diagram 폴더에 다이어그램을 작성하기 위한 UML 파일을 생성하겠습니다.
 - UML Diagram 폴더 오른쪽 클릭 -> New -> Other... 클릭



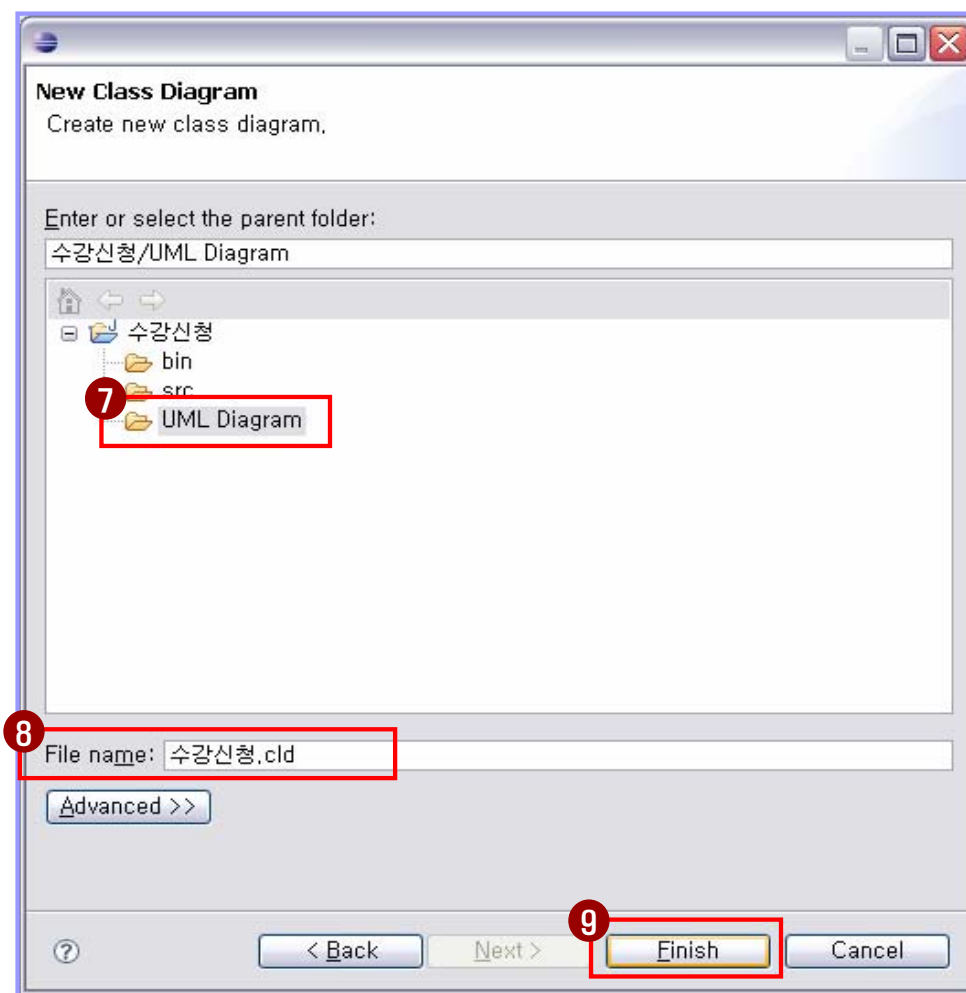
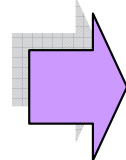
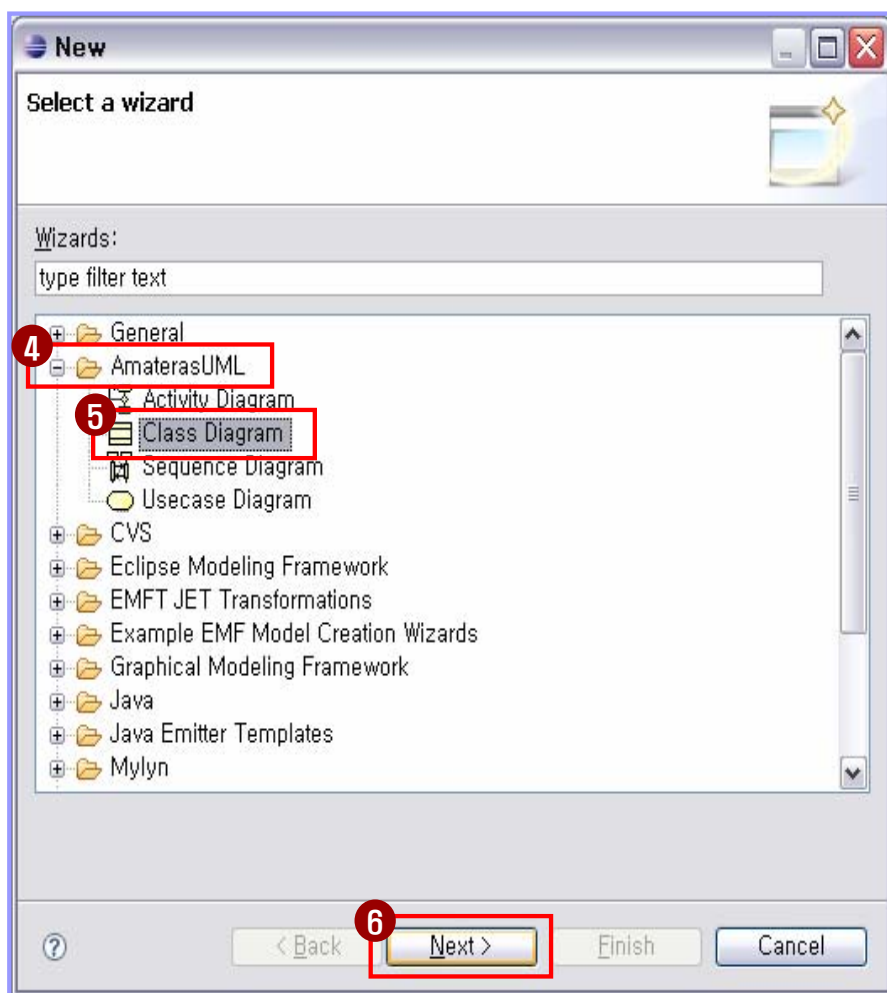
3. 도구 기능 소개

3.5 Class Diagram - 다이어그램 생성 (2/3)

- Class Diagram을 생성합니다.

- 확장자가 cld(Class Diagram)인 파일을 생성합니다.

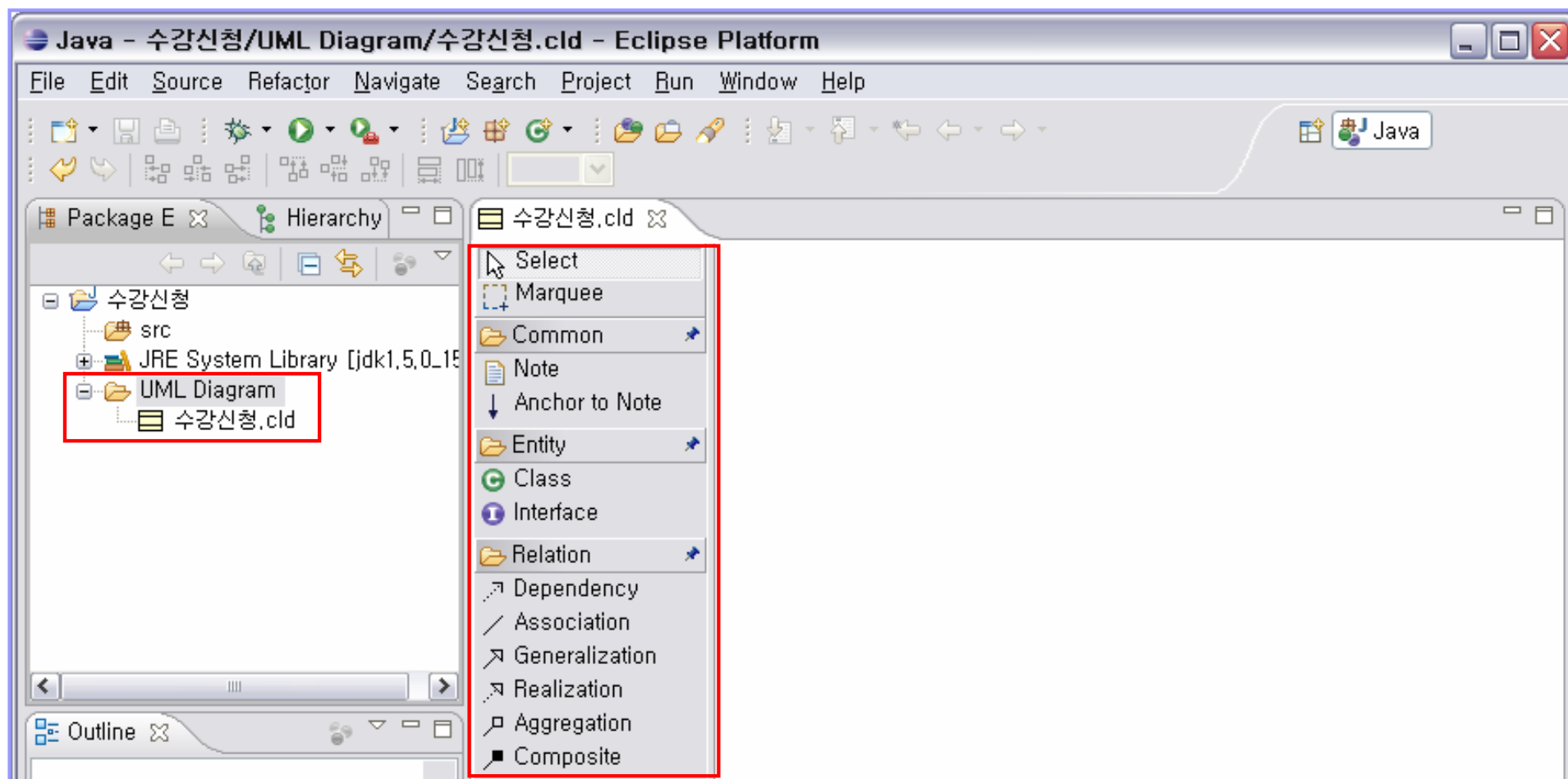
- AmaterasUML 하위 메뉴인 Class Diagram 클릭 -> Next 버튼 클릭
- UML Diagram 폴더 선택 -> File name 입력 -> Finish 버튼 클릭



3. 도구 기능 소개

3.5 Class Diagram - 다이어그램 생성 (3/3)

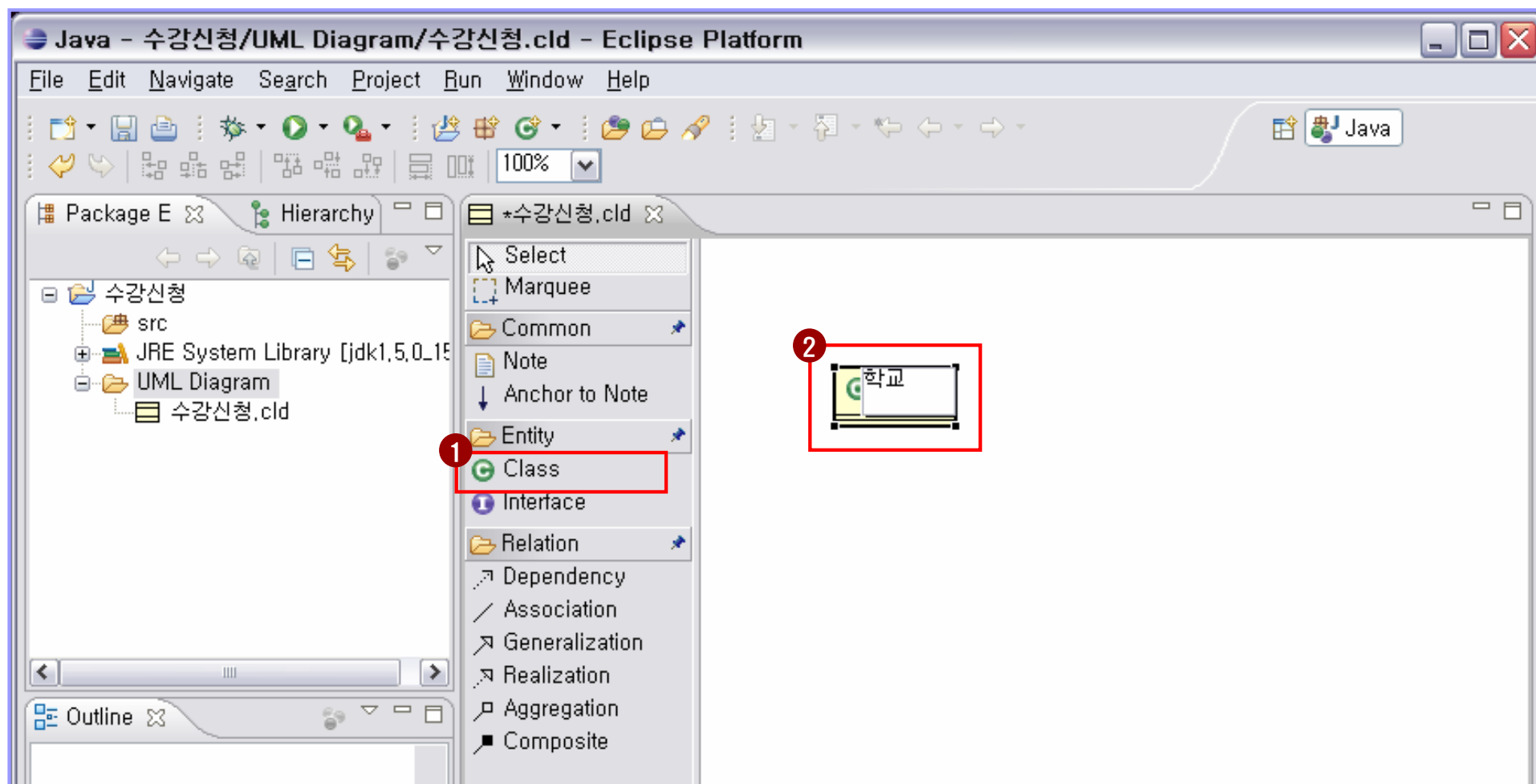
- Class Diagram이 생성되었습니다.
 - 수강신청.cld 파일이 생성되었습니다.
 - Class Diagram과 관련된 팔레트를 확인할 수 있습니다.



3. 도구 기능 소개

3.5 Class Diagram - 클래스 생성 (1/2)

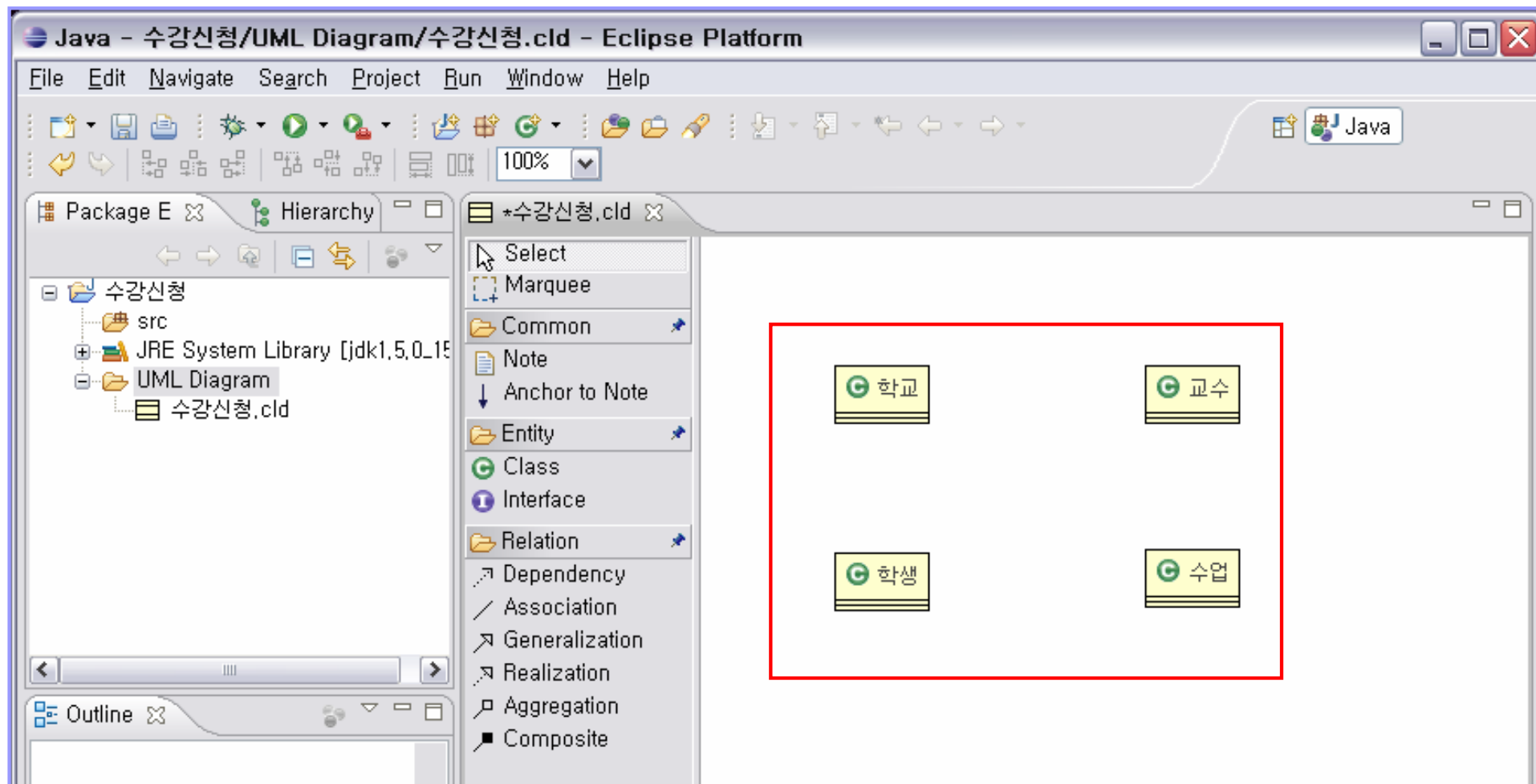
- 클래스를 생성합니다.
 - 팔레트를 사용하여 클래스를 생성합니다.
 - 팔레트에서 Class 클릭 -> 가운데의 작업공간 클릭 -> 생성된 Class 클릭 -> Class 이름 변경



3. 도구 기능 소개

3.5 Class Diagram - 클래스 생성 (2/2)

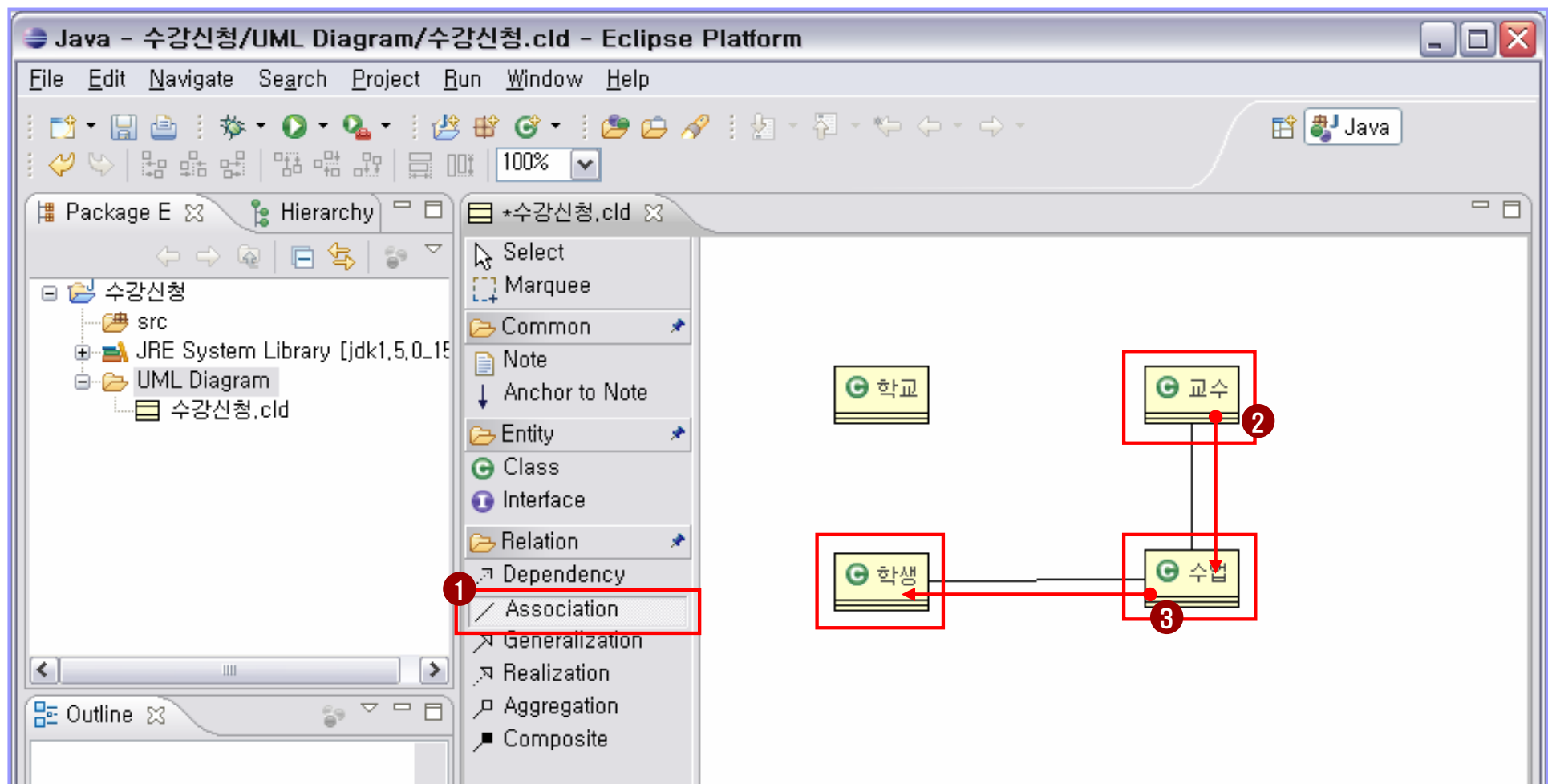
- 이전 페이지와 같은 방법으로 클래스를 생성합니다.



3. 도구 기능 소개

3.5 Class Diagram - 관계 생성 (1/2)

- 연관관계를 생성합니다.
 - 팔레트를 사용하여 연관관계를 생성합니다.
 - 팔레트에서 Association 클릭 -> 교수 Class에서 수업 Class로 클릭 앤 드래그 -> 수업 Class에서 학생 Class로 클릭 앤 드래그



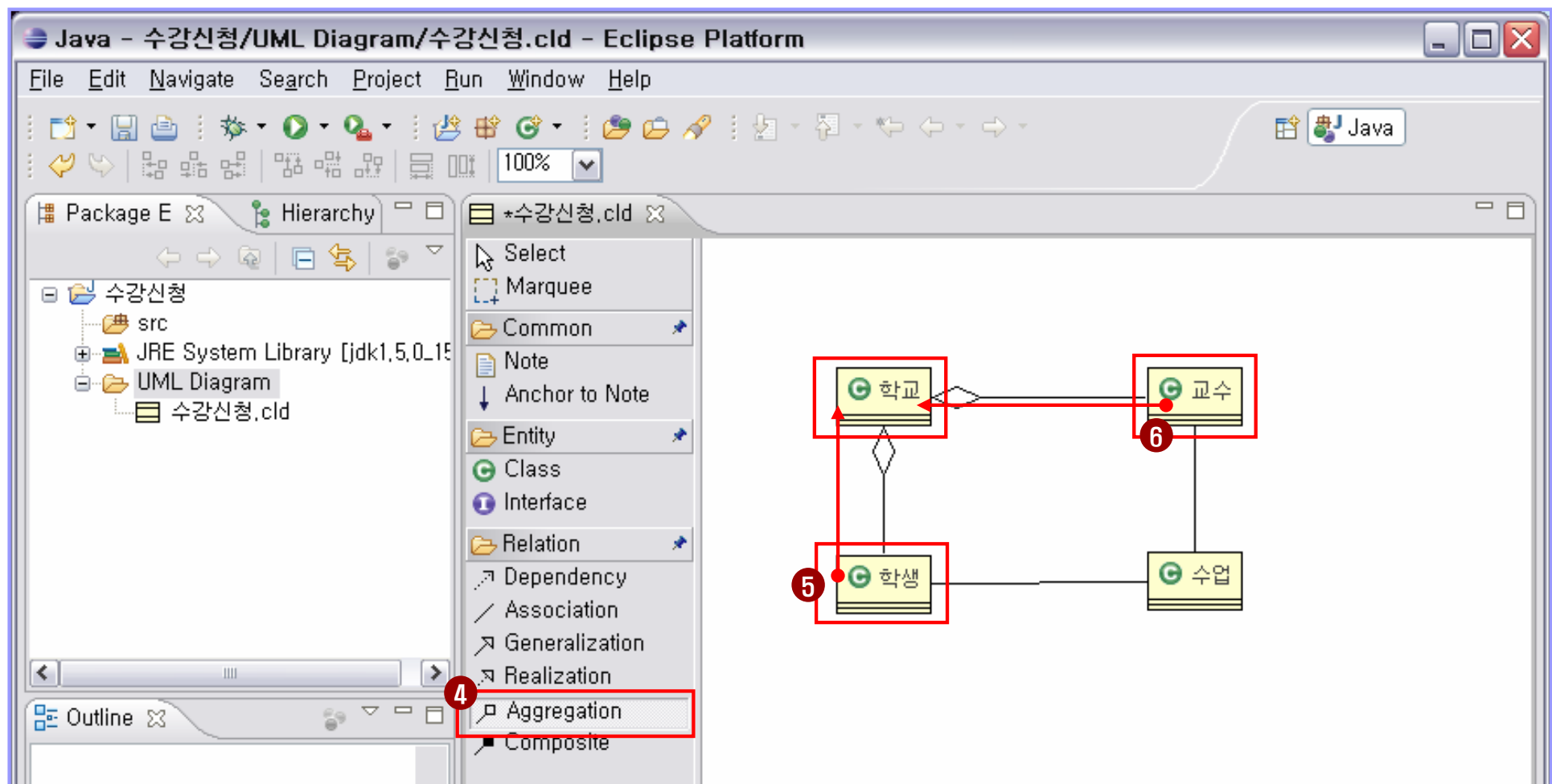
3. 도구 기능 소개

3.5 Class Diagram - 관계 생성 (2/2)

- 집합연관관계를 생성합니다.

- 팔레트를 사용하여 집합연관관계를 생성합니다.

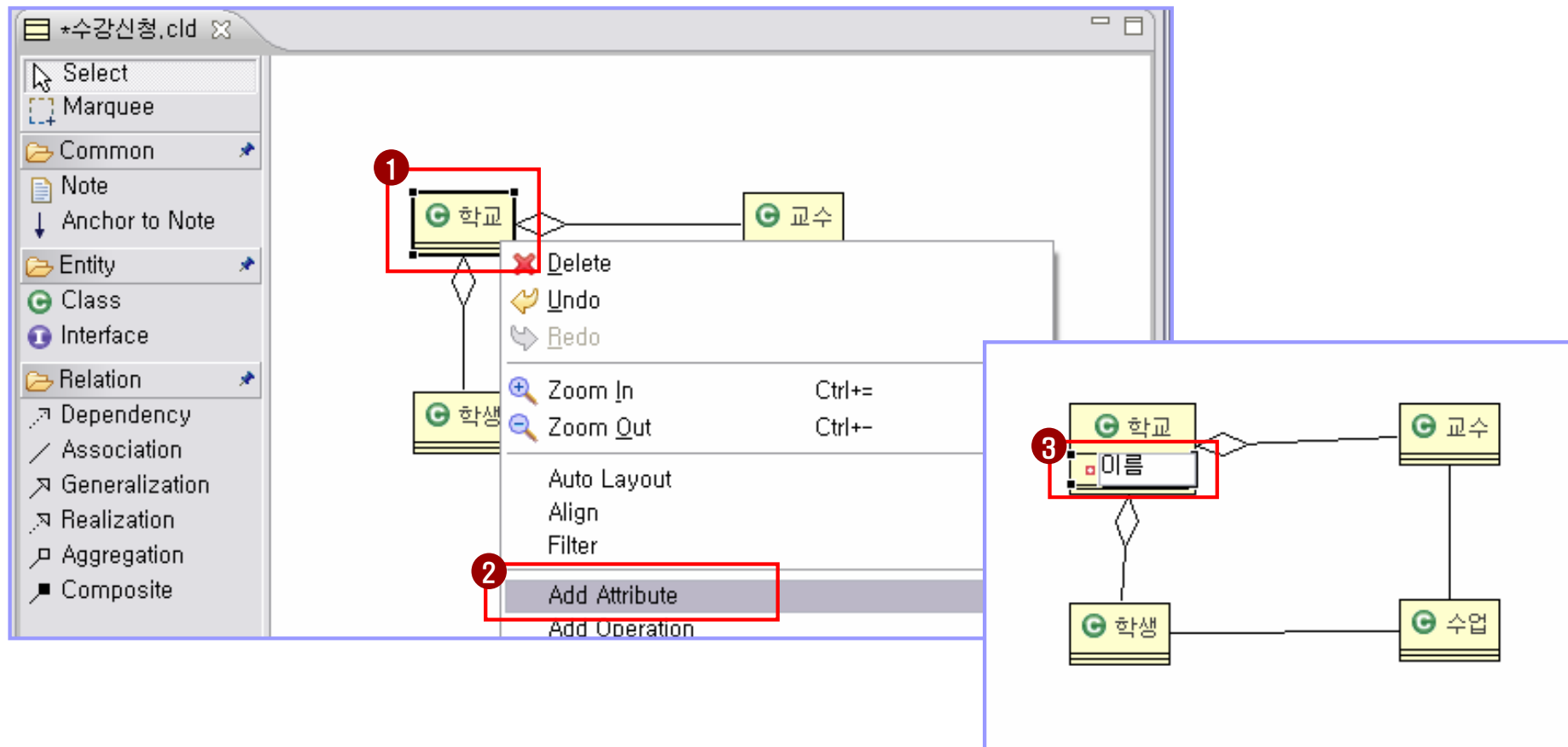
- 팔레트에서 Aggregation 클릭 -> 학생 Class에서 학교 Class로 클릭 앤 드래그 -> 교수 Class에서 학교 Class로 클릭 앤 드래그



3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (1/6)

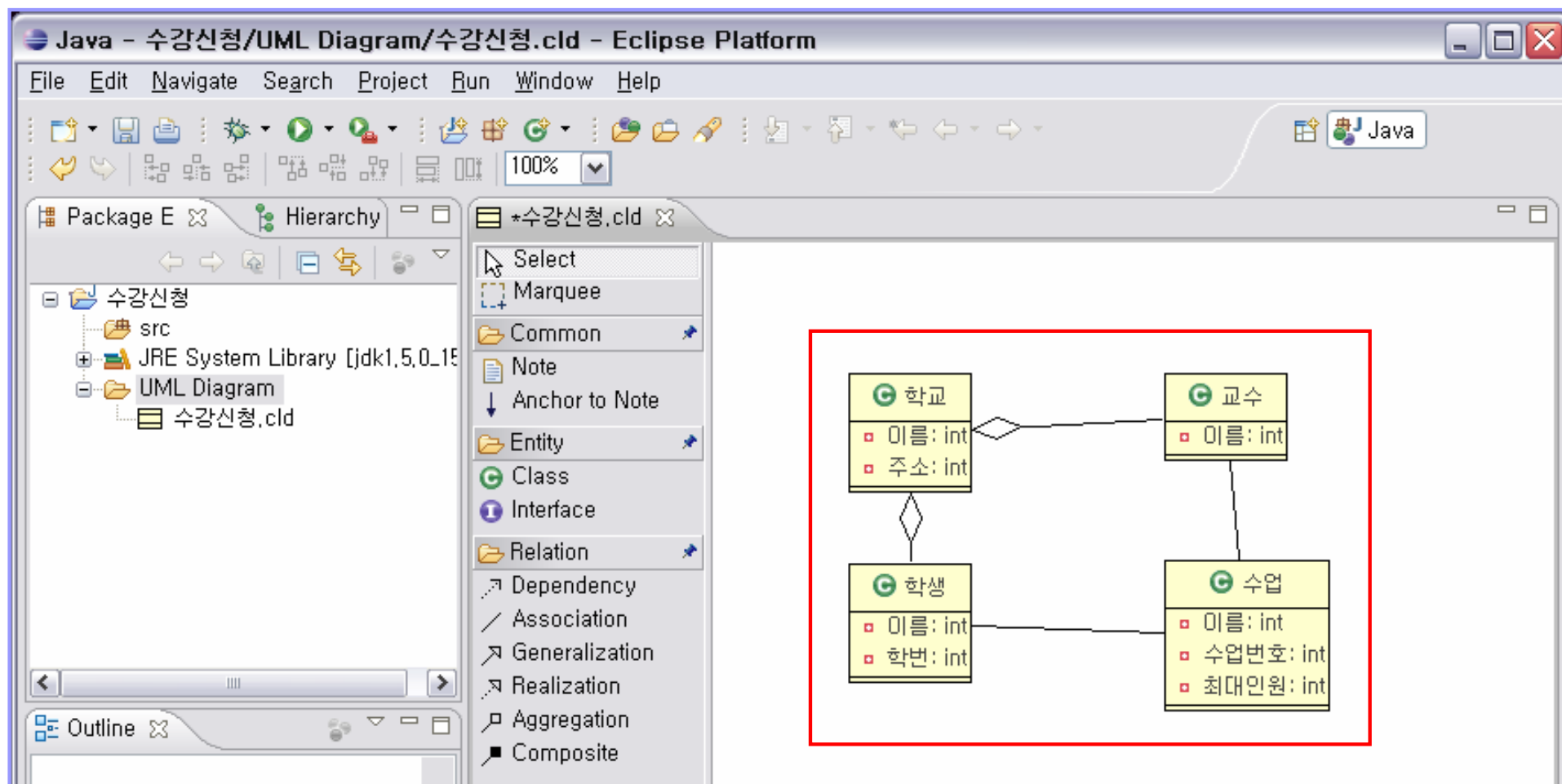
- 클래스의 속성을 생성합니다.
 - 생성되어 있는 클래스에 속성을 추가할 수 있습니다.
 - 학교 Class 오른쪽클릭 -> Add Attribute 클릭 -> Attribute name 지정



3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (2/6)

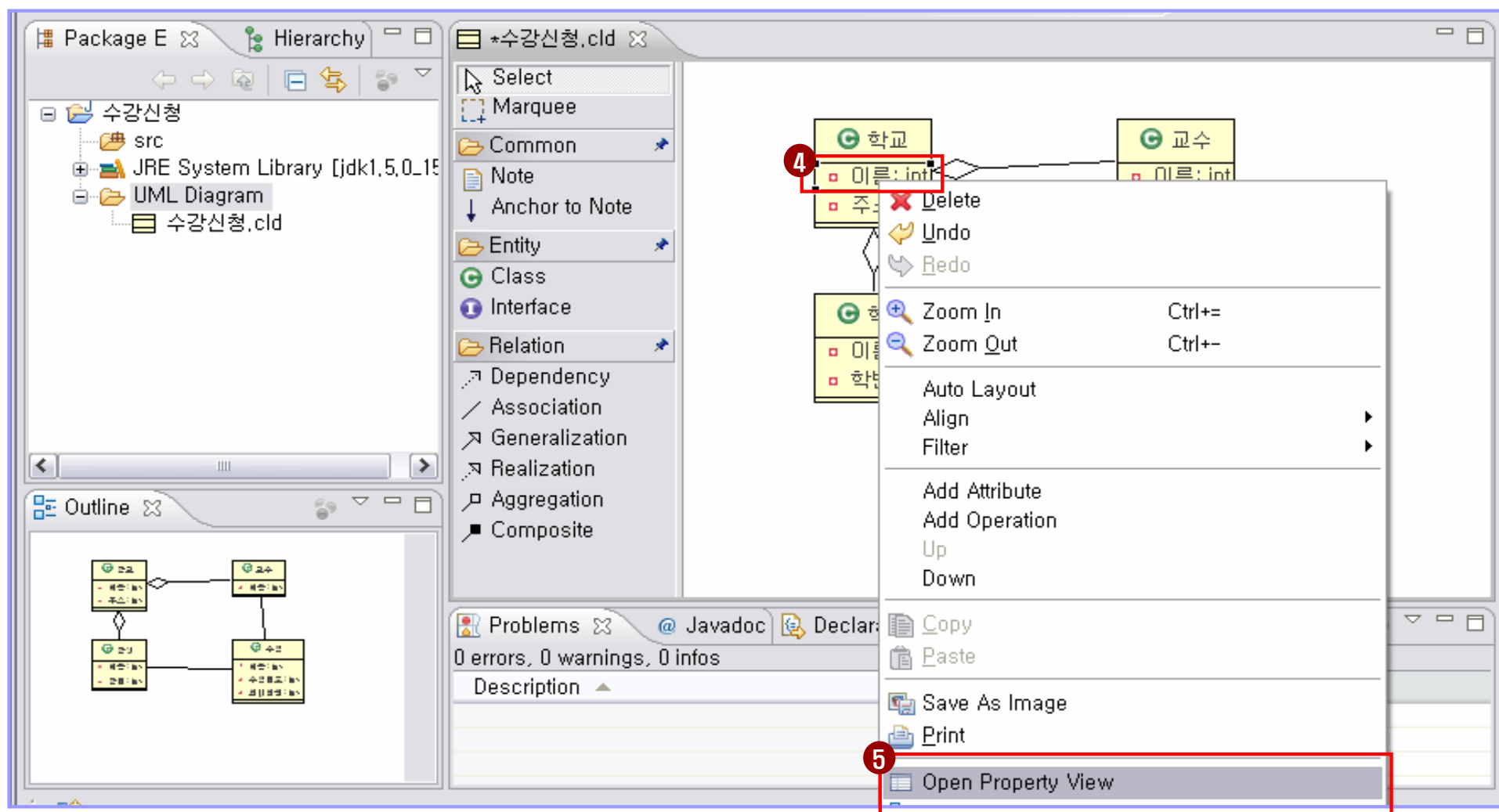
- 이전 페이지와 같은 방법으로 속성을 추가합니다.



3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (3/6)

- 속성 타입을 수정합니다.
 - 속성 뷰를 통해 속성정보를 수정할 수 있습니다.
 - 이름 속성 오른쪽클릭 -> Open Property View 클릭



3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (4/6)

- 속성 타입을 수정합니다.
 - 속성 뷰를 통해 속성정보를 수정할 수 있습니다.
 - Properties View에서 Type Value를 String으로 변경

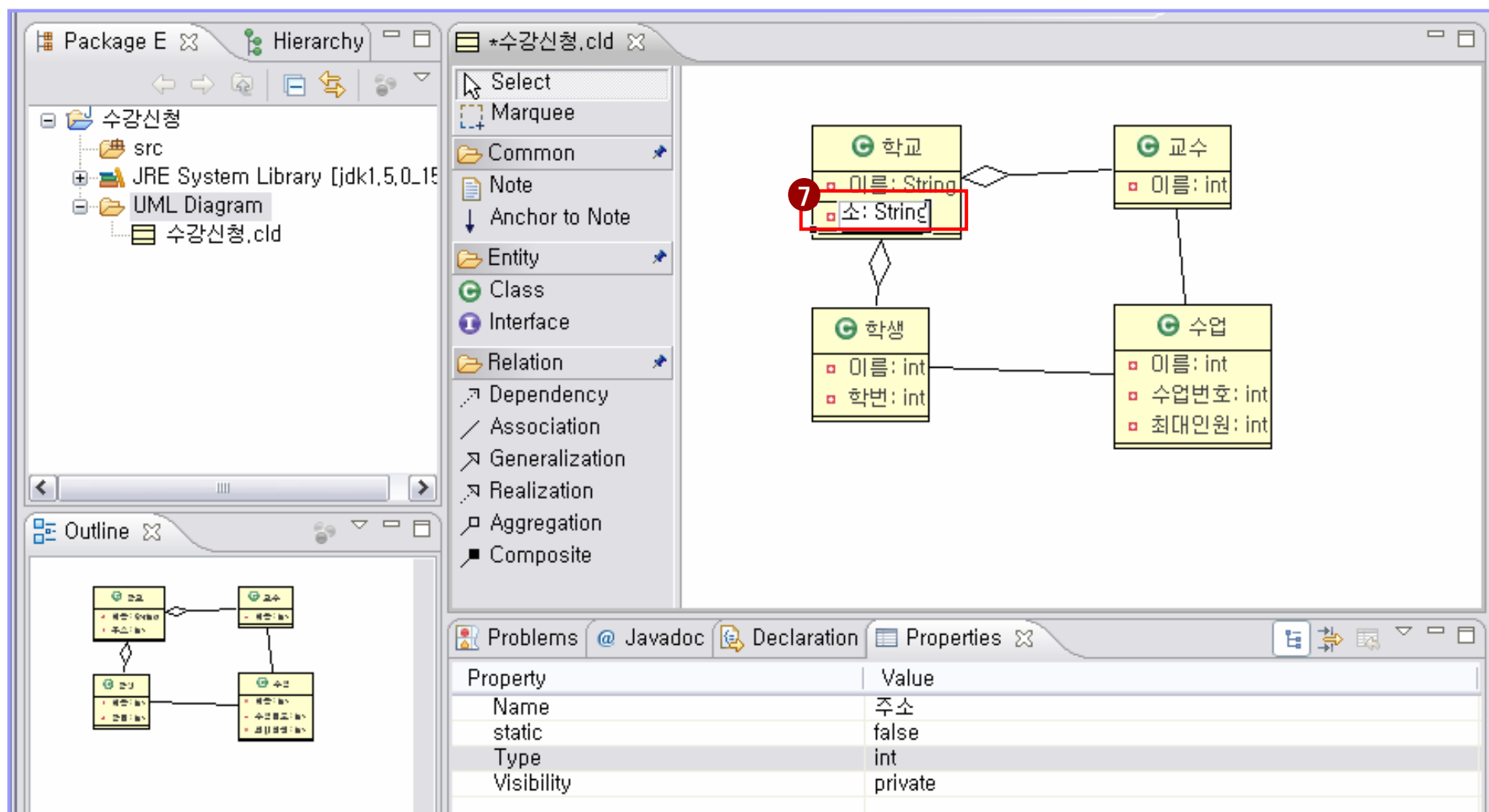
The screenshot displays the AmaterasUML interface. On the left, the Package Explorer shows a project named '수강신청' (Course Registration) with a package 'src' containing a class diagram '수강신청.cld'. The main canvas shows a class diagram with four classes: '학교' (School), '교수' (Professor), '학생' (Student), and '수업' (Class). '학교' and '교수' are connected by a composition relationship. '학교' and '학생' are connected by a composition relationship. '교수' and '수업' are connected by a composition relationship. '학생' and '수업' are connected by a composition relationship. The Properties view at the bottom right shows the properties of the selected class. The 'Type' property is highlighted with a red box and a red circle containing the number 6.

Property	Value
Name	이름
static	false
Type	String
Visibility	private

3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (5/6)

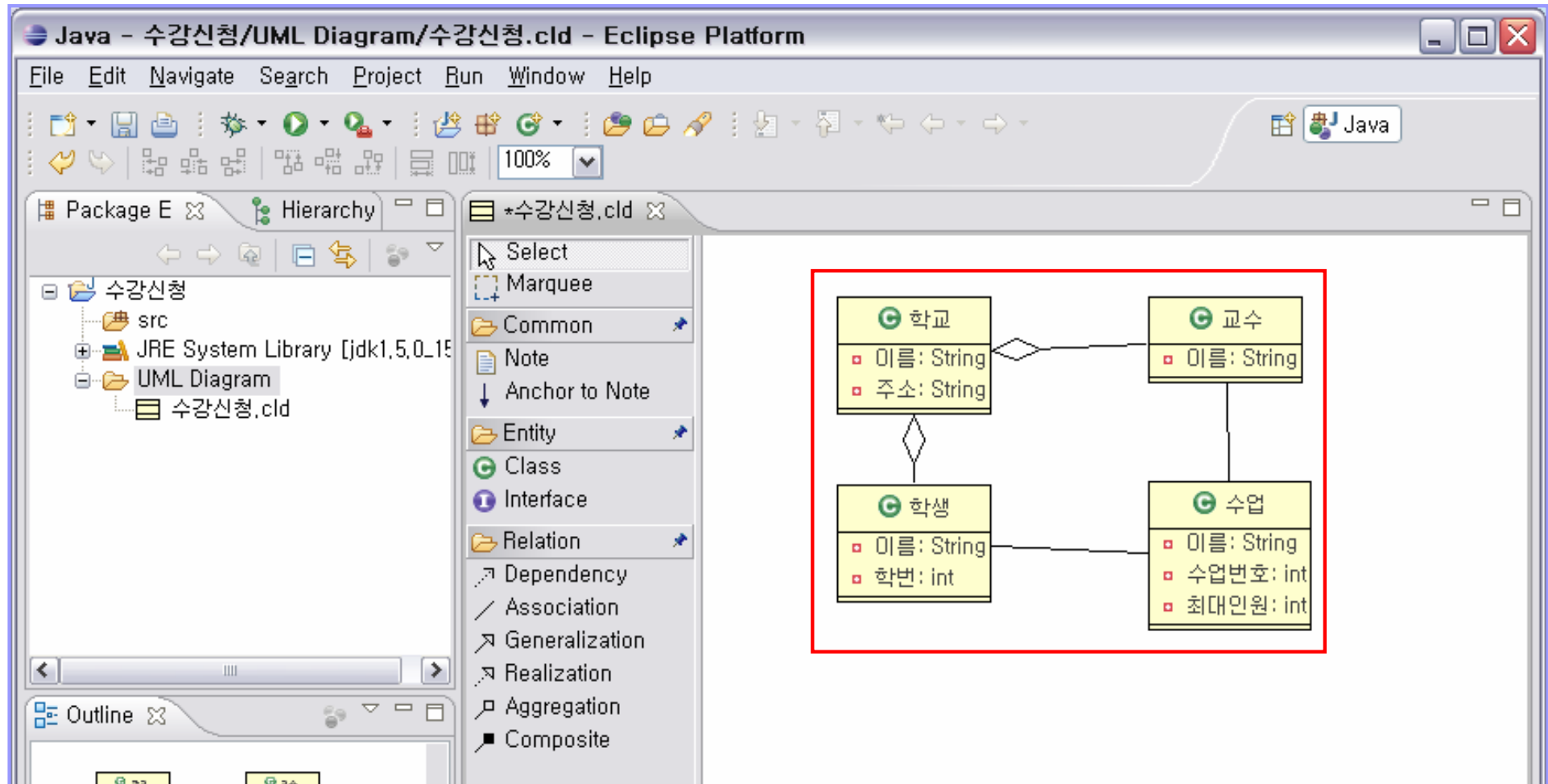
- 속성 타입을 수정합니다.
 - 속성 뷰가 아닌 다이어그램에서 직접 타입을 수정할 수도 있습니다.
 - 주소 속성 클릭 -> int를 String로 변경



3. 도구 기능 소개

3.5 Class Diagram - 속성 생성 (6/6)

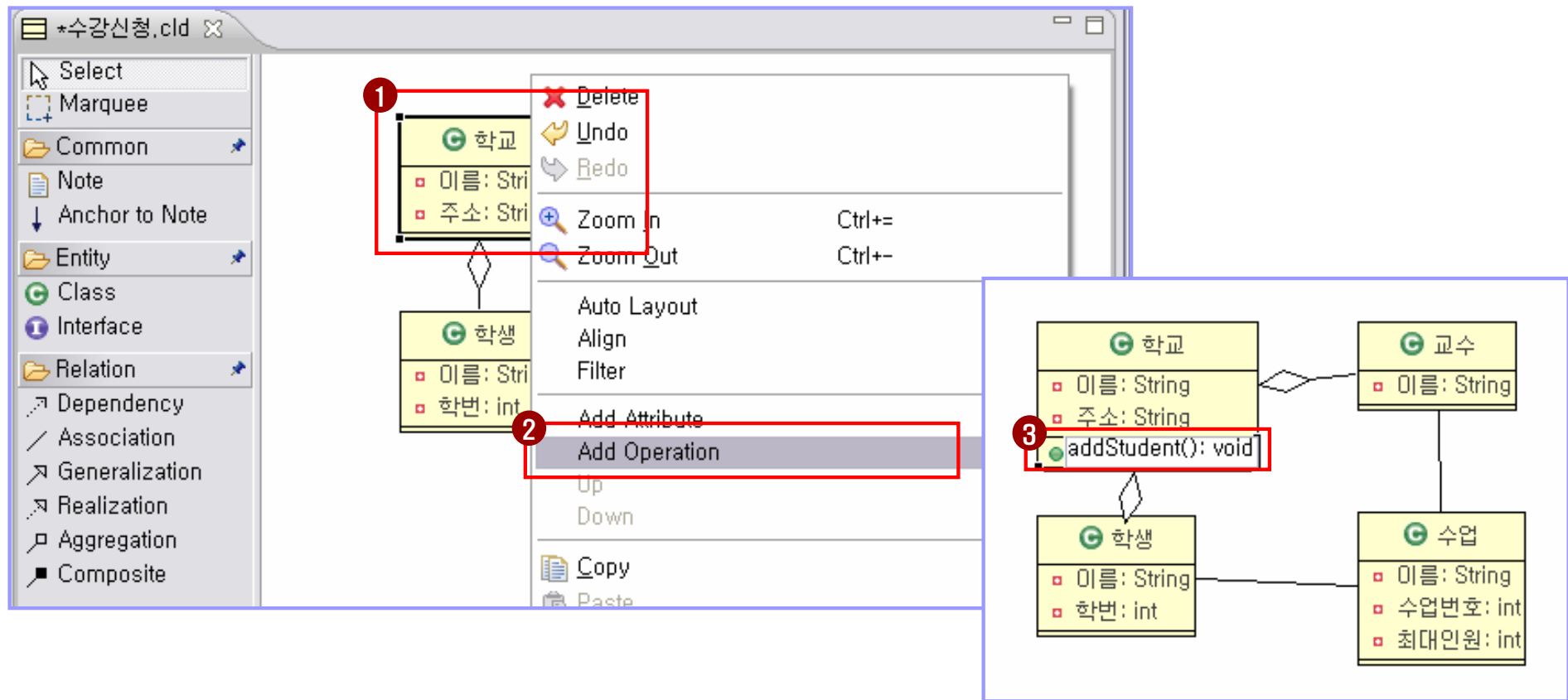
- 34 또는 35 페이지와 같은 방법으로 속성을 추가합니다.



3. 도구 기능 소개

3.5 Class Diagram - 함수 생성 (1/2)

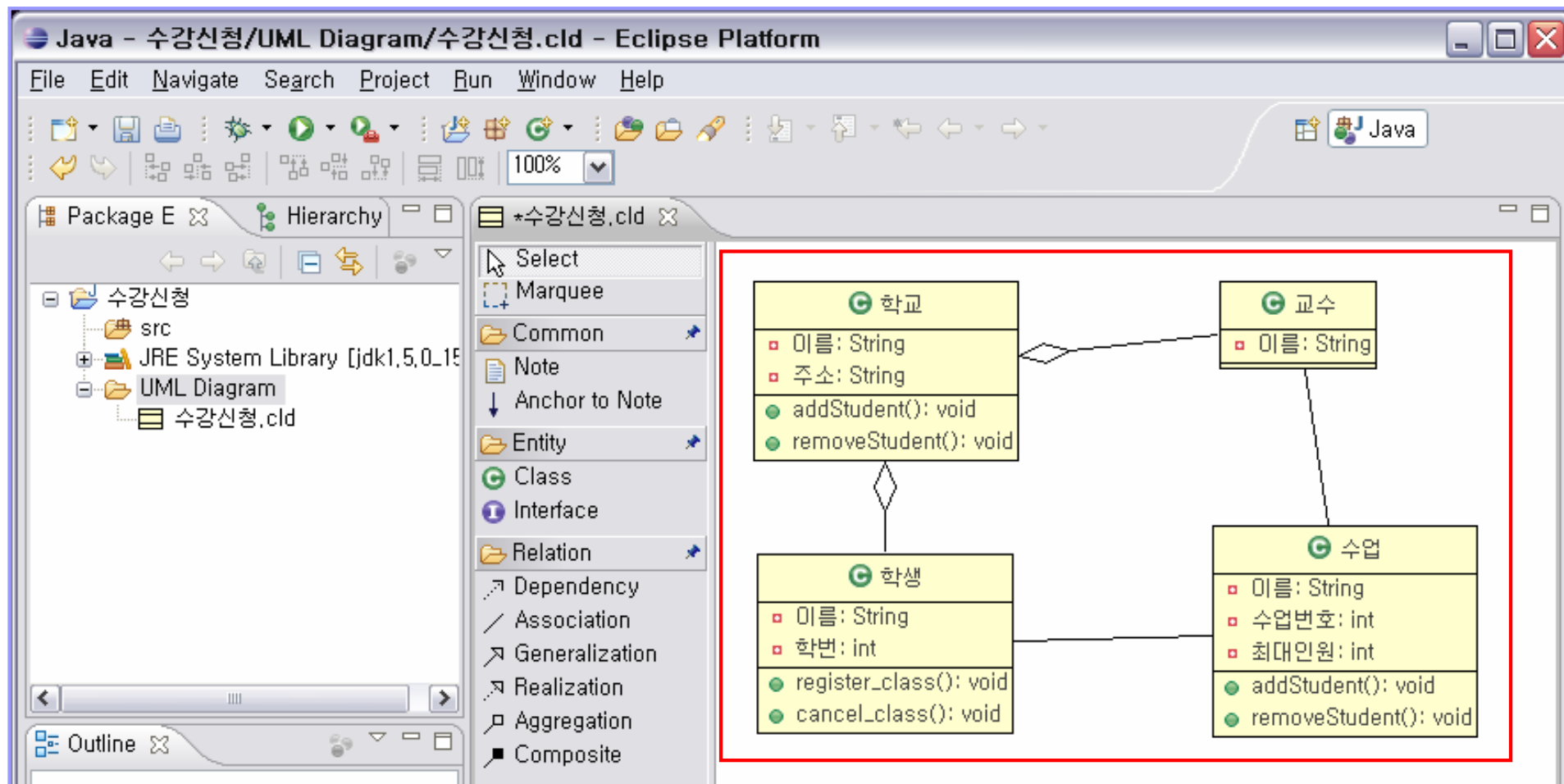
- 클래스의 함수를 생성합니다.
 - 생성되어 있는 클래스에 함수를 추가할 수 있습니다.
 - 학교 Class 오른쪽 클릭 -> Add Operation 클릭 -> Operation name 지정



3. 도구 기능 소개

3.5 Class Diagram - 함수 생성 (2/2)

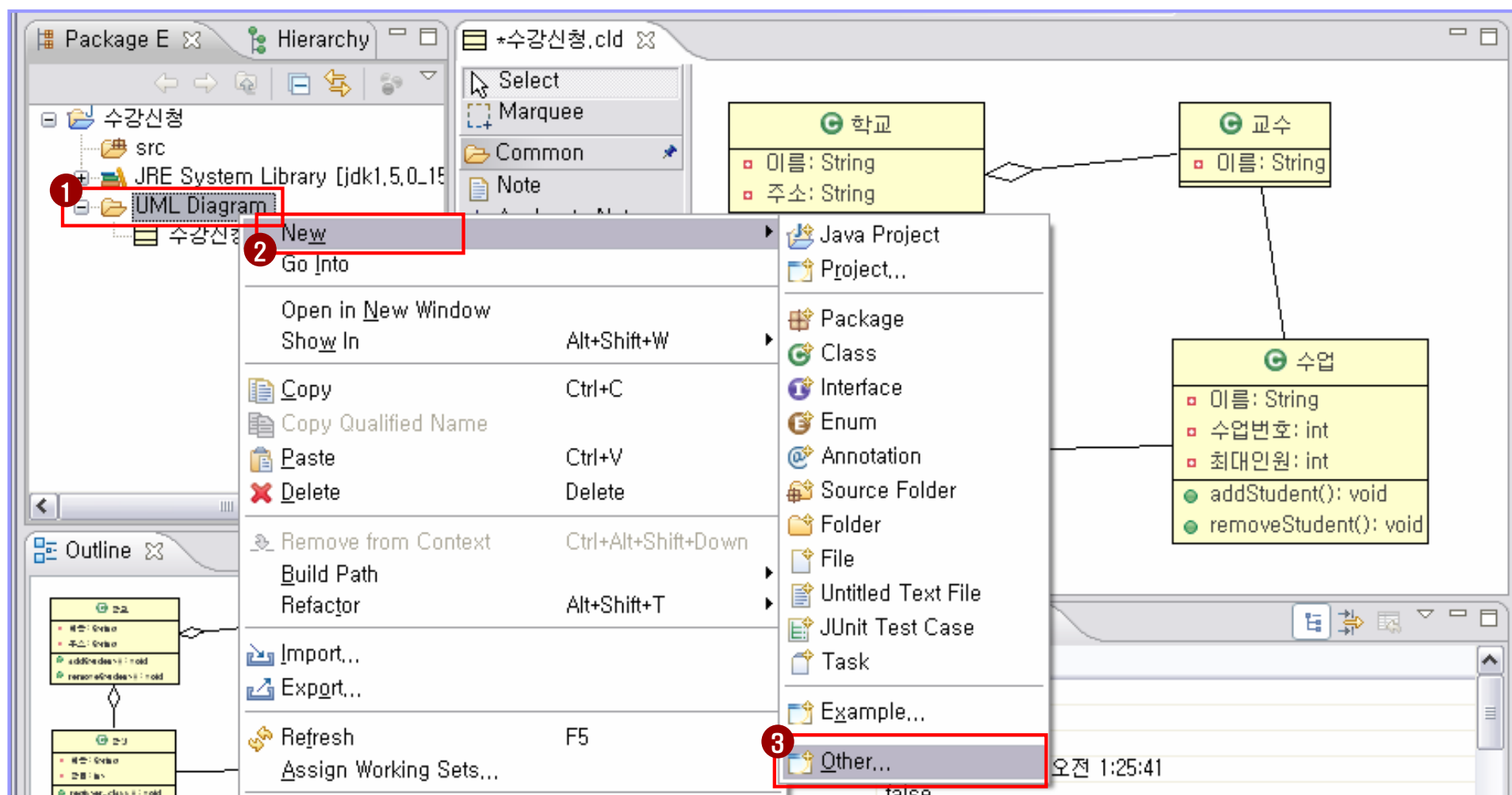
- 이전 페이지와 같은 방법으로 함수를 추가합니다.



3. 도구 기능 소개

3.6 Use Case Diagram - 다이어그램 생성 (1/3)

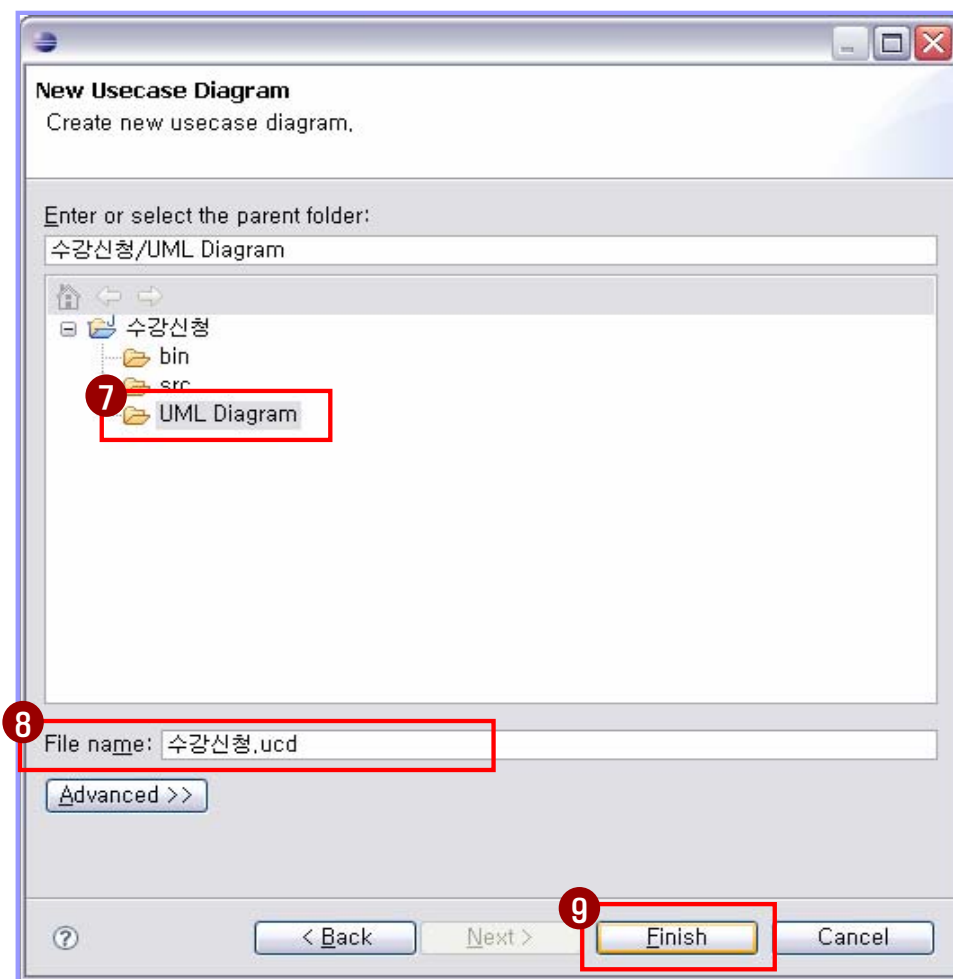
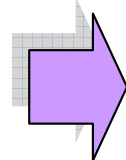
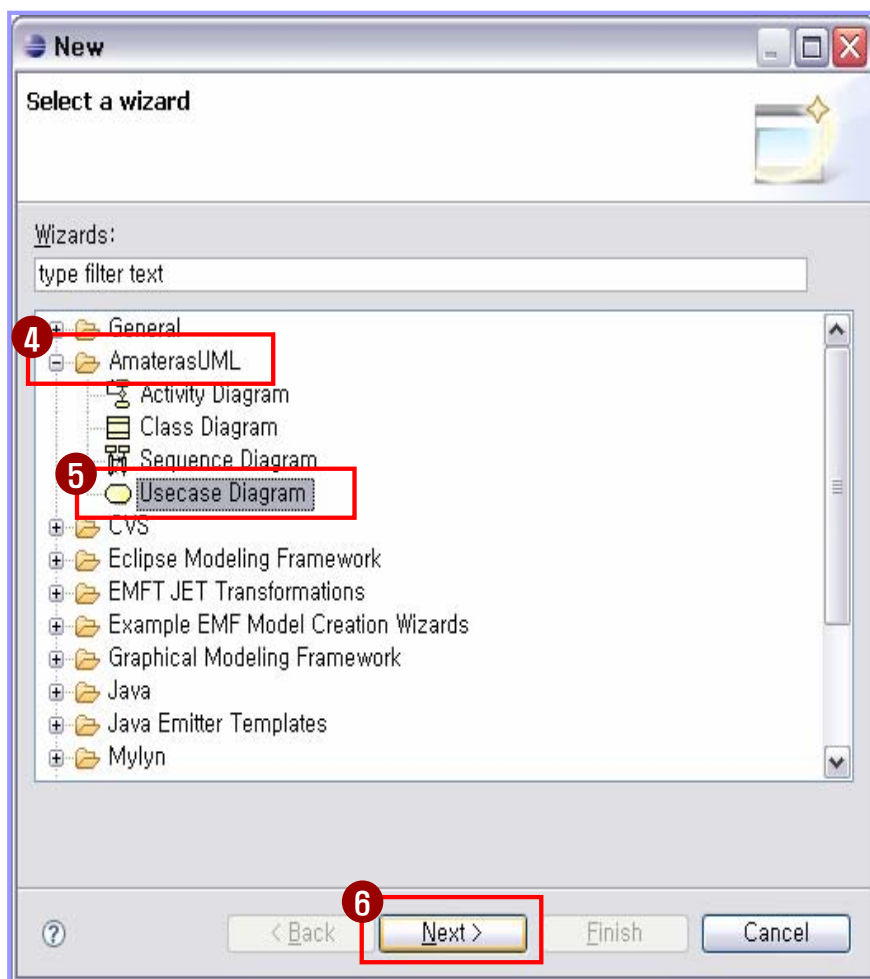
- UML Diagram 파일을 생성합니다.
 - 수강신청 프로젝트의 UML Diagram 폴더에 다이어그램을 작성하기 위한 UML 파일을 생성하겠습니다.
 - UML Diagram 폴더 오른쪽 클릭 -> New -> Other... 클릭



3. 도구 기능 소개

3.6 Use Case Diagram - 다이어그램 생성 (2/3)

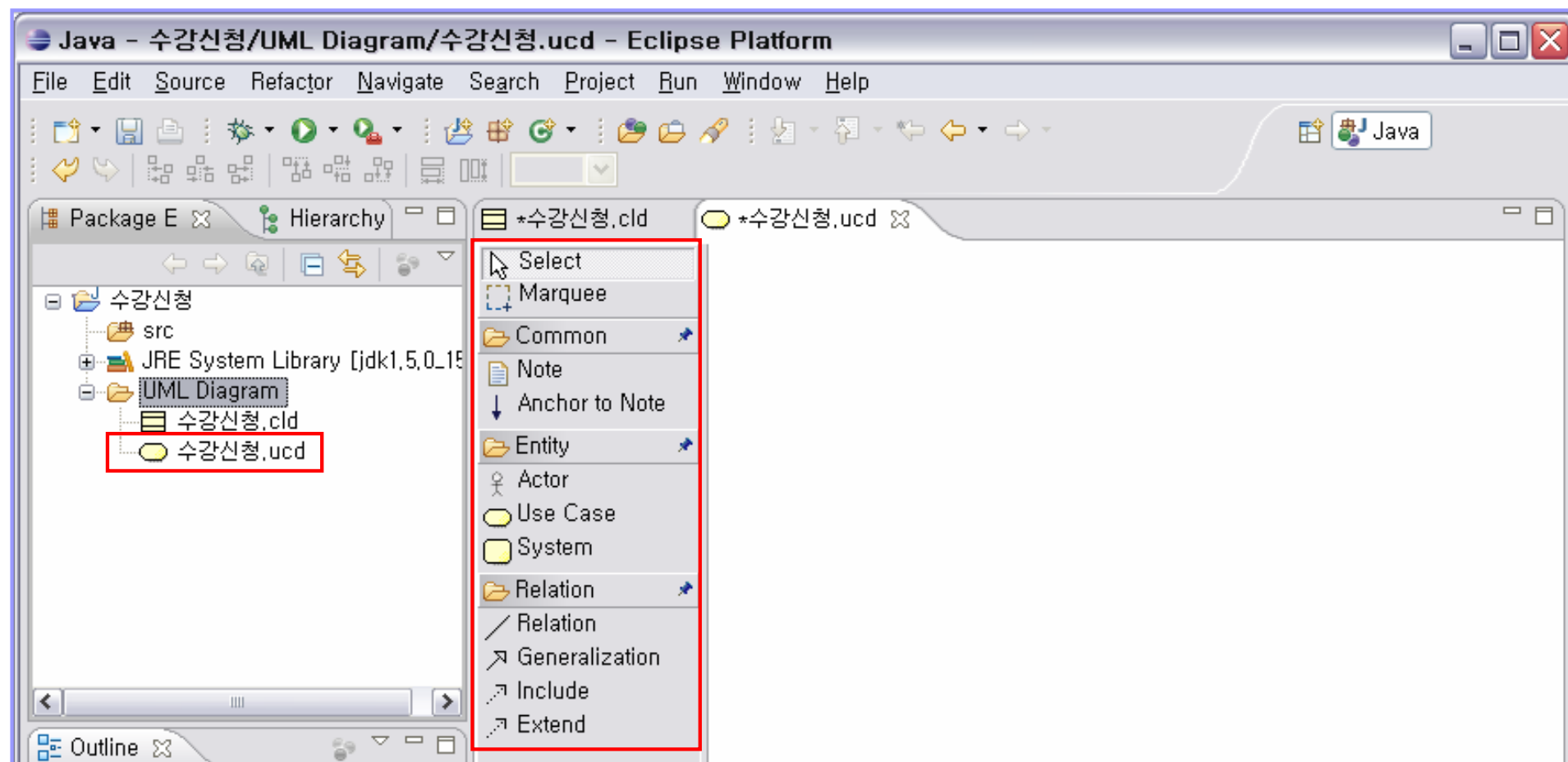
- Use Case Diagram을 생성합니다.
 - 확장자가 ucd(Use Case Diagram)인 파일을 생성합니다.
 - AmaterasUML 하위 메뉴인 Use Case Diagram 클릭 -> Next 버튼 클릭
 - UML Diagram 폴더 선택 -> File name 입력 -> Finish 버튼 클릭



3. 도구 기능 소개

3.6 Use Case Diagram - 다이어그램 생성 (3/3)

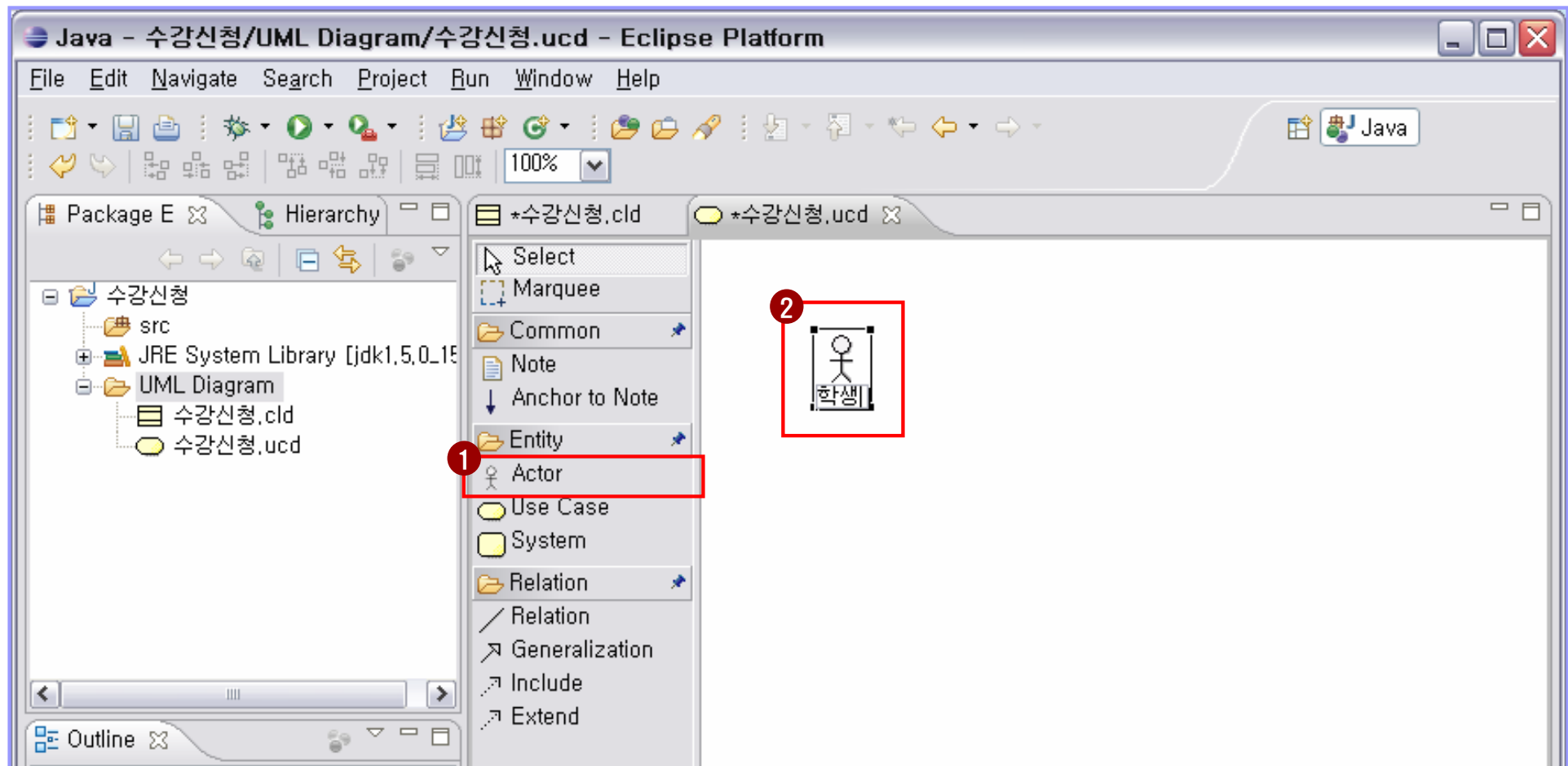
- Use Case Diagram이 생성되었습니다.
 - 수강신청.ucd 파일이 생성되었습니다.
 - Use Case Diagram과 관련된 팔레트를 확인할 수 있습니다.



3. 도구 기능 소개

3.6 Use Case Diagram - 액터 생성

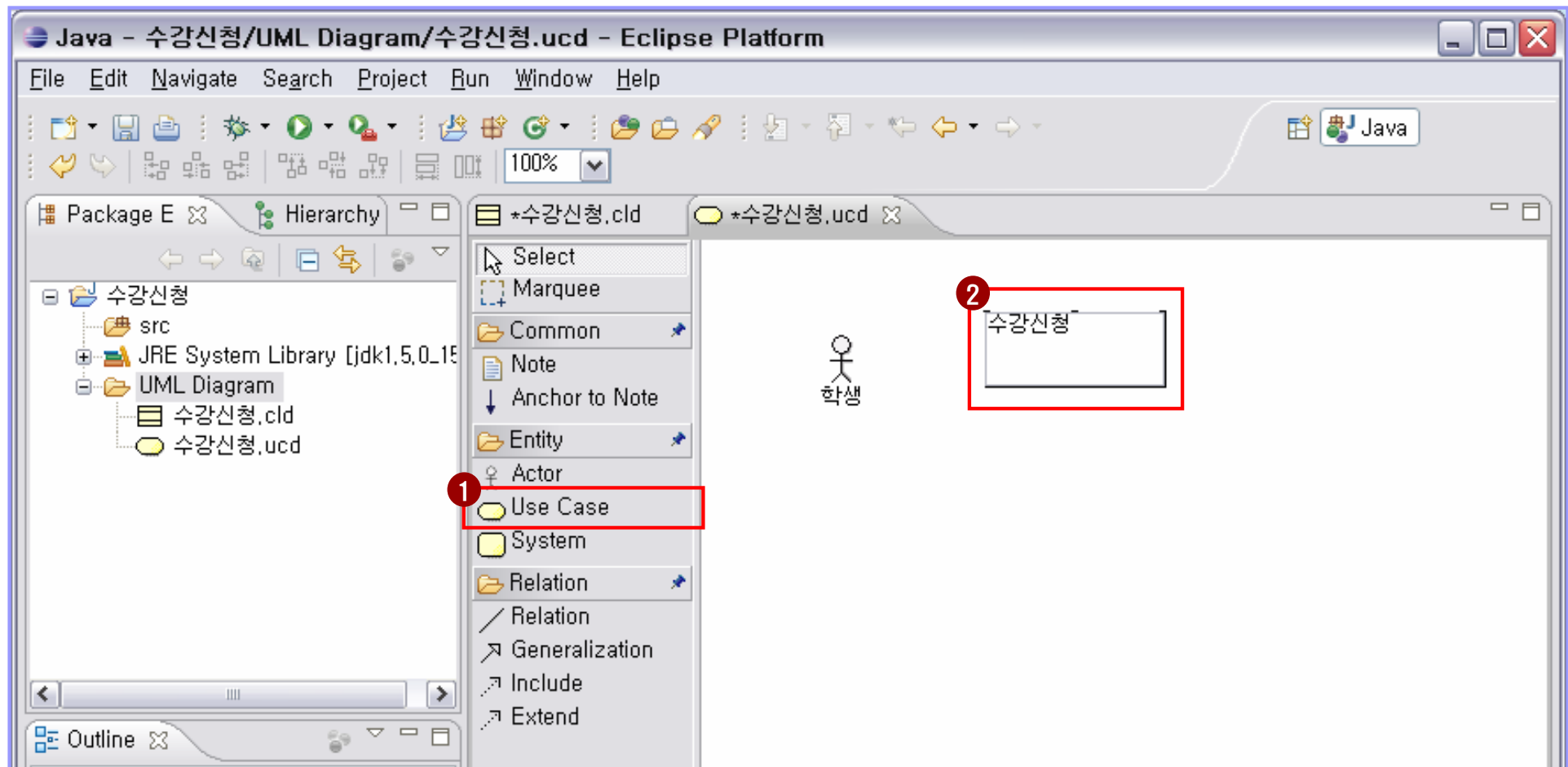
- 액터를 생성합니다.
 - 팔레트를 사용하여 액터를 생성합니다.
 - 팔레트에서 Actor 클릭 -> 가운데의 작업공간 클릭 -> Actor 이름 설정



3. 도구 기능 소개

3.6 Use Case Diagram - 유즈케이스 생성 [1/2]

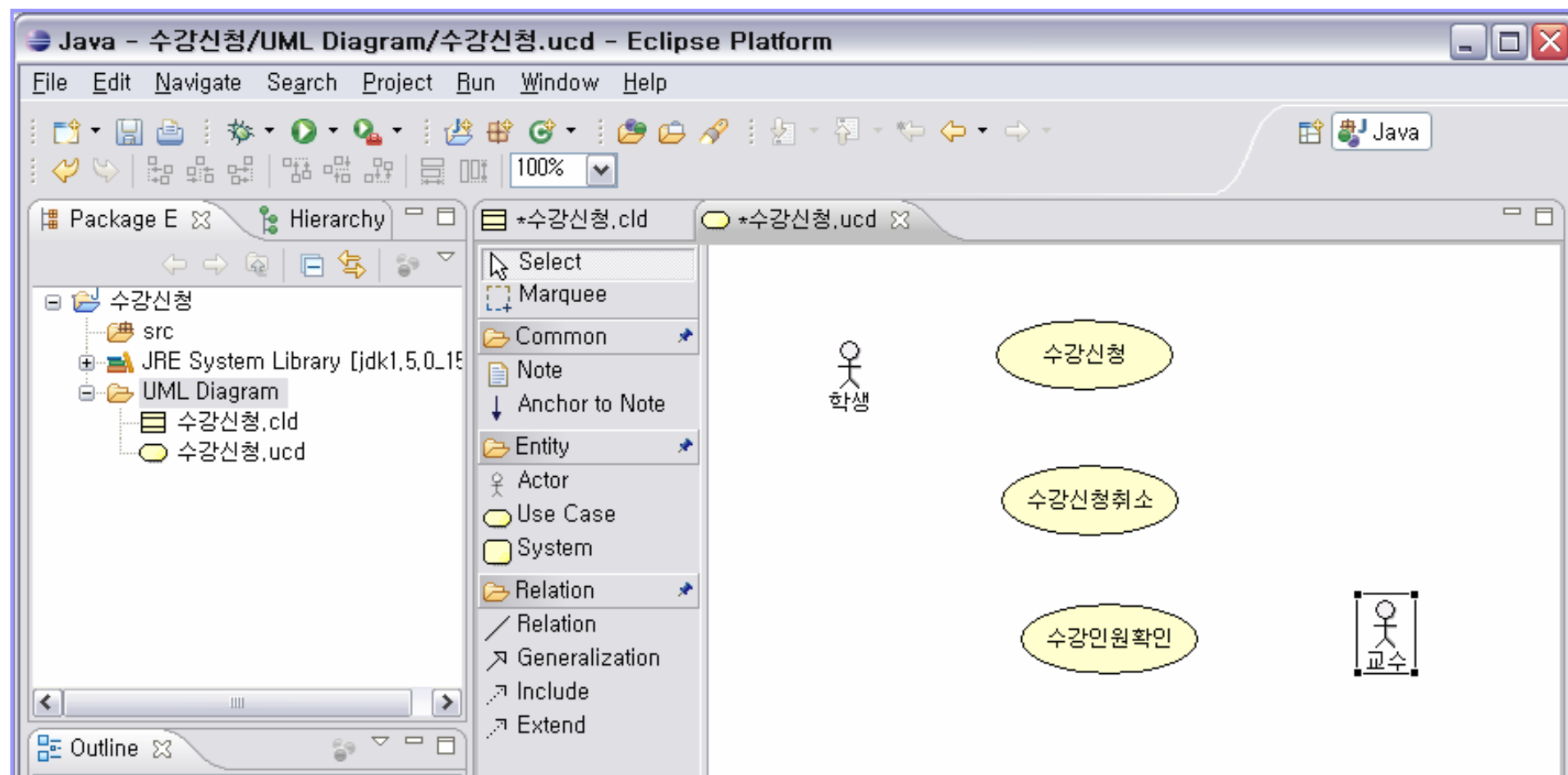
- 유즈케이스를 생성합니다.
 - 팔레트를 사용하여 유즈케이스를 생성합니다.
 - 팔레트에서 Use Case 클릭 -> 가운데의 작업공간 클릭 -> Use Case 이름 설정



3. 도구 기능 소개

3.6 Use Case Diagram - 유즈케이스 생성 [2/2]

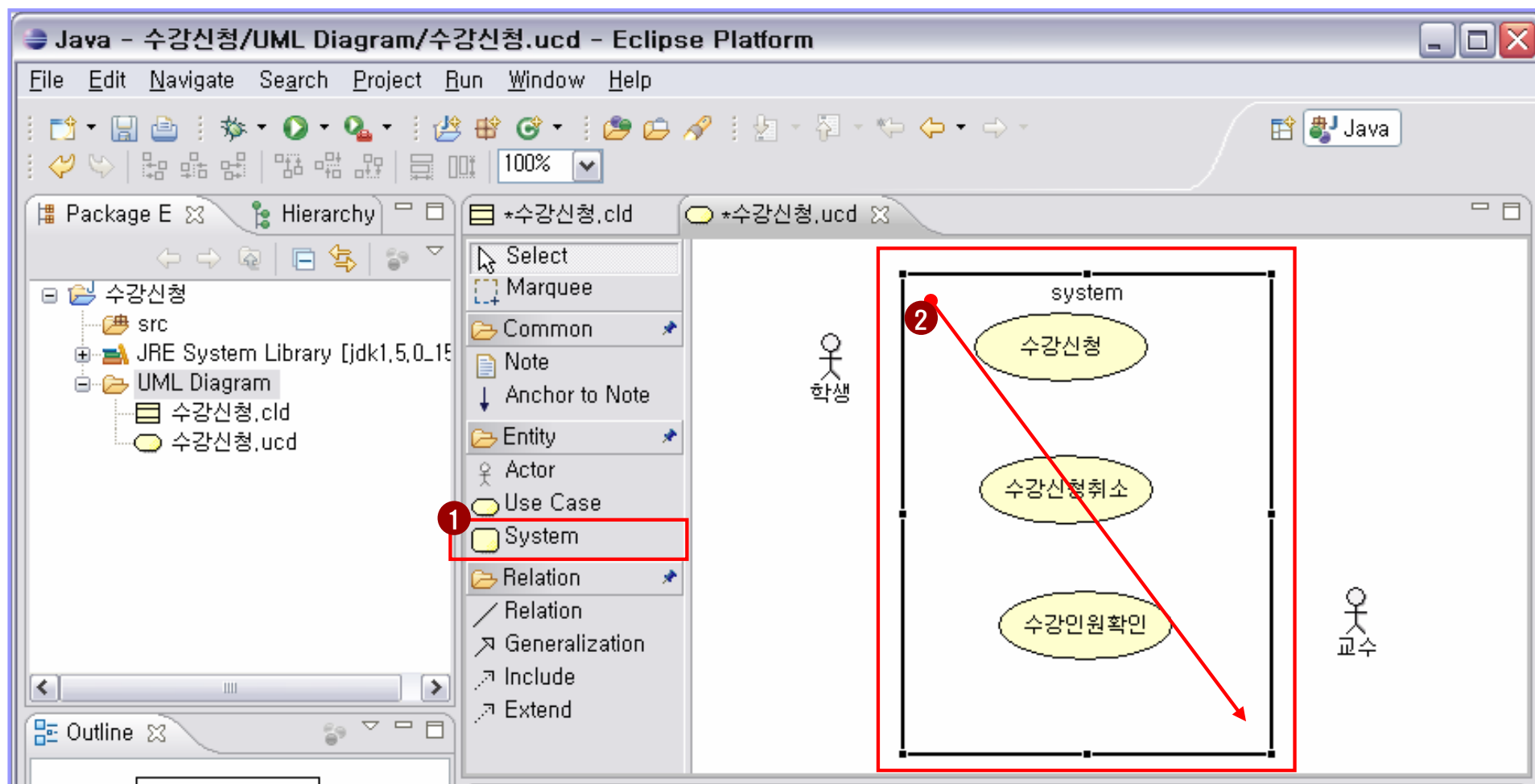
- 42, 43 페이지와 같은 방법으로 액터와 유즈케이스를 추가합니다.



3. 도구 기능 소개

3.6 Use Case Diagram - 시스템 바운더리 생성

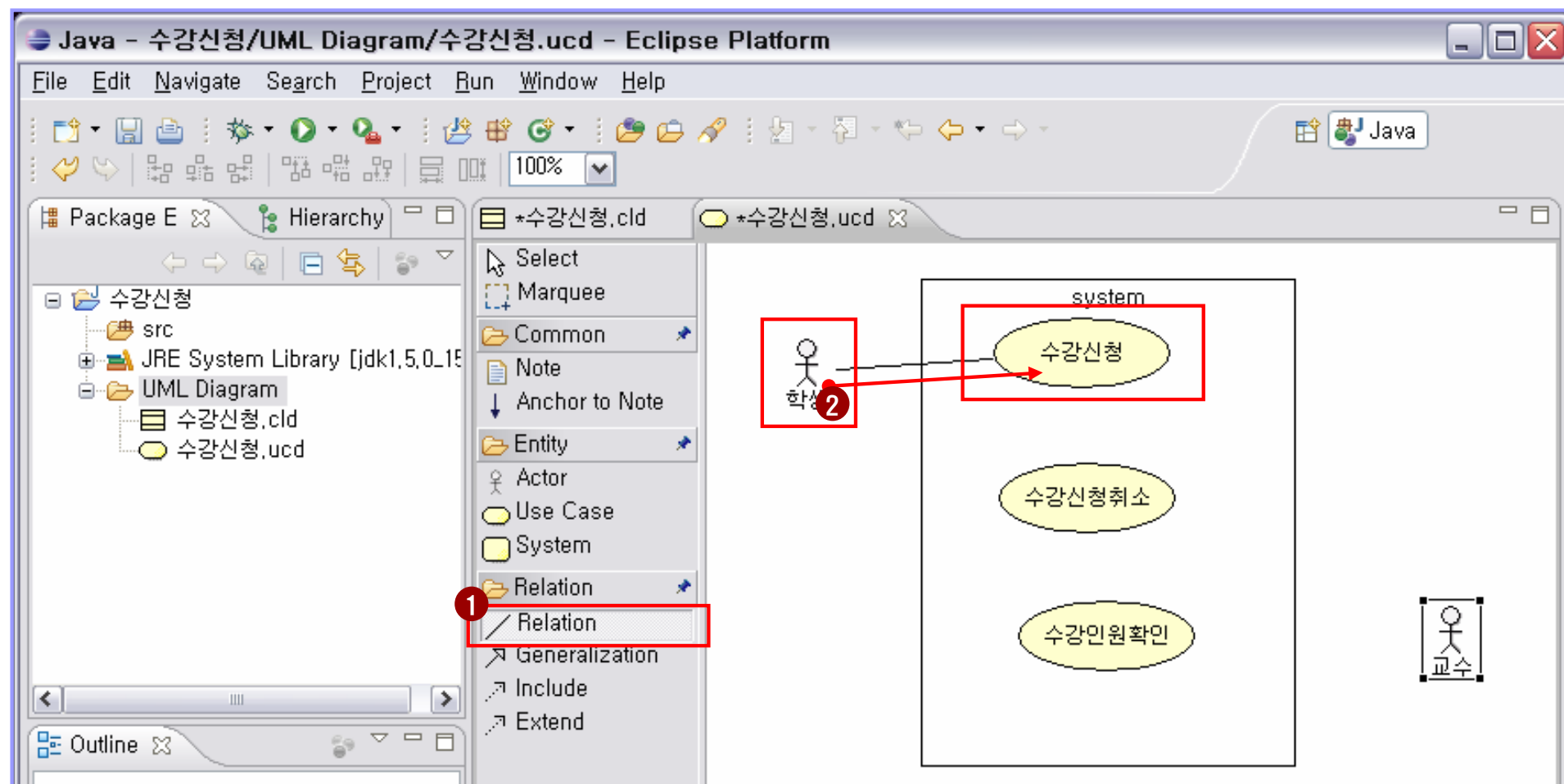
- 시스템 바운더리를 생성합니다.
 - 팔레트를 사용하여 시스템 바운더리를 생성합니다.
 - 팔레트에서 System 클릭 -> Use Case를 포함하도록 드래그



3. 도구 기능 소개

3.6 Use Case Diagram - 관계 생성 (1/2)

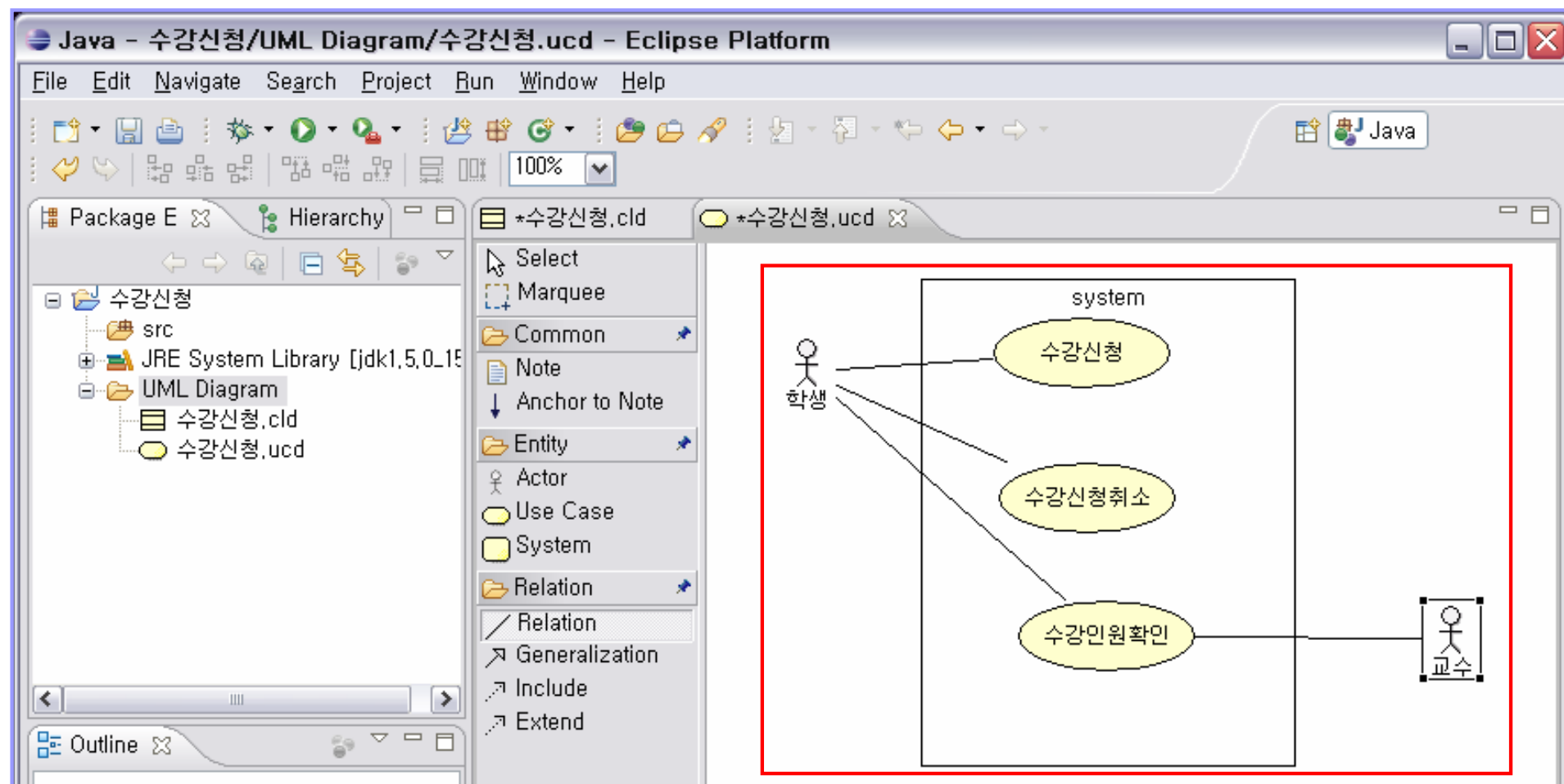
- 액터와 유즈케이스를 연결합니다.
 - 팔레트를 사용하여 액터와 유즈케이스를 연결합니다.
 - 팔레트에서 Relation 클릭 -> 학생 Actor에서 수강신청 Use Case로 클릭 앤 드래그



3. 도구 기능 소개

3.6 Use Case Diagram - 관계 생성 (2/2)

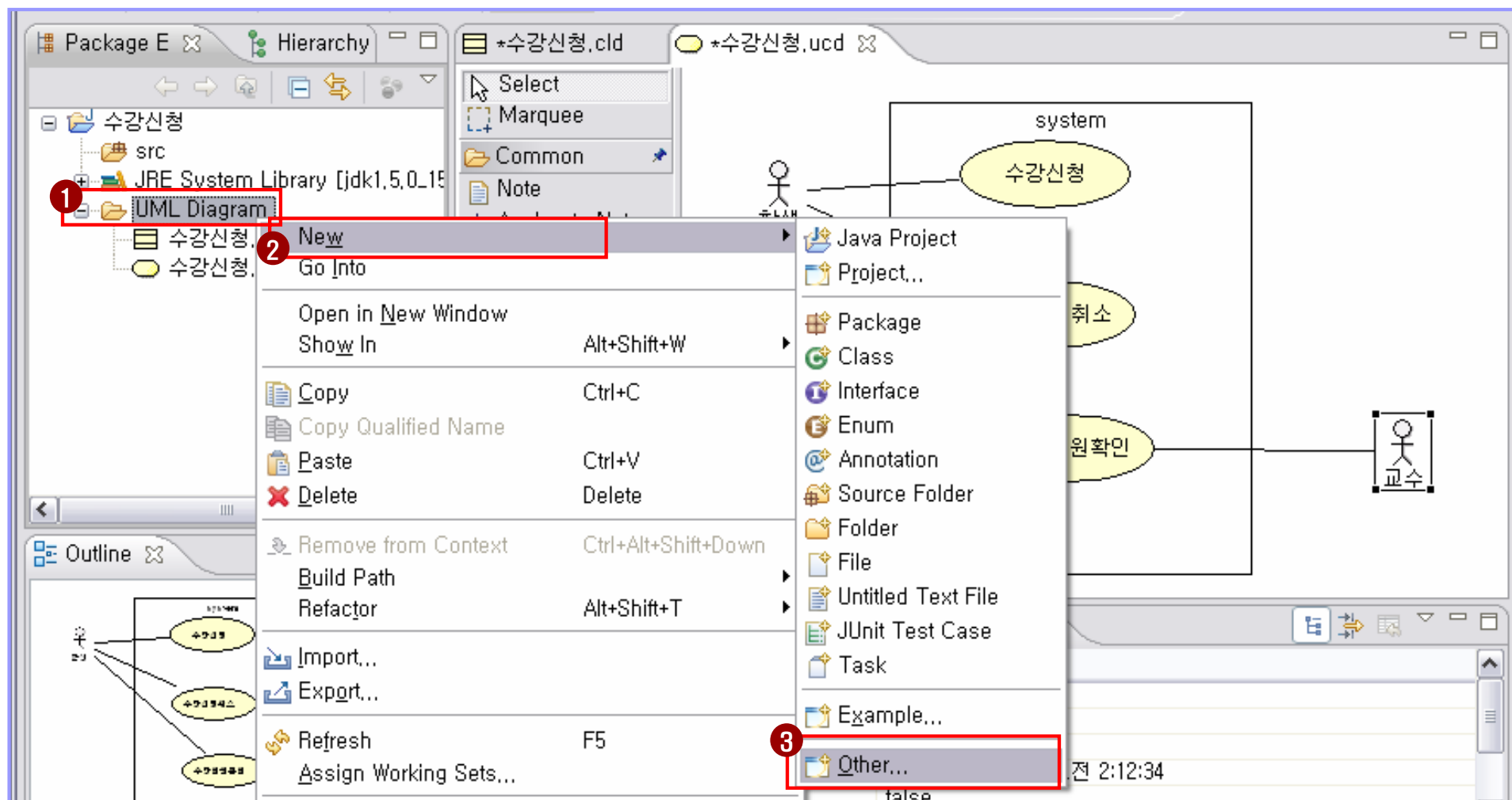
- 이전 페이지와 같은 방법으로 액터와 유즈케이스를 연결합니다.



3. 도구 기능 소개

3.7 Activity Diagram - 다이어그램 생성 (1/3)

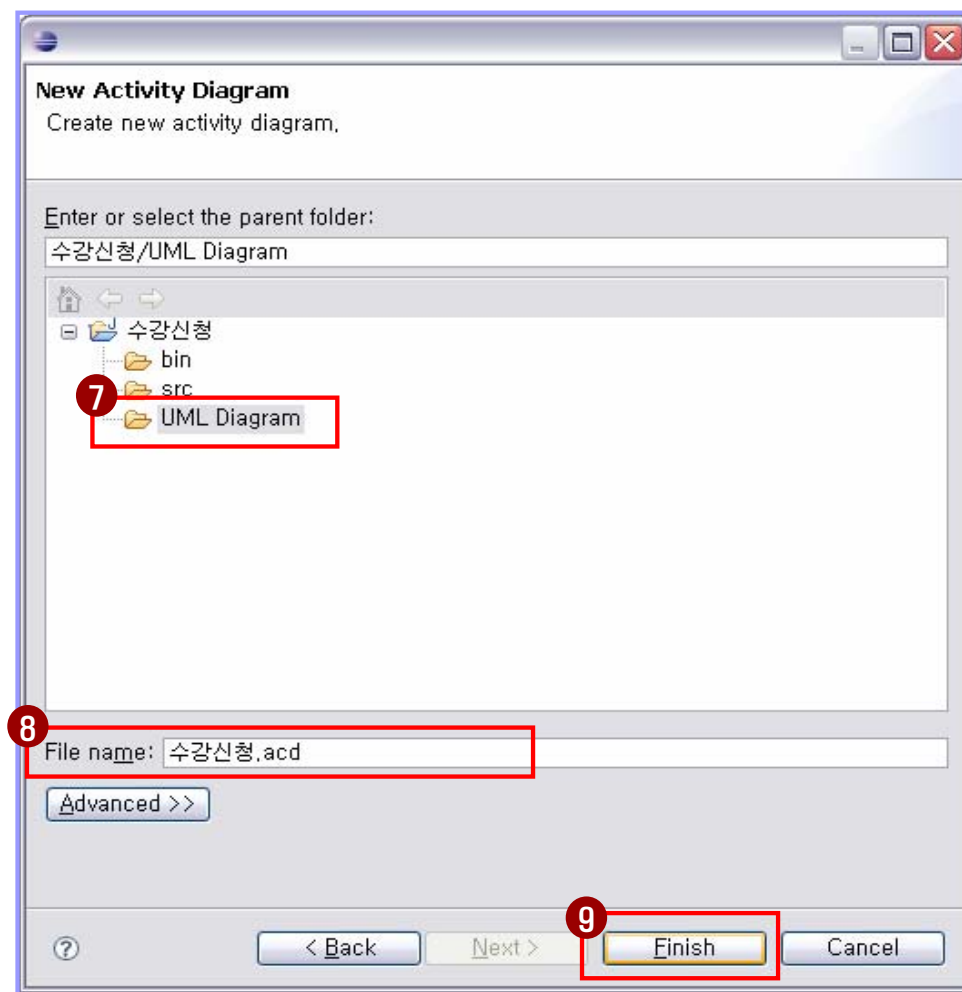
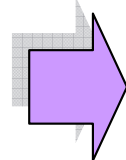
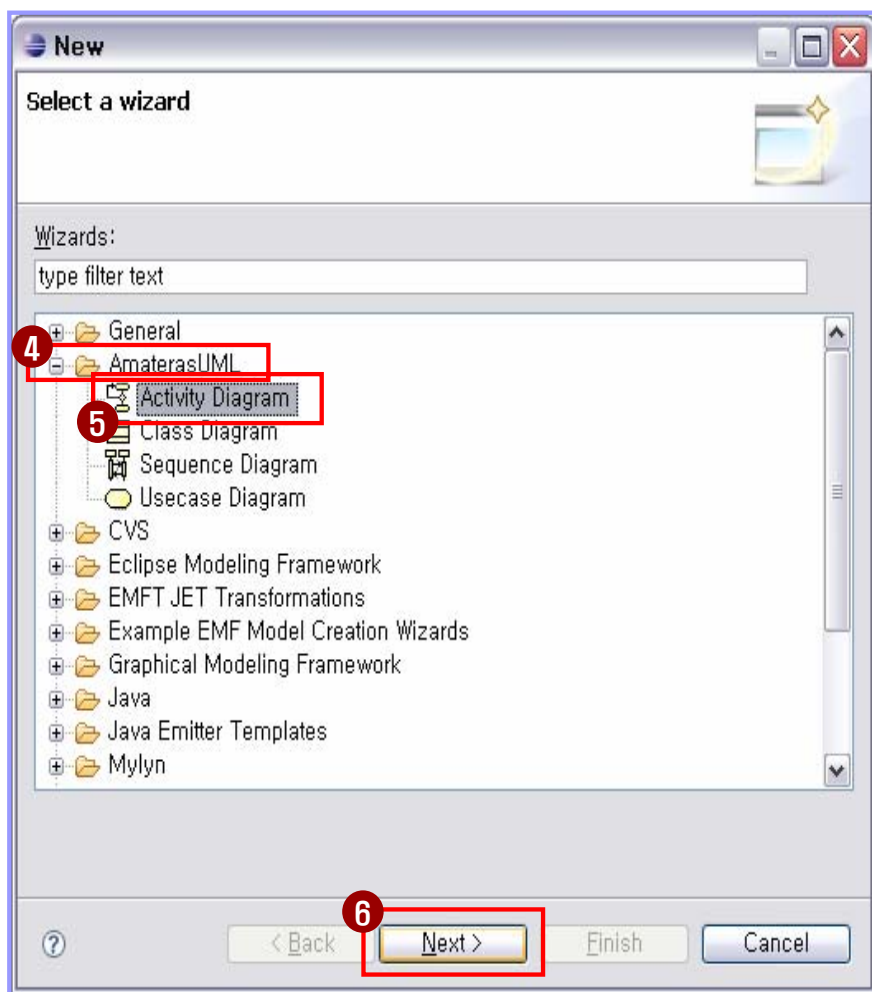
- UML Diagram 파일을 생성합니다.
 - 수강신청 프로젝트의 UML Diagram 폴더에 다이어그램을 작성하기 위한 UML 파일을 생성하겠습니다.
 - UML Diagram 폴더 오른쪽 클릭 -> New -> Other... 클릭



3. 도구 기능 소개

3.7 Activity Diagram - 다이어그램 생성 [2/3]

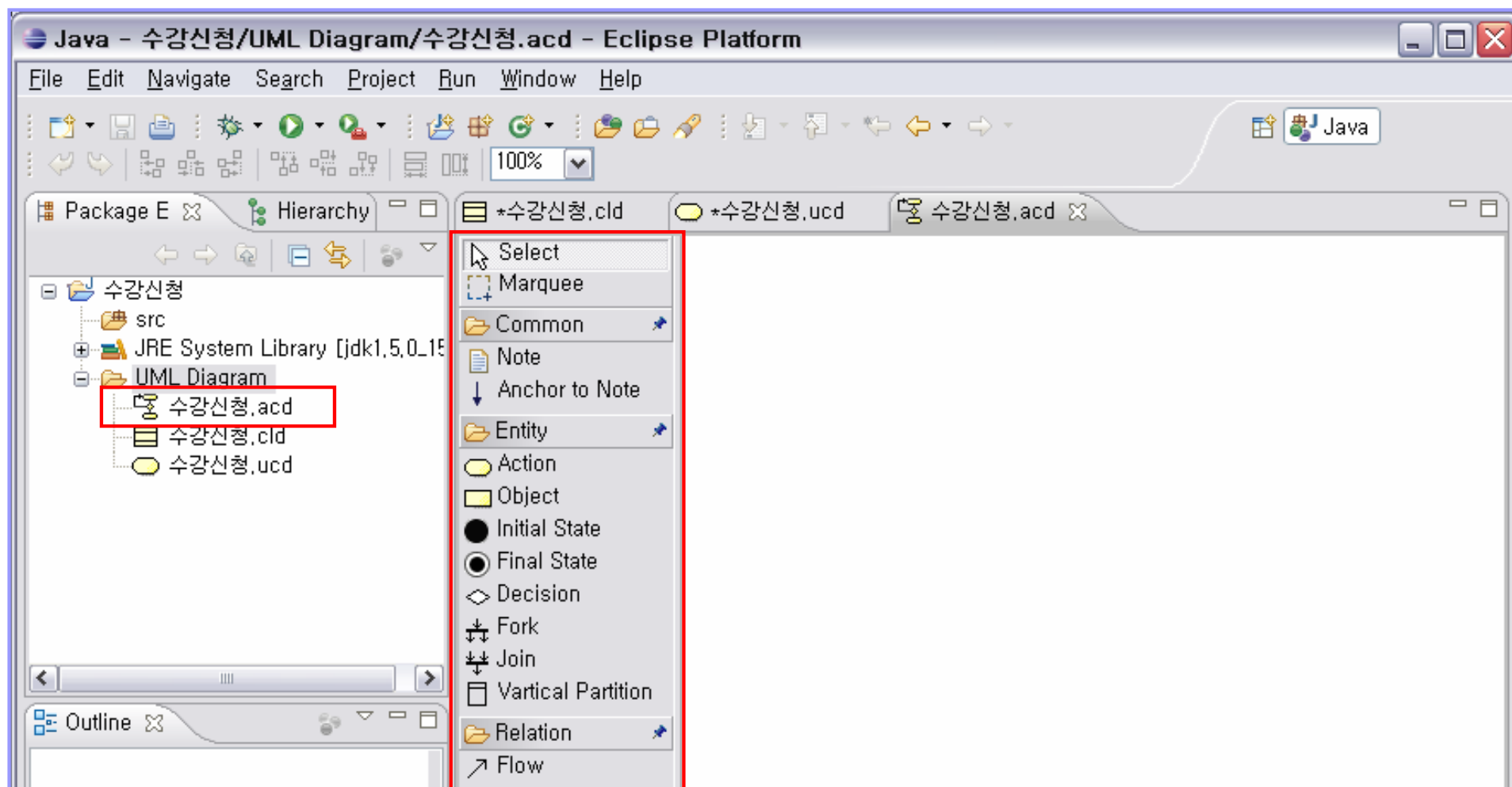
- Activity Diagram을 생성합니다.
 - 확장자가 acd(Activity Diagram)인 파일을 생성합니다.
 - AmaterasUML 하위 메뉴인 Activity Diagram 클릭 -> Next 버튼 클릭
 - UML Diagram 폴더 선택 -> File name 입력 -> Finish 버튼 클릭



3. 도구 기능 소개

3.7 Activity Diagram - 다이어그램 생성 (3/3)

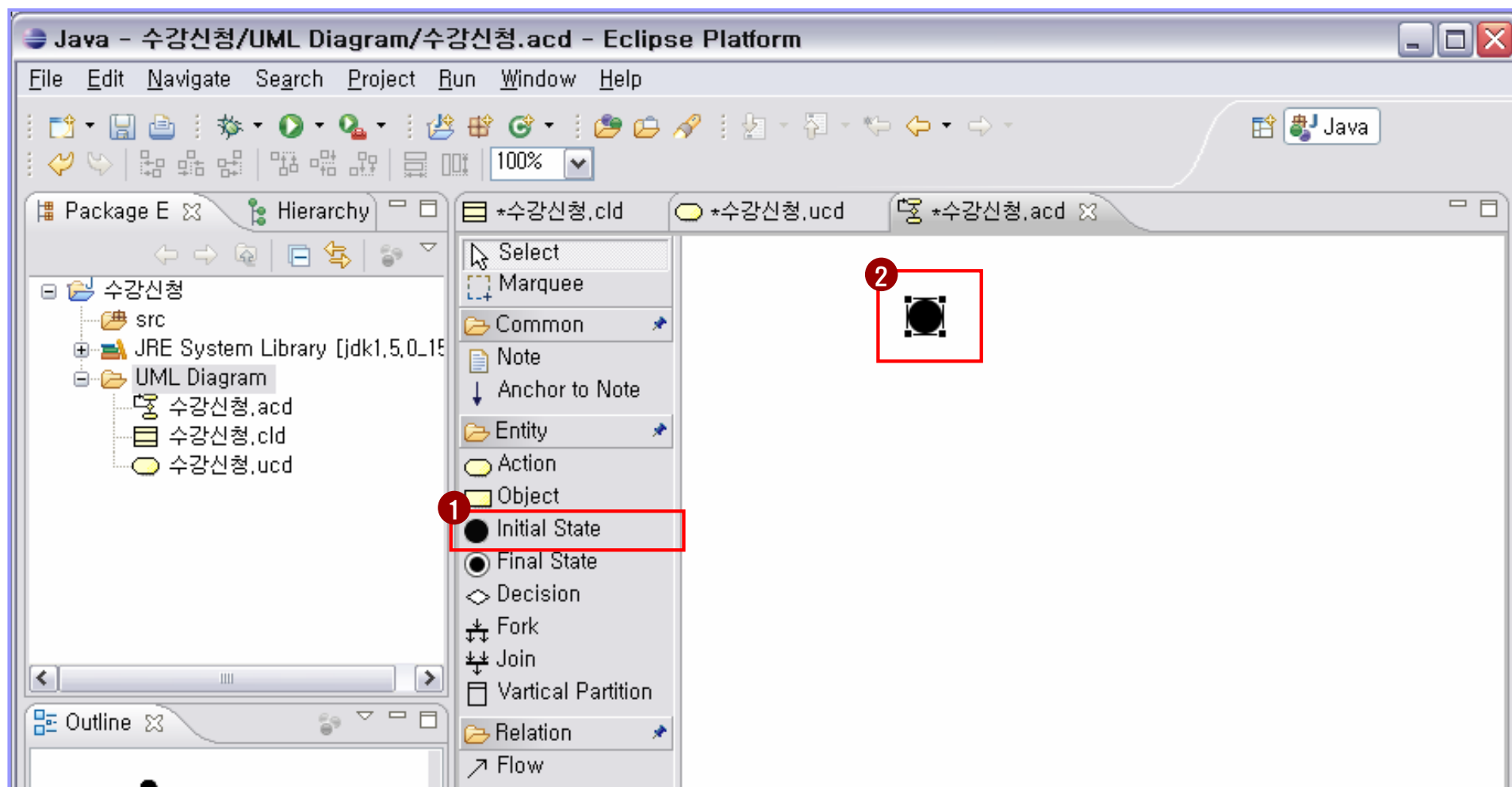
- Activity Diagram이 생성되었습니다.
 - 수강신청.acd 파일이 생성되었습니다.
 - Activity Diagram과 관련된 팔레트를 확인할 수 있습니다.



3. 도구 기능 소개

3.7 Activity Diagram - 초기 상태 생성

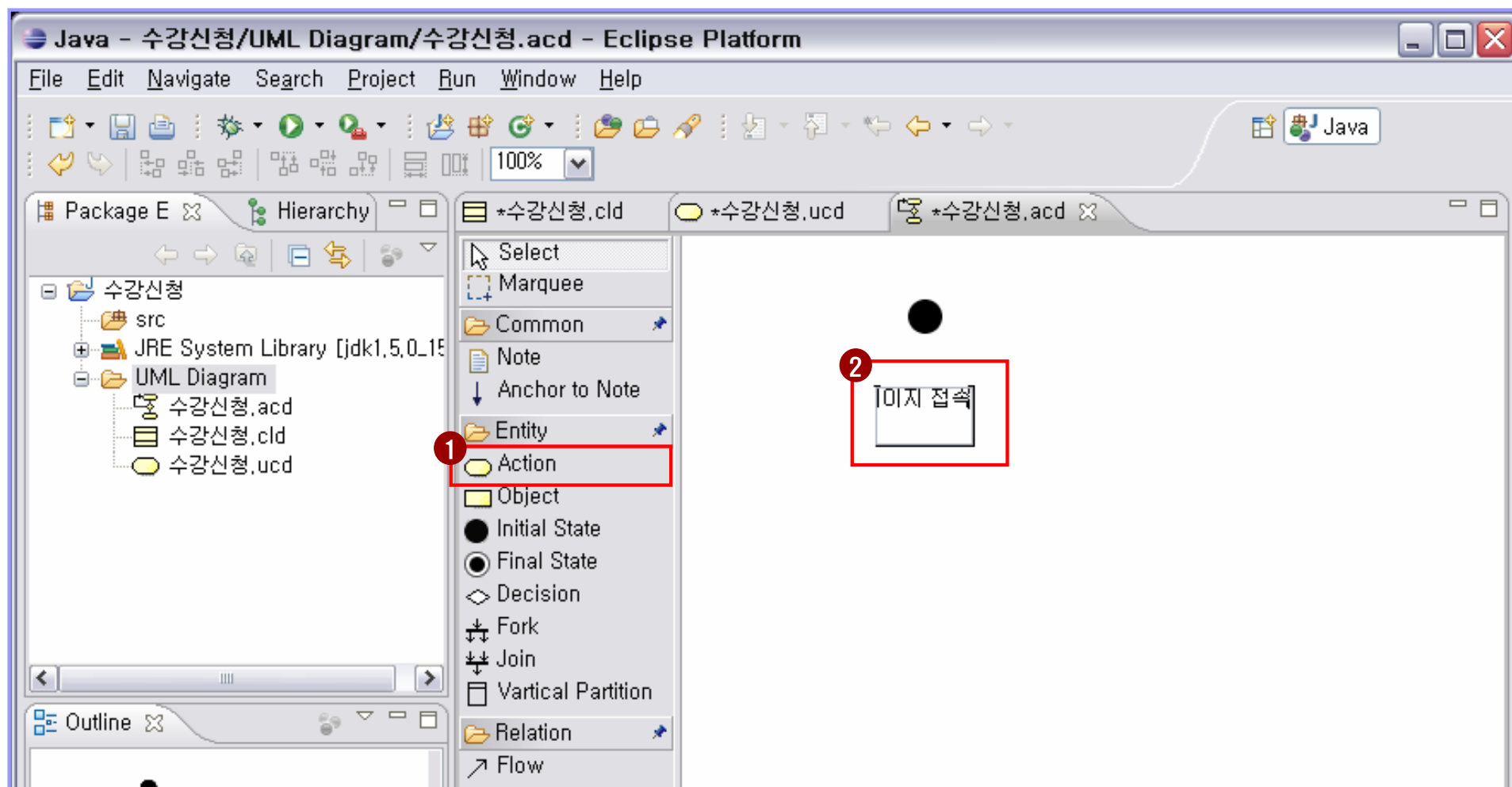
- 초기 상태를 생성합니다.
 - 팔레트를 사용하여 초기 상태를 생성합니다.
 - 팔레트에서 Initial State 클릭 -> 가운데 작업공간 클릭



3. 도구 기능 소개

3.7 Activity Diagram – 작업 생성 (1/2)

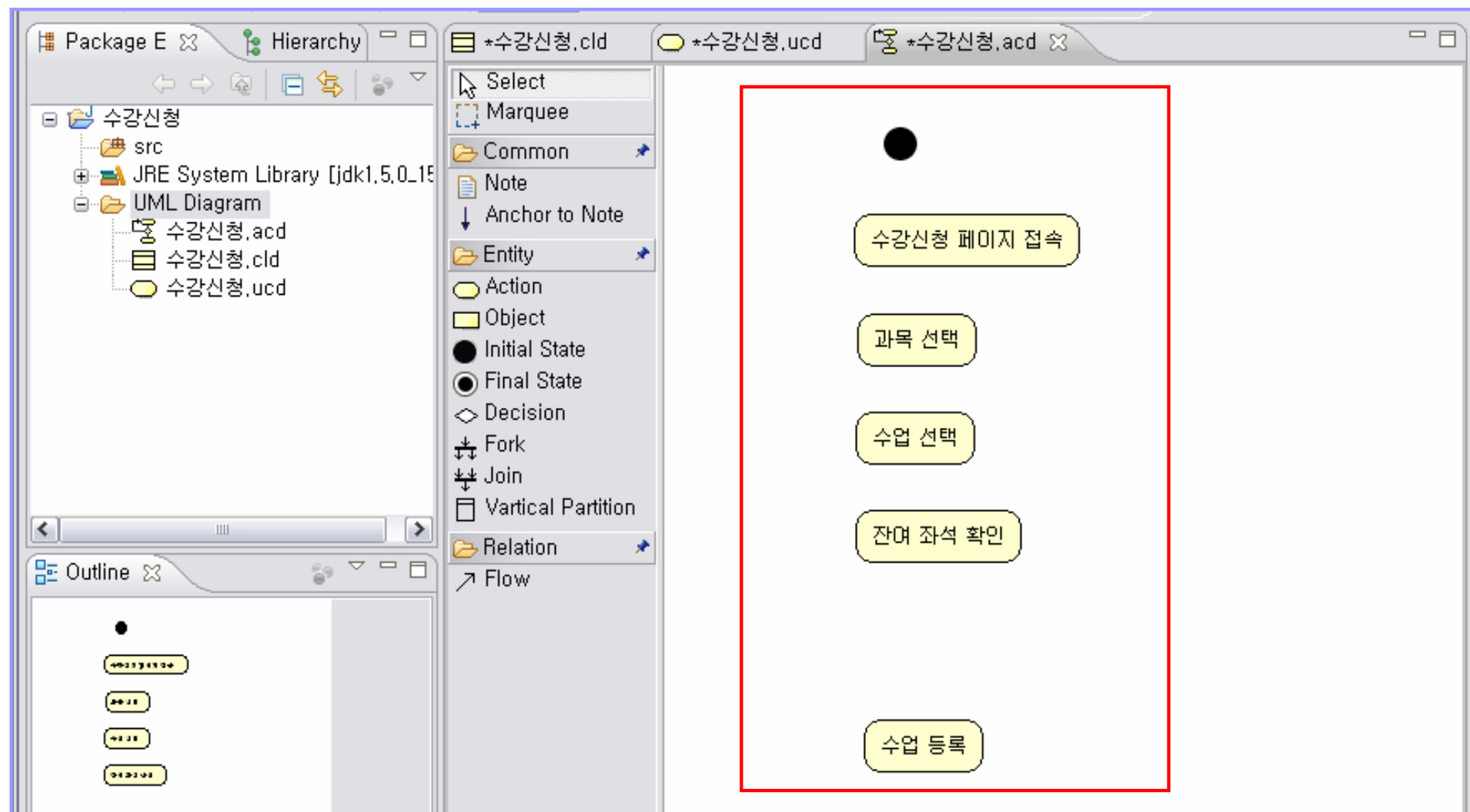
- 작업(Action)을 생성합니다.
 - 팔레트를 사용하여 작업을 생성합니다.
 - 팔레트에서 Action 클릭 -> 가운데 작업공간 클릭 -> Action name 설정



3. 도구 기능 소개

3.7 Activity Diagram – 작업 생성 [2/2]

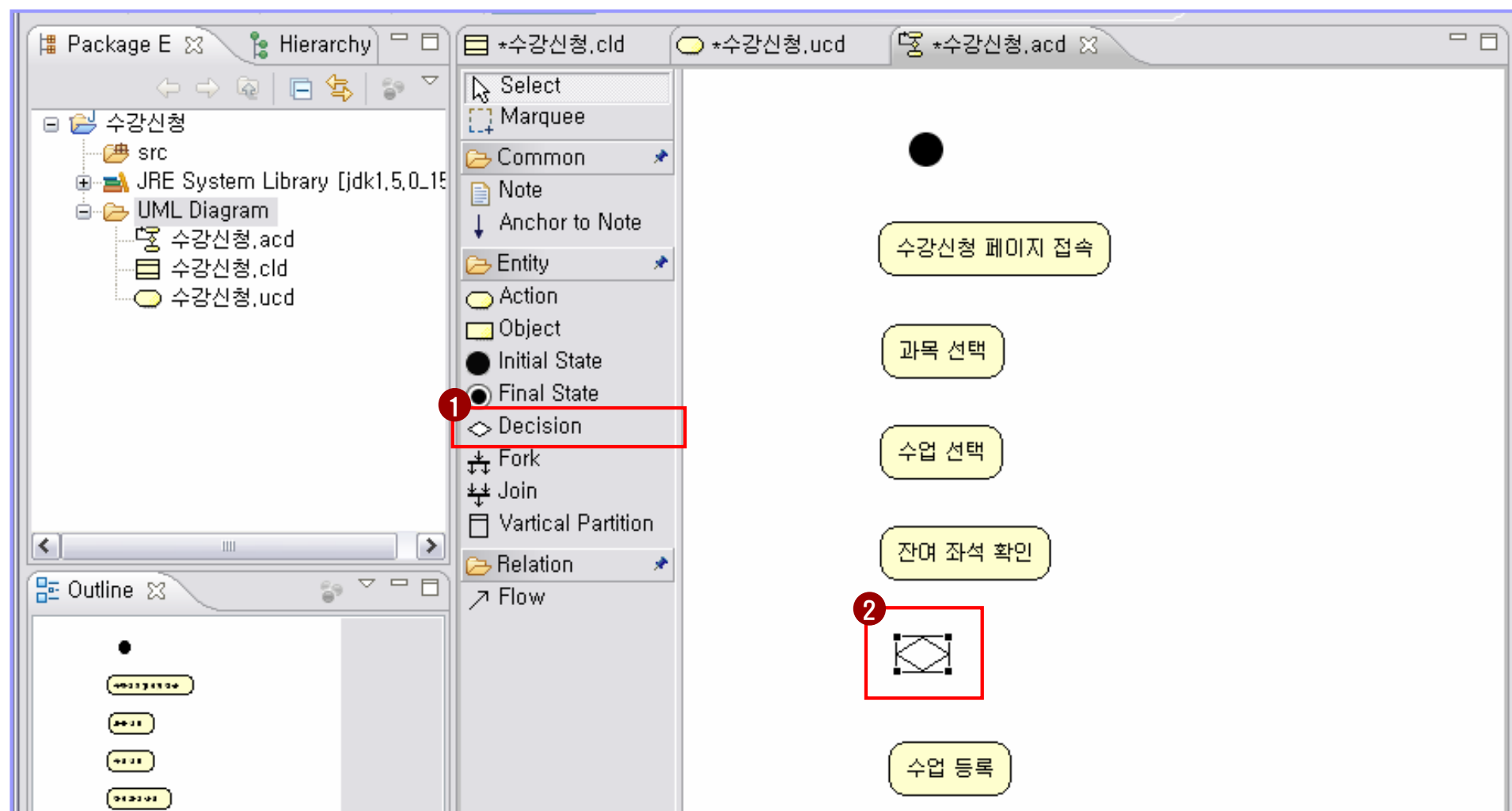
- 이전 페이지와 같은 방법으로 작업을 생성합니다.



3. 도구 기능 소개

3.7 Activity Diagram - 분기 생성

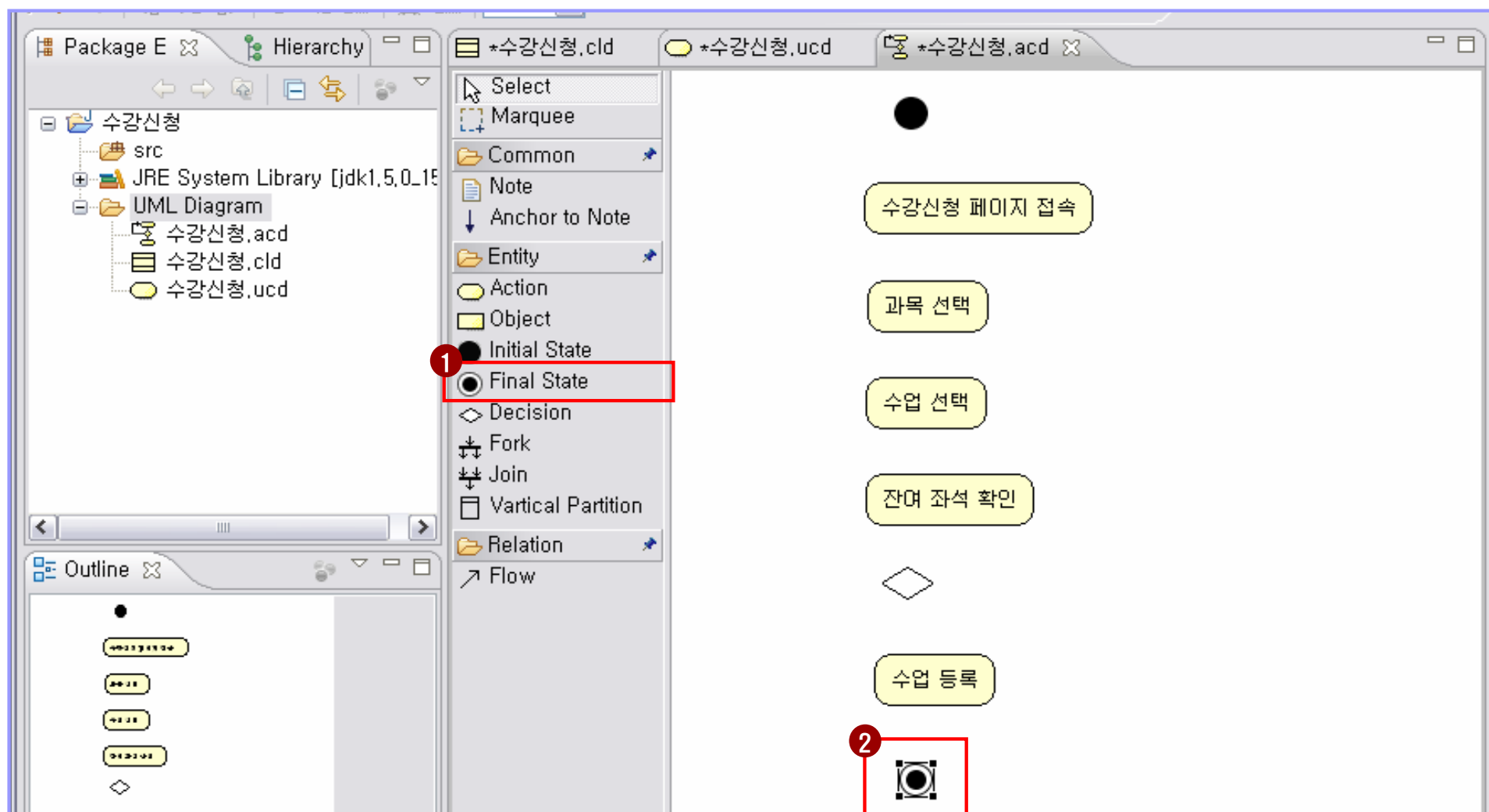
- 분기를 생성합니다.
 - 팔레트를 사용하여 분기를 생성합니다.
 - 팔레트에서 Decision 클릭 -> 가운데 작업공간 클릭



3. 도구 기능 소개

3.7 Activity Diagram - 종료 상태 생성

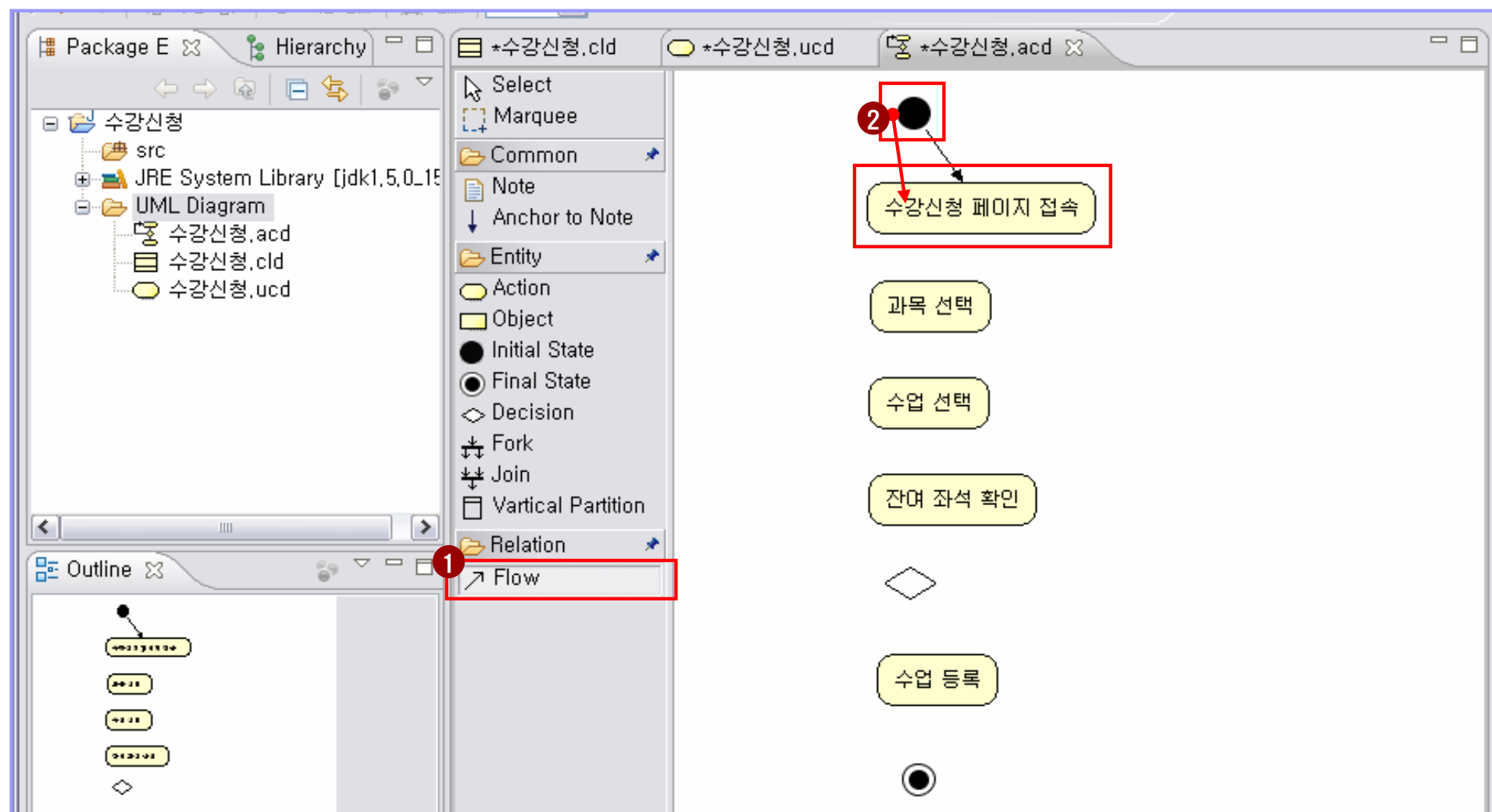
- 종료 상태를 생성합니다.
 - 팔레트를 사용하여 종료 상태를 생성합니다.
 - 팔레트에서 Final State 클릭 -> 가운데 작업공간 클릭



3. 도구 기능 소개

3.7 Activity Diagram – 연결 생성 [1/2]

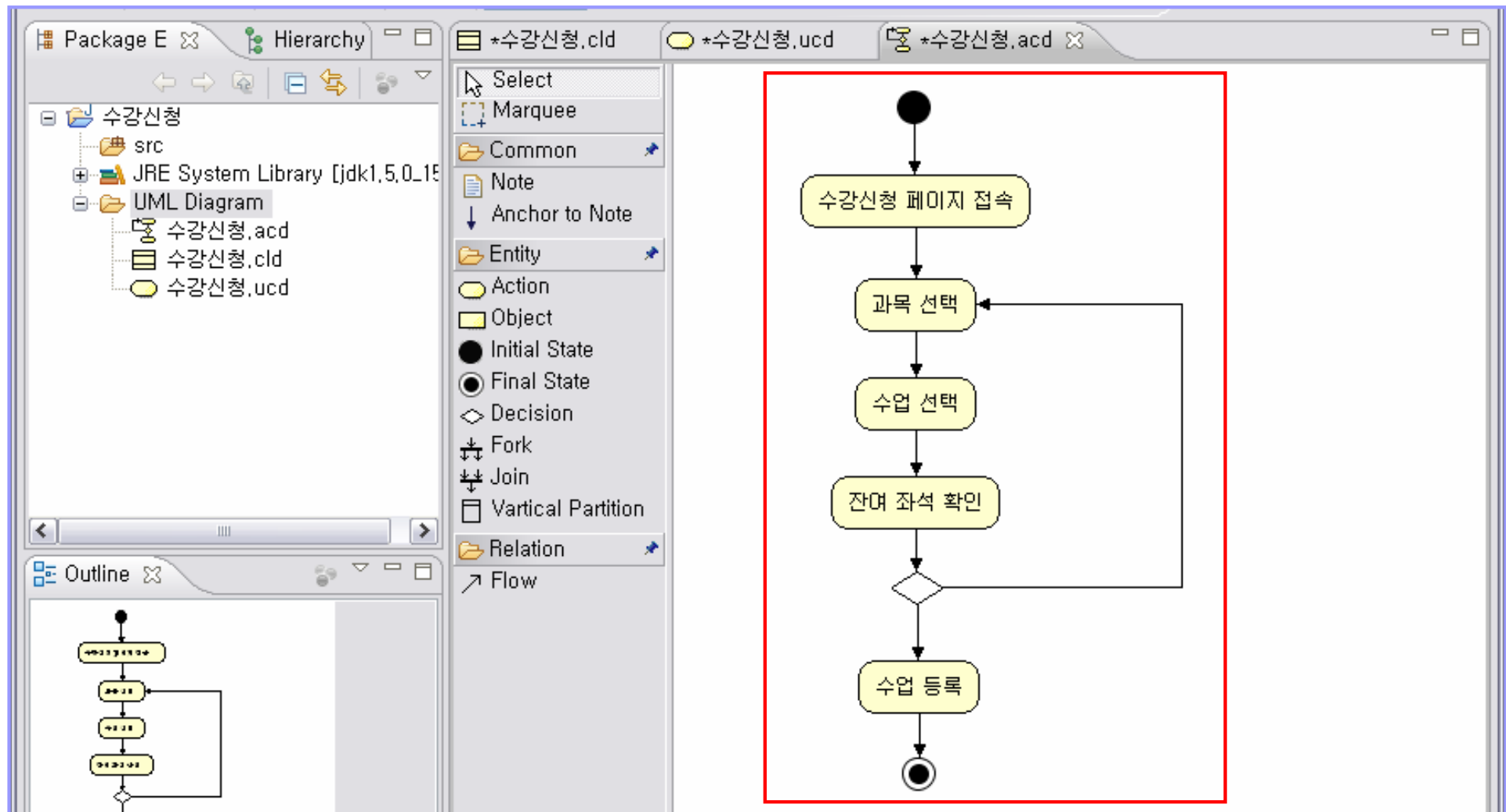
- 연결을 생성합니다.
 - 팔레트를 사용하여 연결을 생성합니다.
 - 팔레트에서 Flow 클릭 -> 초기 상태에서 수강신청 페이지 접속 Action으로 클릭 앤 드래그



3. 도구 기능 소개

3.7 Activity Diagram – 연결 생성 [2/2]

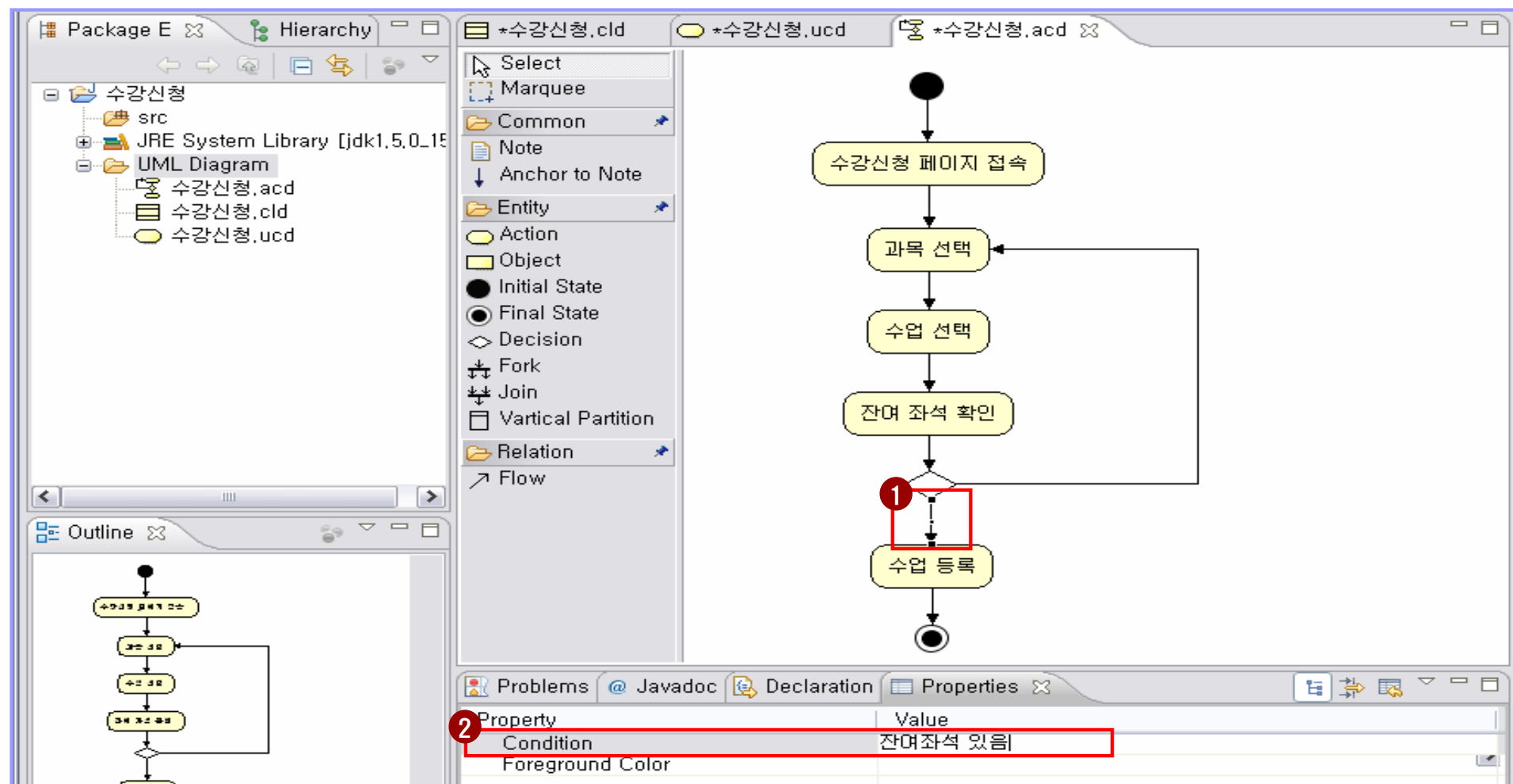
- 이전 페이지와 같은 방법으로 연결을 생성합니다.



3. 도구 기능 소개

3.7 Activity Diagram - 분기 정보 입력 (1/2)

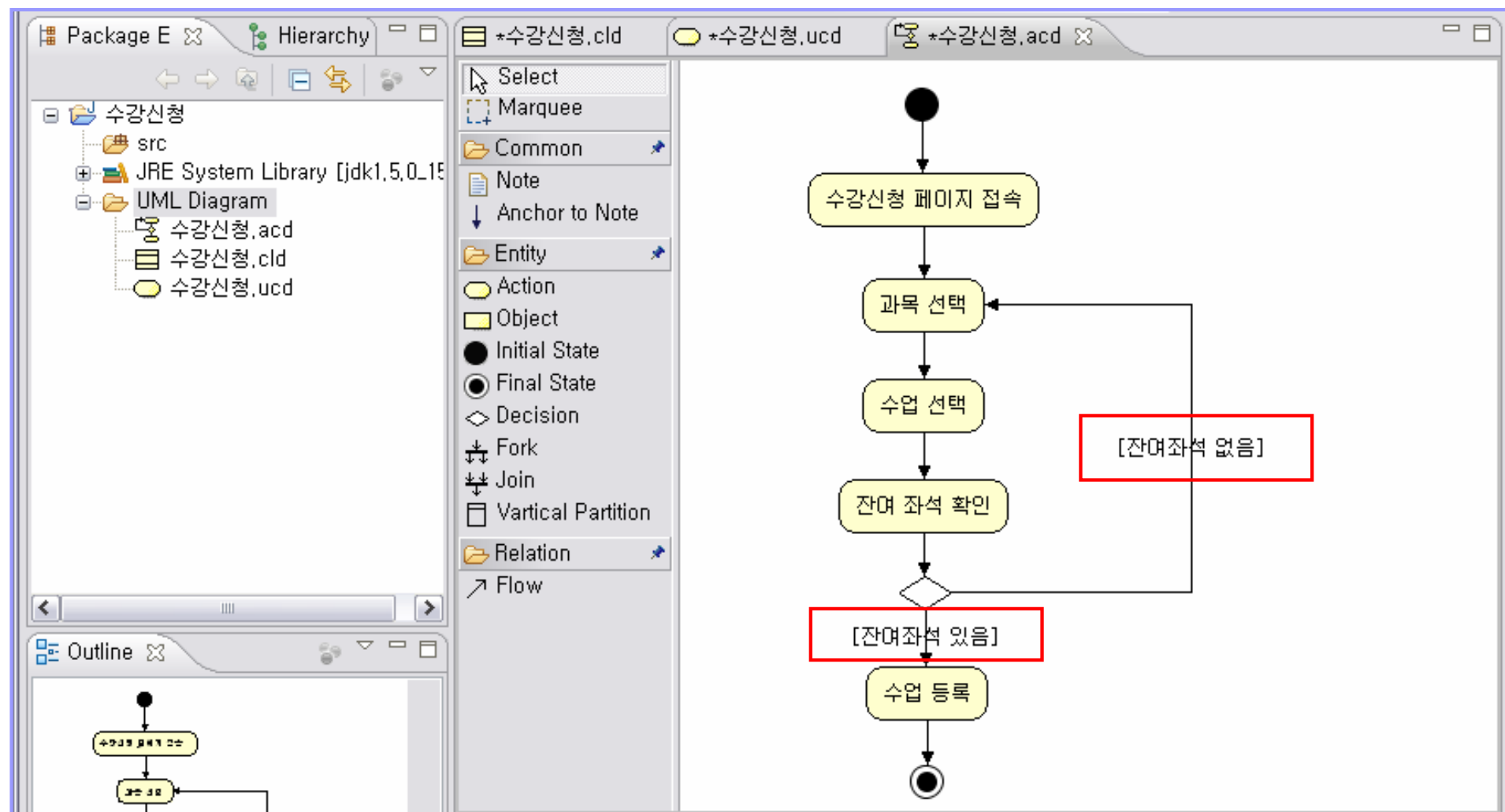
- 분기 정보를 입력합니다.
 - 분기에서 작업으로의 연결에 분기 정보를 입력합니다.
 - 분기에서 수업 등록 작업으로의 연결 클릭 -> Properties View에서 Condition Value에 잔여좌석 있음을 입력



3. 도구 기능 소개

3.7 Activity Diagram – 분기 정보 입력 (2/2)

- 이전 페이지와 같은 방법으로 분기 정보를 입력합니다.



4. 도구 활용 예제

4. 도구 활용 예제

세부 목차

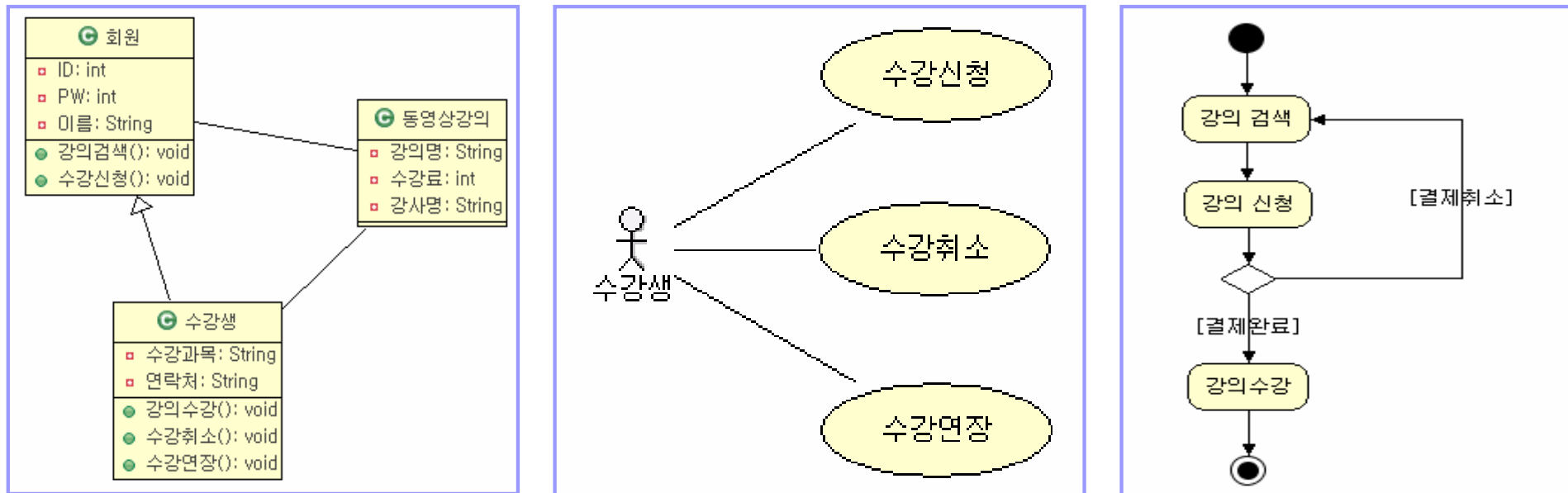
- 4.1 예제 설명
- 4.2 Class Diagram
- 4.3 Use Case Diagram
- 4.4 Activity Diagram

4. 도구 활용 예제

4.1 예제 설명 (1/2)

예제 소개

- A 업체는 S 회사로부터 동영상 강의를 제공하는 사이트의 개발을 의뢰받았습니다.



동영상 강의 사이트의 요구사항

- 회원으로 등록된 사람만이 수강신청을 할 수 있습니다.
- 동영상 강의를 수강하는 수강생 정보는 회원정보와 별도로 관리됩니다.
- 동영상 강의 수강기간은 원칙적으로 결제일로부터 한달간이며, 수강을 연장하기 위해서는 수강 연장 신청을 하여야 합니다.
- 수강신청 후 3일 내에 결제되지 않은 경우에는 수강신청이 취소됩니다.

4. 도구 활용 예제

4.1 예제 설명 (2/2)

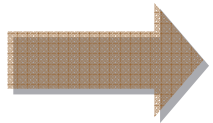
- A 업체의 몇 가지 이슈 상황

1

고객의 요구사항을 받은 A팀은 팀원 각자가 유스케이스 다이어그램을 작성한 뒤, 회의를 통해서 유스케이스 다이어그램을 완성하기로 하였으나, 팀원이 쓰는 표기법이 각자 달라 의사소통에 어려움이 발생하였습니다.

2

시스템이 제공하는 많은 기능 및 액터 그리고 이들 간의 관계를 누구나 알아보기 쉽게 표현하기가 어렵습니다.



공개 UML Diagram 제작 도구인 AmaterasUML을 도입하여 UML을 설계하고 관리하기로 결정하였습니다.

- AmaterasUML 도입 효과

1

표준으로 정의된 표기법을 제공함으로써 의사소통을 원활히 하는데 도움을 줍니다.

2

시스템 전체를 하나의 그림으로 볼 수 있으므로 시스템에 대한 이해도를 높일 수 있습니다.

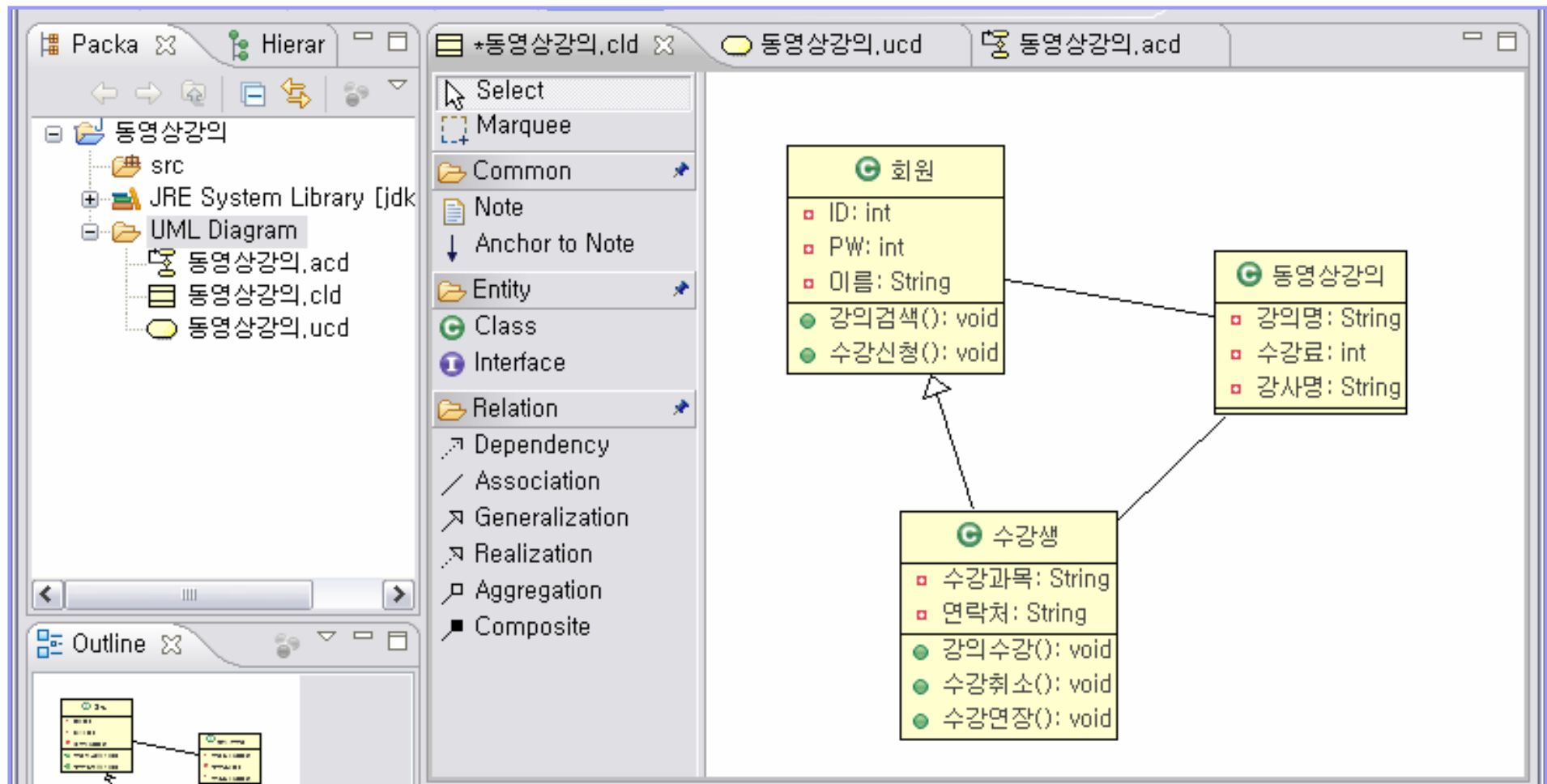
2

팔레트를 제공하여 UML Diagram 작성을 쉽게 할 수 있습니다.

4. 도구 활용 예제

4.2 Class Diagram

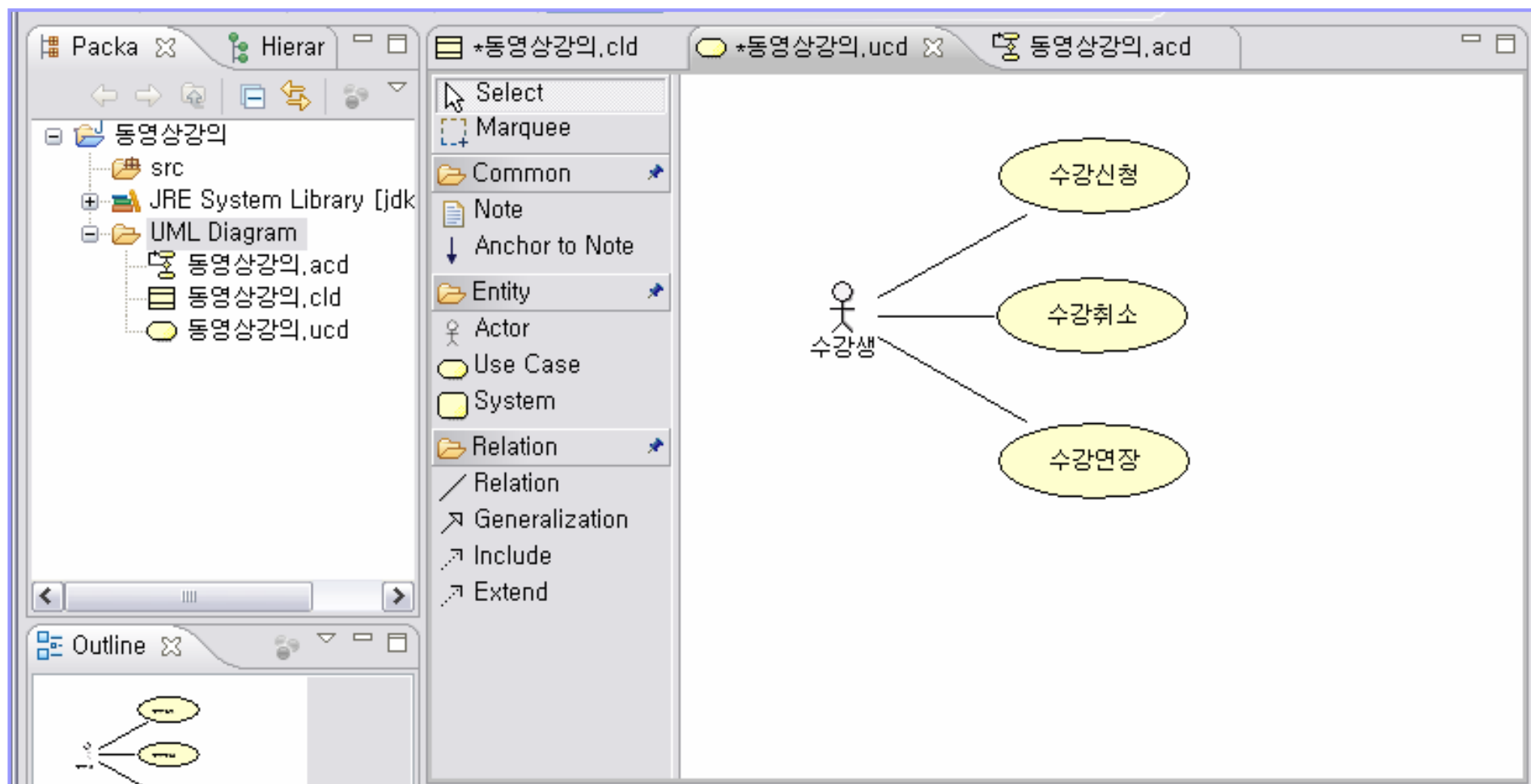
- Class Diagram을 사용하여 설계합니다.
 - 객체지향 시스템 모델링을 위해 사용됩니다.
 - Class Diagram 작성 방법은 21 ~ 35페이지를 참조합니다.



4. 도구 활용 예제

4.3 Use Case Diagram

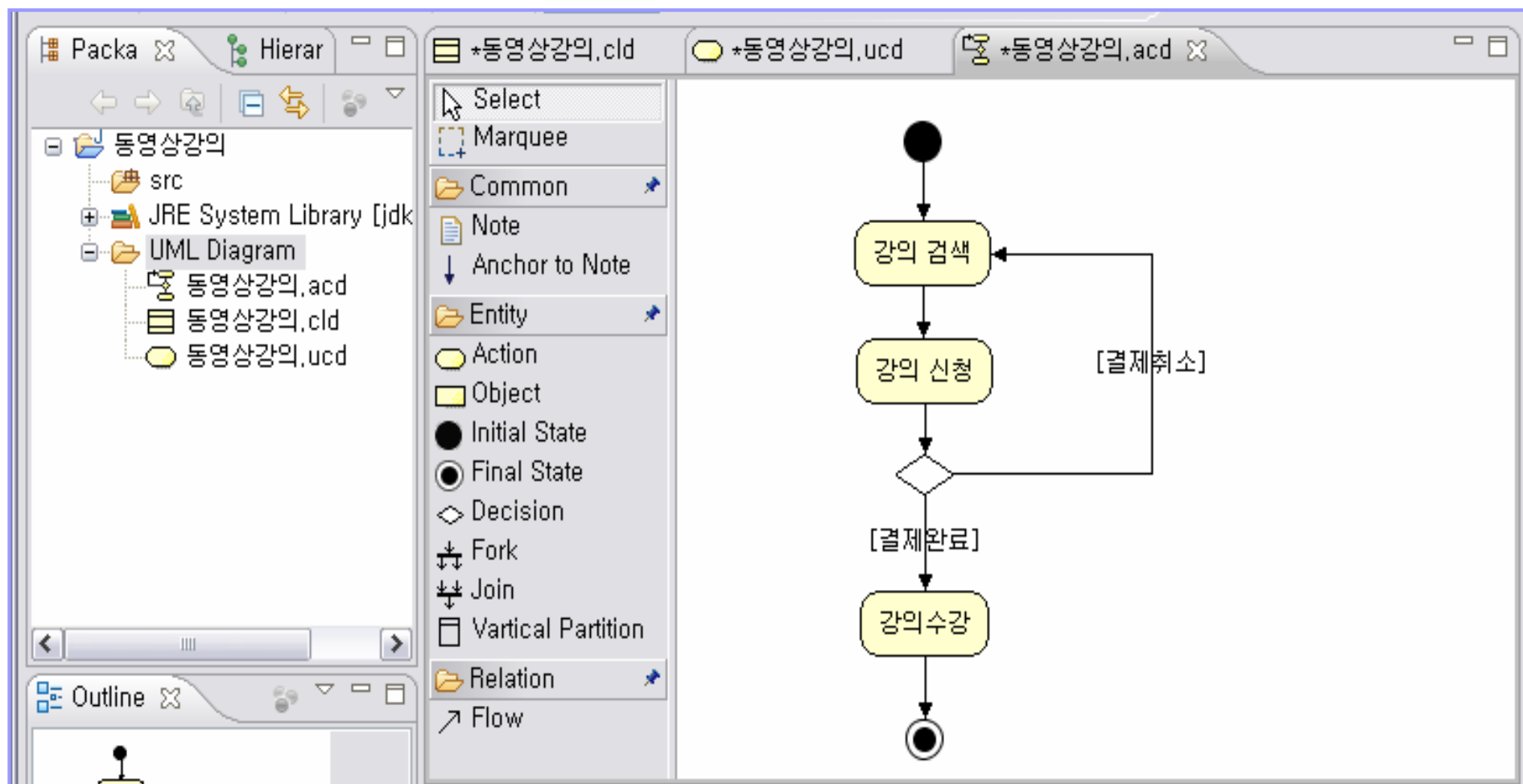
- Use Case Diagram을 사용하여 설계합니다.
 - 사용자의 행위를 기반으로 한 Use Case Diagram을 사용합니다.
 - Use Case Diagram 작성 방법은 36 ~ 44페이지를 참조합니다.



4. 도구 활용 예제

4.4 Activity Diagram

- Activity Diagram을 사용하여 설계합니다.
 - 프로세스의 업무 절차를 표현할 수 있습니다.
 - Activity Diagram 작성 방법은 45 ~ 56페이지를 참조합니다.



5. FAQ

5. FAQ

질문1) AmaterasUML은 무료로 사용할 수 있습니까?

➡ 답1) 예, AmaterasUML은 Eclipse의 Plug-in 으로 무료로 사용할 수 있습니다.

질문2) AmaterasUML을 사용하기 위해 미리 설치해야 하는 도구가 있습니까?

➡ 답2) AmaterasUML을 사용하기 위해서는 EMF, GEF, UML, EMTF 등이 Eclipse에 설치되어 있어야 합니다.

6. 도구 평가

6. 도구 평가

AmaterasUML은 이클립스 플러그인으로 설치가 간단합니다.

AmaterasUML은 팔레트를 사용하여 UML Diagram 작성이 용이하도록 도와줍니다.

7. 용어집

7. 용어집

- AmaterasUML은 UML Diagram 작성 도구입니다. 이번 페이지에서는 UML Diagram에 대해서 정리를 하고, AmaterasUML에서 제공하고 있는 UML Diagram을 구분하여 표시하였습니다. 더 자세한 내용을 확인하기 위해서는 UML 관련 책을 참조해 주시기 바랍니다.

용어	설명
(사용사례) Use Case Diagram	시스템이 어떻게 사용되는지를 나타내며, 액터(사용자)와 사용사례(사용자와 시스템 사이의 인터페이스)로 구성됩니다.
(클래스) Class Diagram	클래스, 속성, 오퍼레이션, 연관관계 등을 이용하여 시스템의 정적인 관점으로 나타내어지며, 설계자가 생각하고 있는 사물을 그대로 표현합니다.
(패키지) Package Diagram	관련된 클래스를 패키지로 그룹화하여 의존도를 낮추기 위하여 사용합니다.
(순차) Sequence Diagram	인터랙션 다이어그램으로 소프트웨어 시스템의 동적인 측면을 모델링하기 위해 사용하며, 특정 작업을 수행하기 위하여 객체의 집합 또는 액터가 교환하는 메시지 교환의 순서를 나타냅니다.
(협동) Collaboration Diagram	인터랙션 다이어그램으로 소프트웨어 시스템의 동적인 측면을 모델링하기 위해 사용하며, 여러 객체들이 협력하여 동작하는 것을 네트워크 형태로 표현합니다.
(상태) State Diagram	시스템에 대한 동적 정보를 나타내는 방법으로, 시스템이나 개별 객체들의 외부에서 보이는 행위를 기술하기 위해 사용합니다.
(액티비티) Activity Diagram	액티비티 단계별로 제어하는 흐름을 모델링하여 시스템의 동적 특징을 표현합니다.
(컴포넌트) Component Diagram	소프트웨어 부품(원시코드, 런타임 라이브러리, 실행파일 등)의 구성을 나타냅니다.
(배치) Deployment Diagram	노드, 컴포넌트, 커넥터 등 시스템의 물리적인 자원 배치를 나타냅니다.