

# 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 작 성할 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

-1 - A 2 - D - A	Tool Bar , balantan	in her b
noe 🛛 🗴 Hain (Design Moor)		Nodal Explorer 🗆 0
tabon 🖉 Man 📓 Man		11日 二日・名 ・ ・
alact		🖉 🖨 🍞 Unbilled
ubsystem		<ul> <li>B) Construction of the Cate Model</li> <li>B) Construction of the Cate Model</li> <li>B) Construction of the Cate Model</li> </ul>
ackage		🛞 🧱 < <design vadel=""> &gt; Design Madel</design>
		🖶 🛅 < <mplementationmodel>&gt; (mplementationModel&gt;&gt; (mplementationModel&gt;&gt; (mplementation))</mplementationmodel>
Tool Box		🕷 📴 < <deploymentmodel>&gt;Deployment M</deploymentmodel>
		Madal
ignal	· · · · · · · · · · · · · · · · · · ·	INIOGEI
xception	Mate Minutesu	Explorer
ort.	Main Window	
lw1		
saociation		The Model Explorer The Diversion Explorer
PrectosdAssociation		Description of the second seco
Aggregation		(UM ClassDiagram) Main
Composition		General
ieneralization		• Nano Man
Sepandancy		e Disgranifyps
ledication		• Lessinospan <u>C</u>
Aspocation Class		The second se
Connector		
bbject		2 Droportion
rk - c		Properties
a		10.33
19:49(20) Add-In "Generator Addin" menu is registered successfully. 19:49(20) Add-In "Java Addin" menu is registered successfully.		<i><i>c</i></i>
9.49.20] Add-In "Pattern AddIn" menu is registered successfully.	I Output I	



# StarUML 설치

#### • 아래의 사이트에서 StarUML을 다운 받아 설치

<u>http://sourceforge.net/projects/staruml/files/staruml/5.0/staruml-5.0-with-cm.exe/download</u>





# StarUML : 다이어그램 생성 방법

• Step Step 1 (계속)

- StarUML을 실행하고 새 Project를 생성
- Approach는 'Default Approach'를 선택

Approaches Open Files Recent Files	
Approaches	88 12
4+1 View Model	L Empty Project
Default Apporach for Stari ML	
Set As Default Approach	



# 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

Toobox 🛛 🗘 🛛	Activity sagram1 (Analysis Model: ActivityGraph1)	Model Explorer D
Annotation	[고 Man 고 Main 그 ActivityDiagram]	1111 篇· 贺 9 8
Activity -		A Good States
Select		🛞 👔 < cuseCaseModel>> Use Case Mode
ActionState		🖶 📴 < <analysismodel>&gt; Analysis Model</analysismodel>
😑 SubactivityState		Main
<ul> <li>InitialState</li> </ul>		E 2 ActivityGraph1
FinalState		
) - Synchronization		i Set < <designmodel>&gt; Design Model</designmodel>
Decision		
@ Flow Final		<deploymentmodel>&gt; Deployment</deploymentmodel>
Chiert Flow		
Signal Accept State		
Signal Send State		201 Bodel Fabrier, 1231 and an Exactor
1 Transition		Properties D
Contranction		(UMLActivityDiagram) ActivityDiagram1
1 Sumlane/Vector/		Dame Activity Clancers 1
C information (rerucal)		DiagramType
- Swittenio(Honzonkai)		
		+ -



• 사용사례

StarUML의 기능을 소개 위한 간단한 수강신청 시스템

• 수강신청 요구사항





#### • Use Case Diagram 그리기

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



#### • Use Case Diagram Tool Bar

UseCase	Select	Diagram 요소를 선택합니다.
R Select	Package	모델 요소들을 노리적으로 그룹하 한 때 사용한니다
🔁 Package	i denago	
🔾 UseCase	UseCase	시스템이 제공하는 기능을 의미합니다.
옷 Actor	Actor	시스템을 사용하는 사용자나 외부 시스템을 의미합니다.
Association	Association	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다.
 介 Generalization 介 Dependency	DirectedAssociation	유스케이스와 액터간에 연관 관계가 있을 때 사용합니다. 종속, 포 함등의 관계를 표시할 수 있습니다.
	Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용 <mark>합니다.</mark>
Include		어떤 유스케이스를 위해 다른 요소의 존재가 요구 되어지는 의존
£û Extend	Dependency	적인 관계를 의미합니다
System Boundary	Include	어떤 유스케이스를 수행하는 데, 반드시 수행해야 하는 유스케이 스를 나타내기 위해 사용합니다.
	Extend	한 유스케이스가 특정 시점에 여러 가지 형태로 분류될 경우에 사 용합니다.
	System Boundary	시스템과 외부 시스템의 경계를 의미합니다.



• Use Case Diagram – Main 선택





• Use Case Diagram – ToolBox -> Actor 선택





Use Case Diagram – Actor 박스 이름 기입





• Use Case Diagram – UseCase 선택





Use Case Diagram – UseCase 이름 기입





• Use Case Diagram – Association 선택





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





• Use Case Diagram





• 10page의 수강신청 시스템의 요구사항을 Class Diagram으로..

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





#### Class Diagram Tool Bar(1/2)

lass 🔶	Select	Diagram Q스를 서택하니다
😞 Select	Select	Diagram 표도를 전득합니다.
A Subsystem	Subsystem	물리적인 시스템의 부분 또는 전체를 의미합니다.
🔁 Package		
🚽 Class	Package	관련된 클래스들을 모아서 모델링 한 것을 의미합니다.
🔶 Interface		
Enumeration	Class	객체의 Attribute와 Method을를 모델링 한 것을 의미합니다.
n Signal	19 - 20 - 20 19 - 20 - 20	
🔩 Exception	Interface	클래스에서 Method의 선언부분만 모델링 한 것을 의미합니다.
' Port	-	
📙 Part	Enumeration	비디 영의된 없을을 디스트도 가지는 데이터다입을 의미입니다.
Association	Signal	객체가의 비도기전 통시 시승를 의미합니다
	olgital	
	Exception	실행 오류시 Operation에 의해 발생되는 신호를 의미합니다.
	/	
了 Generalization	Port	인터페이스와 클래스의 연결 단자를 의미합니다.
1 Dependency		
1 Realization	Part	클래스 내부의 특정한 부분을 의미합니다.
AssociationClass		



💶 Object

🔟 Link



#### Class Diagram Tool Bar (2/2)

Class 🔶	Association	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 가능합니다.
<ul> <li>Subsystem</li> <li>Package</li> <li>Class</li> </ul>	DirectedAssociation	한 클래스와 다른 클래스가 연관 관계가 있을 때 사용합니다. Qualifier의 사용이 불가능합니다.
-> Interface	Aggregation	한 클래스가 다른 클래스를 포함하는 관계일 때 사용합니다.
Signal     Excention	Composition	한 클래스가 다른 클래스에 완전히 종속되는 관계일 때 사용합니 다.
Port	Generalization	일반적인 요소와 더 구체적인 요소의 관계일 때 사용합니다.
	Dependency	한 클래스의 변화가 다른 클래스의 변화에 영향을 주는 관계를 의 미합니다.
	Realization	인터페이스와 클래스와의 연결에 사용합니다.
_↑ Generalization ↑ Dependency	AssociationClass	클래스와 연과의 연결에 사용합니다.
 Realization ■: AssociationClass	Connector	Part 사이의 연결에 사용합니다.
Connector	Object	특정 클래스의 인스턴스를 의미합니다.
Link	Link	객체 사이의 연결에 사용합니다.



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

n nahar a sa dar	Main (Design Model)	1000	delExplorer	<b>T</b> (1)	📰 🔛 🔄 < <analysismodel>&gt; Analysis Model</analysismodel>
inotation	[ [7] ma [7] ma ]	11	ē1 🔅 .	94 0 0	🖶 📴 < <designmodel>&gt; Design Model</designmodel>
ass in the second s		N N	Die inter	X	Man
a Select	A I CERTIFICATION CONTRACTOR CONT		E E cer	useCaseModel>> Use Case Model	GassDiagram1
J Subsystem	10110101010100010000000000000000000000		Che ce	naivsistilodel >> Anabsis Model	😥 🗇 < <implementationmodel>&gt; Implement</implementationmodel>
Package			<b>GI</b> <51	lesianModeli>> Desian Model	🛁 📃 🗟 < <deploymentmodel>&gt; Deployment I</deploymentmodel>
Class	di tali antistatana si da tadi tali	0.	⊕ <u> </u> 2	lda	
> Interface	The second secon	G ass Diagram		kdd Diagram	2
Enumeration		Vise Case Diagram	\$ 0	Ctrl+X	<b>T</b>
Signal		E Sequence Diagram	四 (	ζοργ Ctrl+C	
Exception	a demotrationer a be blacker	5equence Diagram (Role)	185 E	Saiba EblyV	
Port	SOBE E GERTONO COR E SOE BOBE	Collaboration Diagram	E	Delete From Model Ctrl+Del	Model Explorer 450 Diagram Explorer
Part		Collaboration Diagram (Role)		Jnit )	Properties 0 4
Association	A CONTRACTOR AND A MADE	Statechart Clagram		election Editor - Ctris FE	(UMLClassDiagram) Main
) DirectedAssociation	d statementer tensi de di stat	Activity Diagram	2	igrection colors. Ctri+Fo	E General
Aggregation		🔟 Component Diagram	121 -	Constraints Ctrimpe	Name Main
Composition		🔁 Deployment Diagram		Lagged values Ctriffer	🔒 DiagramType
P Generalization		රිට Composite Structure Diagram	2	++	🕨 🔷 DefaultDiagram 🗹
Dependency		84 Robustness Diagram	(	<u>*</u>	
Realization			2	ava	
AssociationGass				terite the second s	
Connector				chhià Parnauri	
Object	ALL THE THE THE CONTENT OF ALL THE THE	Construction and a lot of the sector of the	♦ IsLeaf		
LIK					
(put		4.A)			



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

Eile Edit Format Model	Yiew Iools Help								
	XINNA	* . 9   9 % .	100% • 0, 6	V 🛄 💽 🗸					
ዥ Tahoma 🔹 🔹	8 · A ./ 3	명 🍅 🏟 🖣 -	🗵 🗵 🔟 🗐 🎾		유 다 다	臣 母	今 1 4	~ 10	움 🗸
foolbox 🛛 🗘 🛛	ClassDiagram1 (Design	n Model)							
nnotation	I A Main A Main	최 ClassDiagram1							
Jass									
Select									
🗄 Subsystem		*							
🛅 Package									
a Class									
📀 Interface									
Enumeration									
5 Sgnal									
Exception									
Port									
🖸 Part									
Association	dama								
↑ DirectedAssociation	112222-00123								
Aggregation	The second second								
Composition	10000000000000								
ー 全 Generalization	1910203121								
A Realization	1000000000000								
AssociationClass	ninge and								
	1010102 00000								
	Constant and a								
	10001033322200								



학교 Class에 Attributes를 입력합니다.

· 학교 Class 선택 -> 마우스 오른쪽 버튼 클릭 -> Collection Editor... 클 릭

• Attributes 탭 -> Insert 아이콘 클릭 -> Edit 클릭-> 마우스 오른쪽 버튼 클릭 -> Name 변경



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



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





- 클래스간의 관계를 표현해 줍니다.
- 집합연관관계(한 클래스가 다른 클래스를 포함하는 관계를 의미합니다.)

- Toolbox의 Class 에서 Aggregation 클릭 -> Class와 Class 마우스로



Elle Edit Format Model View	Icols Hep 이 다 🚜 🗔 🕄 🖓 🕒 🛠	<b>1</b> 100%	- Q Q 🛄 🗗 .		
1⊈Tahoma ▼ 8	· A ./ & R · A		■ > () =	58 6 d 4	 £ _
Toolbox 🛛 🕮 🗙 🔤	Diagram I (Design Model)				
Annotation	Main 최 Main 최 ClassPlanmant				
Class					
2 Select					
🔀 Subsystem					
🔁 Package	학교		교수		
📑 Class	+ 이름 : String	이번 날날	+0)를 : 5tring		
-O Interface	+수소 : String	0			
Enumeration	+addStudent() +removeStudent()				
📑 Signal		100000000			
Exception					
Port	<b></b>		수업		
🔄 Part	3 LAI	1 1 1 1 1 1 1	+이름: 5tring	and the second second	
Association			+수업민호 : int +최대인원 : int		
	+학번 : int	1998 999	+addStudent()		
			+removeStudent()		
Composition					
Ĵ Generalization					
1 Dependency					
1 Realization					
P: AssociationClass					
Z Connector					
🚍 Object					
🔟 Link					



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

Ele Edit Format Mode	View Iools Help				
		* 🤮 🖓 🖂 🛠 . 🛛 100'		×	
型 Tahoma	• 8 • A ./ 🕅	🖪 🐂 🏠 🗟 · 🗵 🗷	[□] □ ■ > () 匹 ∞.		<u>타</u> 4만 1000 중
Toolbox 4	ClassDiagram1 (Design	Model)			
Annotation	Nain 🖓 Main	A ClassDiagram1			
Class		Contraction of the second			
A Select	14/2010/2018 130				
E Subsystem					
🛅 Padkage	Distances of	학교	교수	14.446.0	
📃 Class	20 000 000000	+이름 : String	+0[름: 5tring	121010	
-💛 Interface	44.147.52.52				
🧮 Enumeration		+removeStudent()			
🖬 Signal					
Exception					
🔁 Port	Re les miner		수업	in one of	
Part		5hAll	+미를 : String		
Association	[4] 2. 23. 33 (3) (3)		+추립인호 : Int	B1381 U	
	C. In the second	+학변 : int	+addStudent()	1	
Aggregation			+removeStudent()		
Composition					
了 Generalization	12.00.00.00				
Dependency					
🕂 Realization	18 18 HE 18				
E AssociationClass					
Z Connector					
🚍 Object					
Junk	And the second				



▪ 수강신청 시스템의 요구사항을 Sequence Diagram으로 표현하면





#### Sequence Diagram Tool Bar

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



32

Sequence > Select 🖃 Object → Stimulus

▪ 수강신청 시스템의 요구사항을 Sequence Diagram으로 표현하면





• 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

![](_page_35_Figure_1.jpeg)

![](_page_35_Picture_2.jpeg)

# 연습 과제 2-3: 순서 다이어그램

![](_page_36_Figure_1.jpeg)

![](_page_36_Picture_2.jpeg)

Eiguro B-11-9 EimCito undato/) Bobaulor