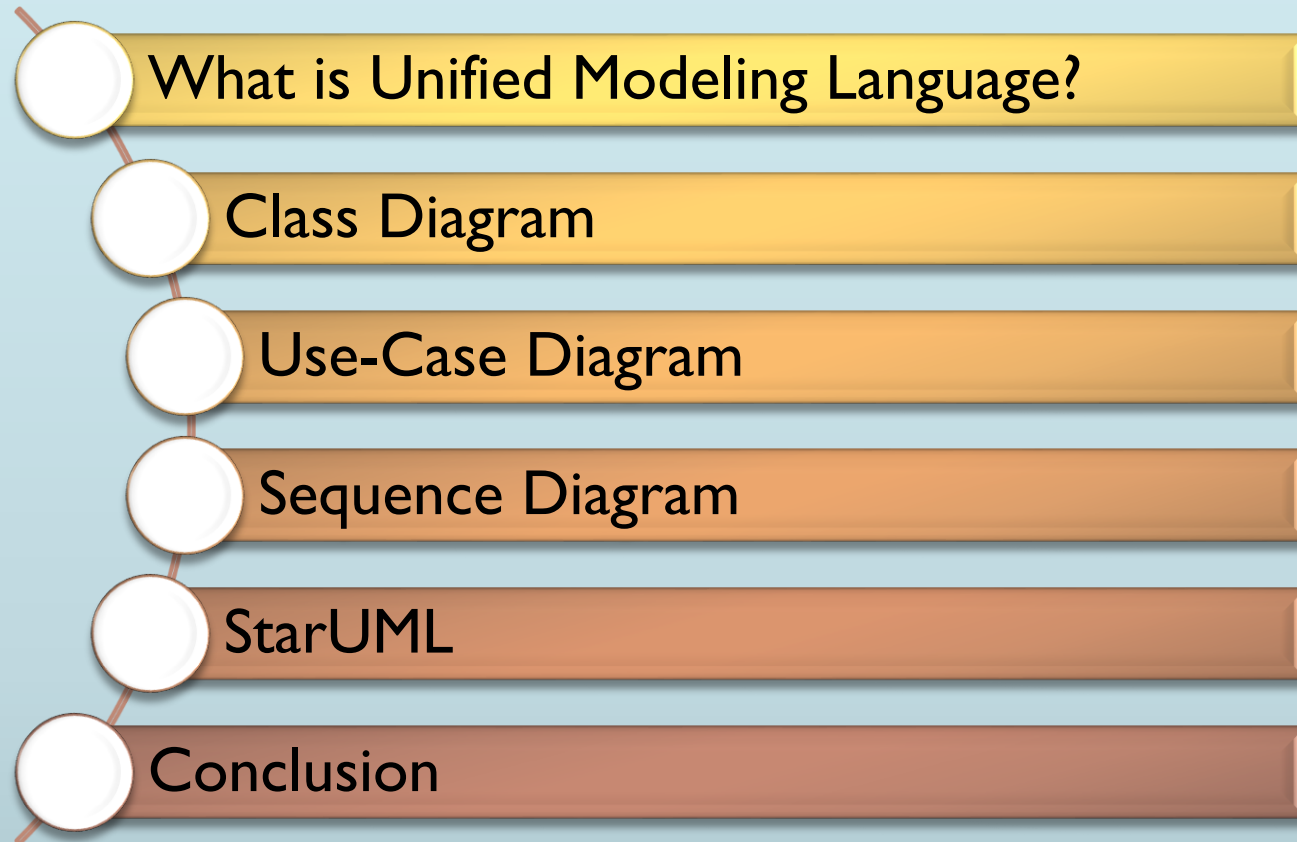


# Unified Modeling Language

유 용 길

# 목차

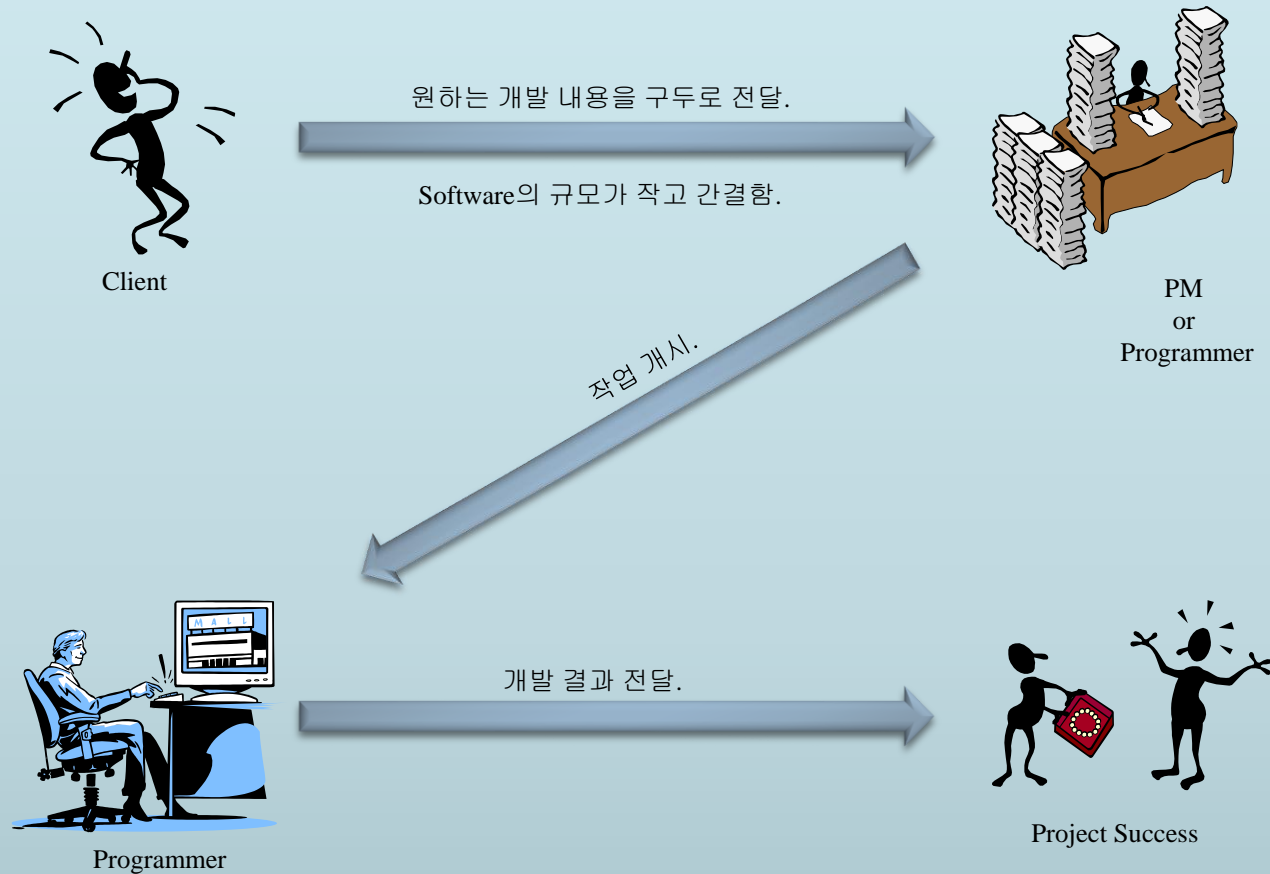
---



- What is Unified Modeling Language?
- Class Diagram
- Use-Case Diagram
- Sequence Diagram
- StarUML
- Conclusion

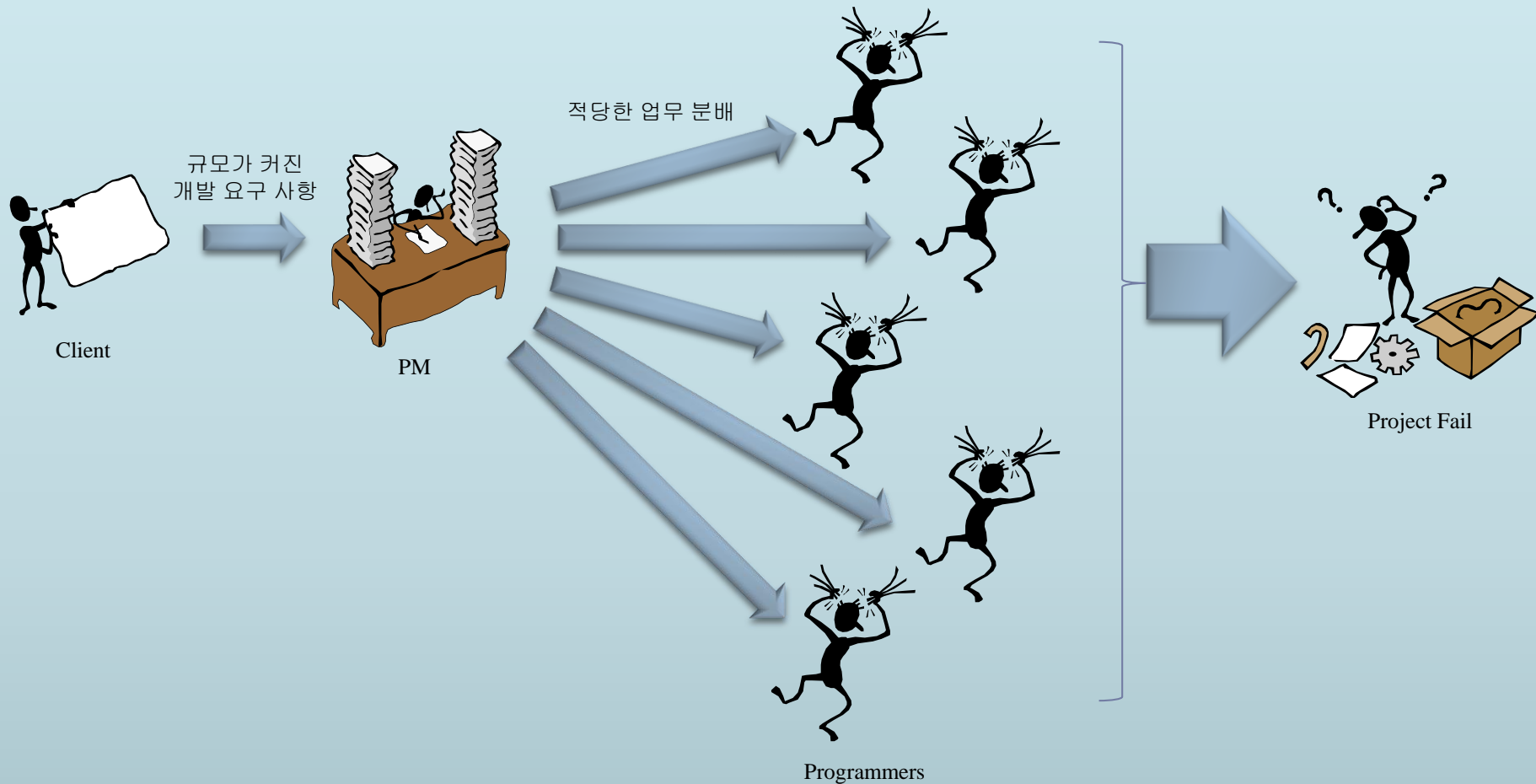
# What is Unified Modeling Language?

- 기존의 Software 개발 구조



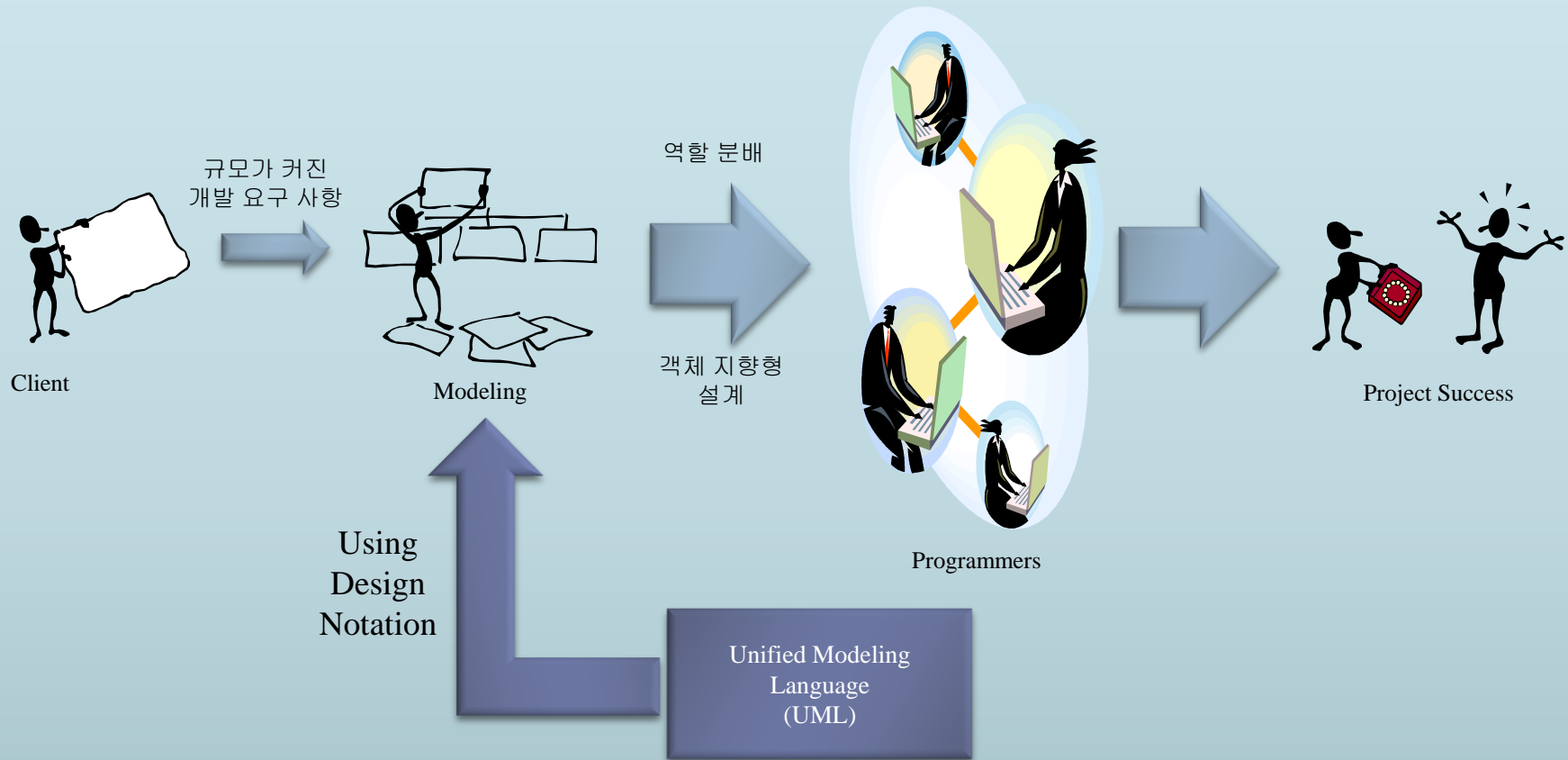
# What is Unified Modeling Language?

- 기존 Software 개발 구조의 문제점



# What is Unified Modeling Language?

- 최근 Software 개발 구조



# What is Unified Modeling Language?

---

- History of UML

- Tree Amigos(Grady Booch, James Rumbaugh, Ivar Jacobson)에 의해 94년 최초 작성.



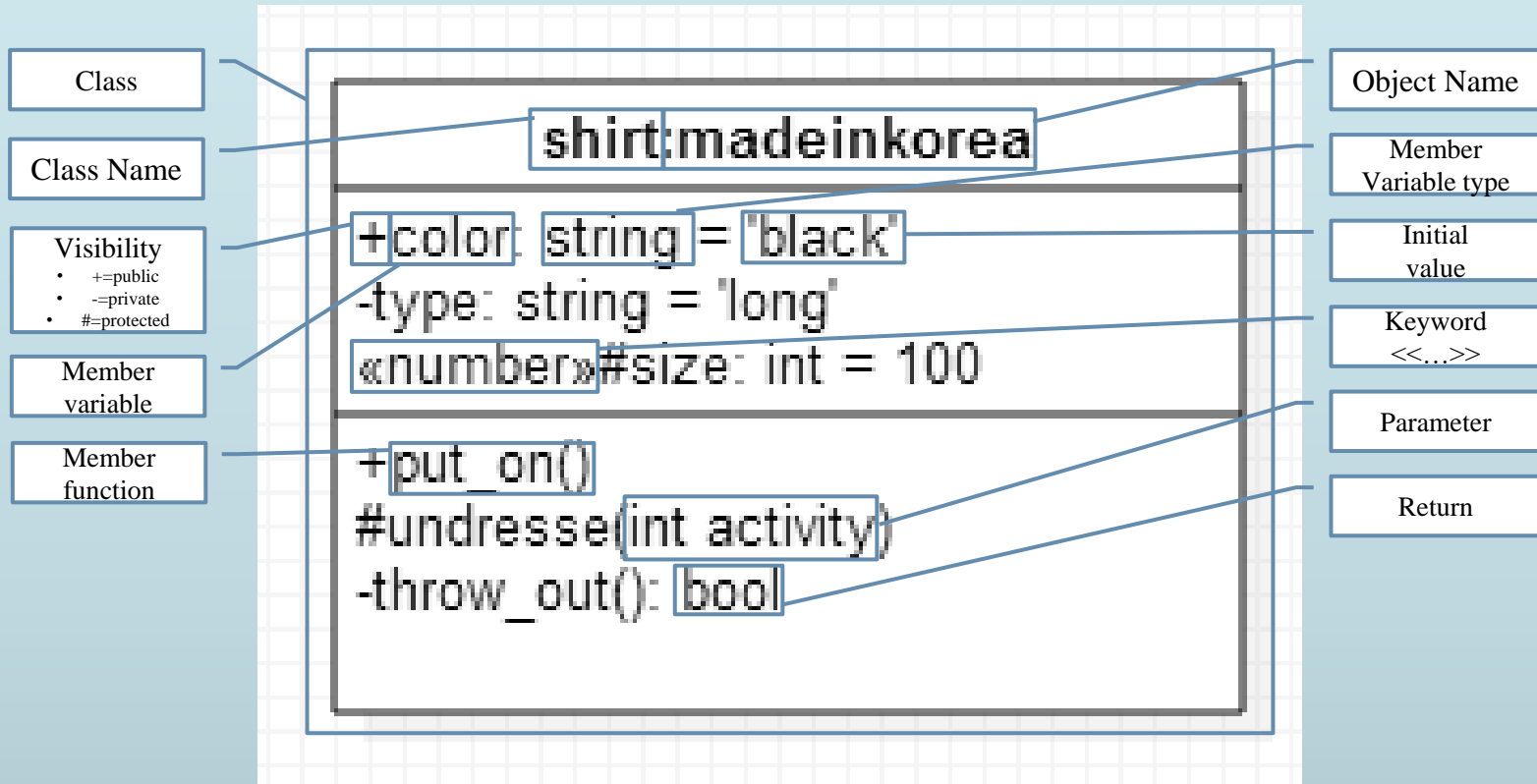
- Grady Booch가 세운 Rational Software Corporation에 James Rumbaugh와 Ivar Jacobson이 각각 94년과 95년에 입사하여 지속적인 개발 개시.
- UML 컨소시엄이 설립되고 97년 정식 버전 1.0을 배포.



- 배포 직후 OMG(Object Management Group)이 표준 모델링 언어 제안서 제출 요구.
- 97년 말 UML1.1 배포와 함께 표준 모델링 언어 상정.
- 현재 UML2.3 배포 및 사용 중.

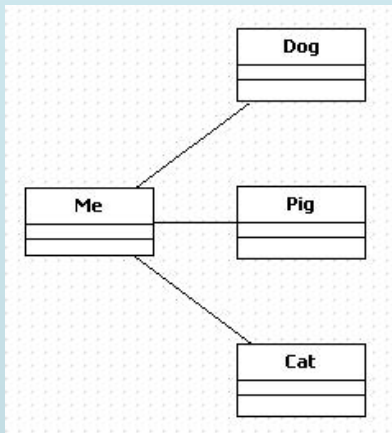
# Class Diagram

- Class

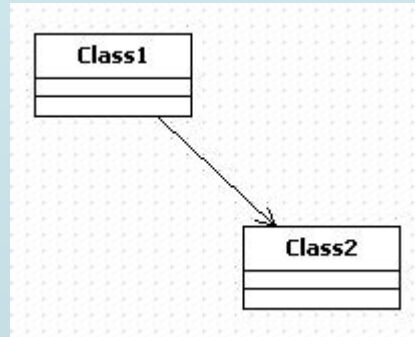


# Class diagram

- Class Association

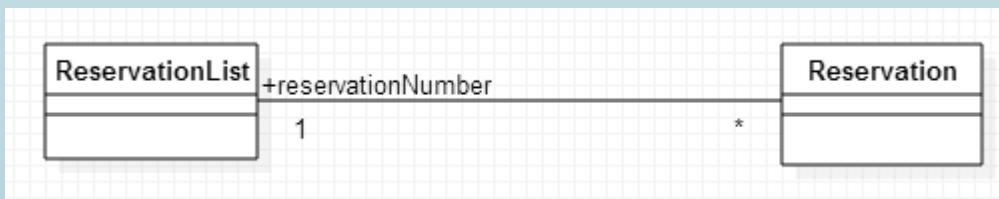


<Association>



<Directed Association>

- Multiplicity

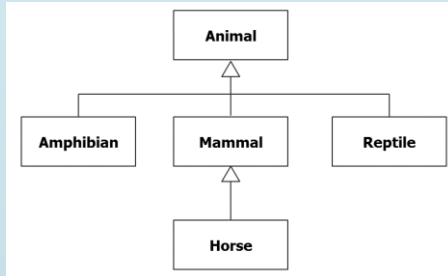


- 0 = Zero 1 = One \* = Many .. = Or , = And

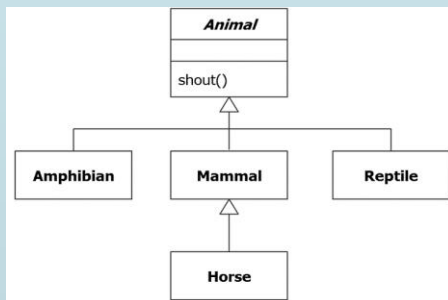


# Class diagram

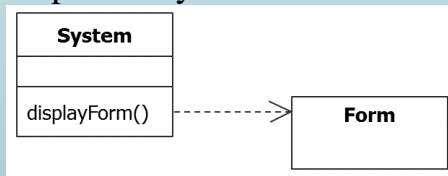
- Inheritance



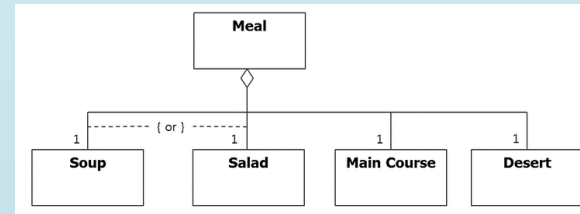
- Abstract



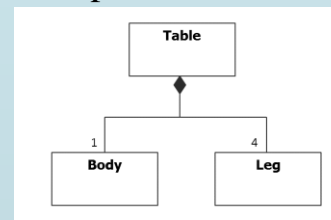
- Dependency



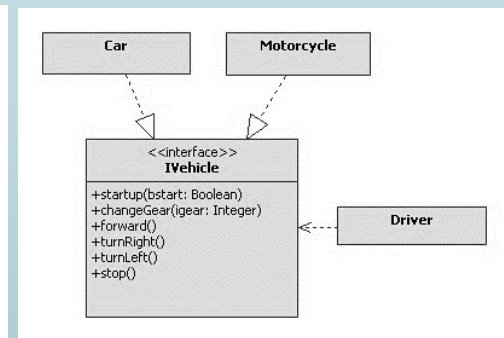
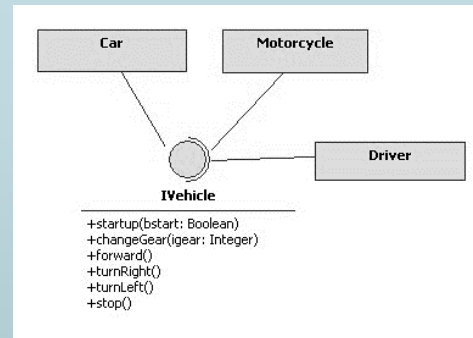
- Aggregation



- Composite



- Interface



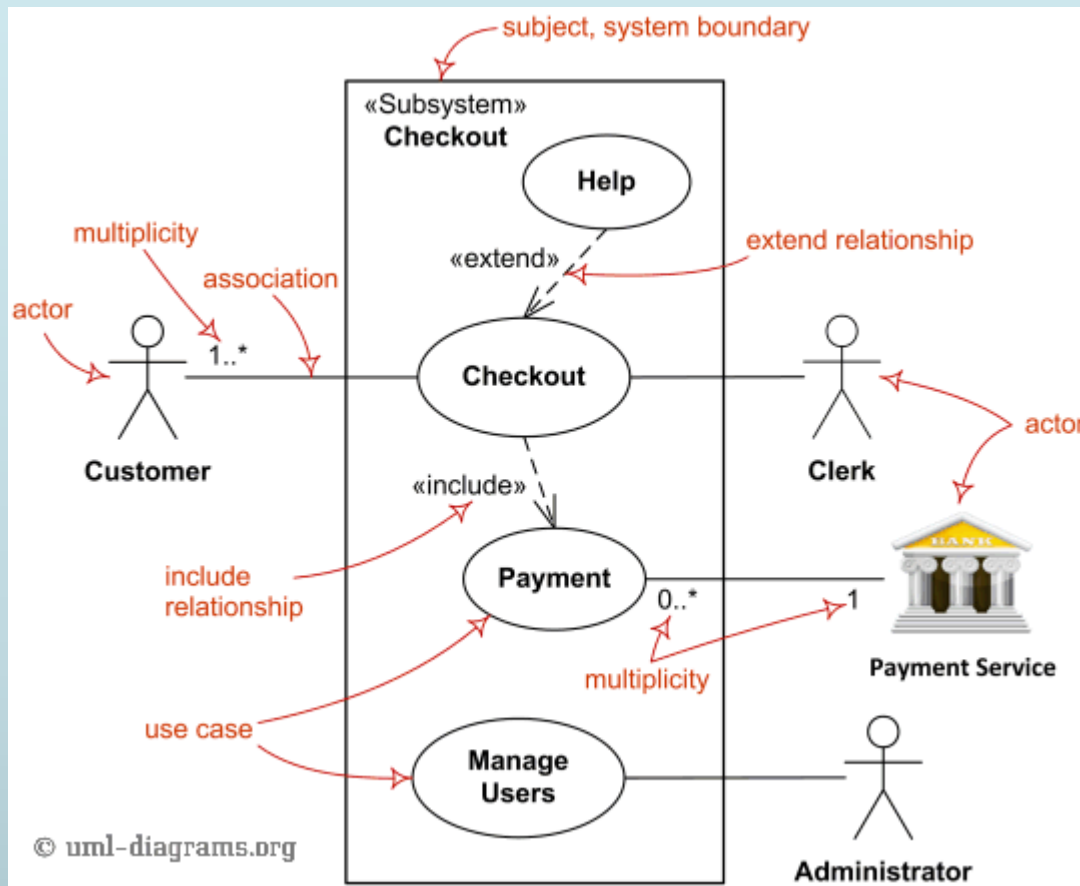
# Use-Case Diagram

---

- What is Use-Case Diagram?
  - Diagram for user or actor.
  - Use-Case는 System이 작동하는 방법과 System과 교류하는 사람들을 나타낸다. (Booch)
  - 때로는 업무 분담용으로 Use-Case가 사용됨.
- Use-Case 작성 원칙 ( WAVE )
  - W : Use-Case는 어떻게 할 것 인지가 아니라 무엇(What)을 할 것 인지 나타내고 있는가?
  - A : Use-Case는 행위자(Actor)의 관점에서 설명되는가?
  - V : Use-Case는 행위자의 가치(Value)를 포함하는가?
  - E : 이벤트들의 흐름은 전체(Entire) 시나리오인가?

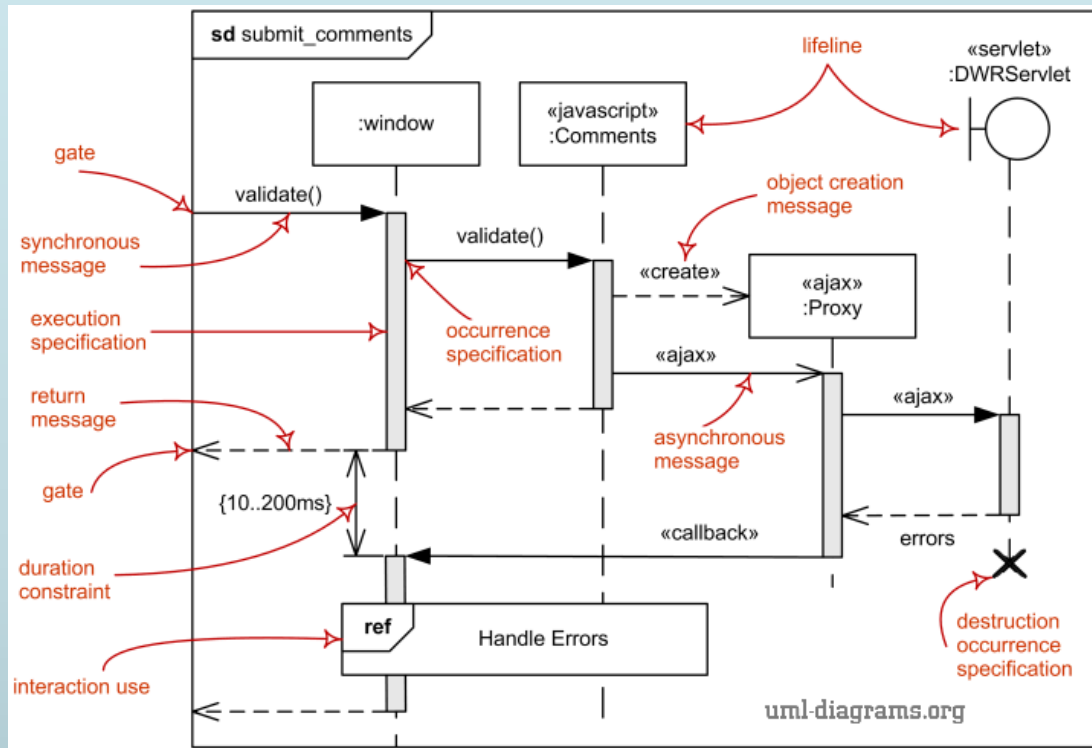
# Use-Case Diagram

- Example



# Sequence Diagram

- What is Sequence Diagram?
  - 객체간의 교류 + 시간.
  - 객체들 사이의 교류는 반드시 특정한 Sequence를 가짐.
- Example



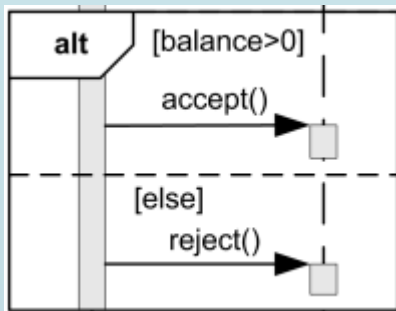
# Sequence Diagram

- Combined Fragment

- 참조, 반복, 조건, 병행처리 등을 표현

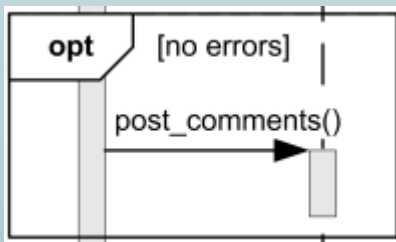
- Alt

- 조건 선택 실행
- 조건에 따른 선택이 다수일 경우



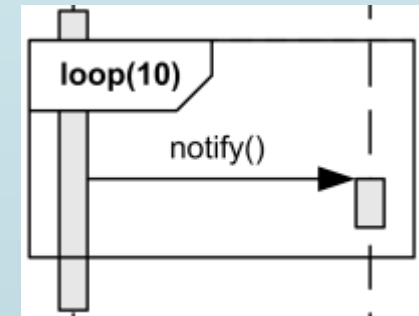
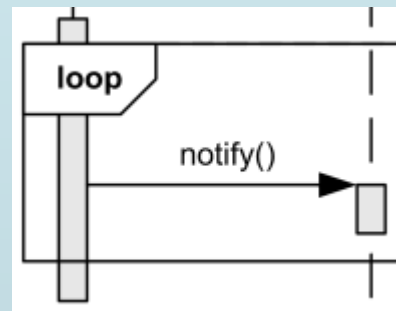
- Opt

- 조건 선택 실행
- 조건에 따른 선택이 하나일 경우



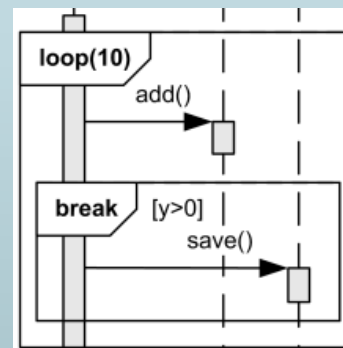
- Loop

- 반복 실행



- Break

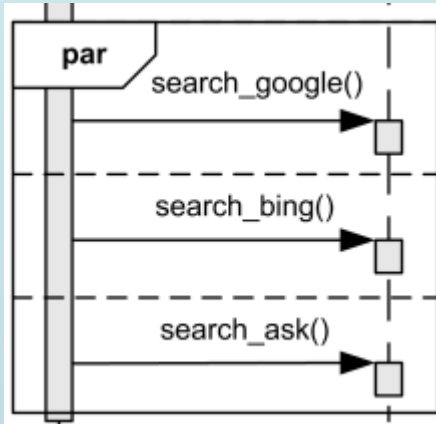
- 반복 중지



# Sequence Diagram

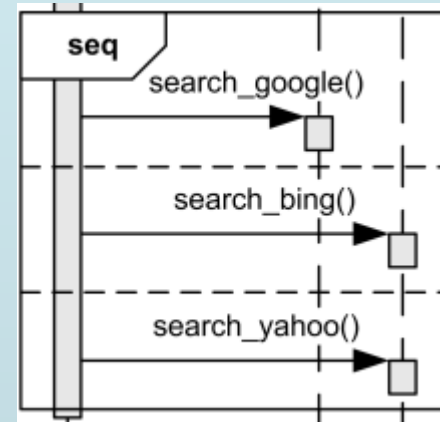
- Par

- 병행 실행



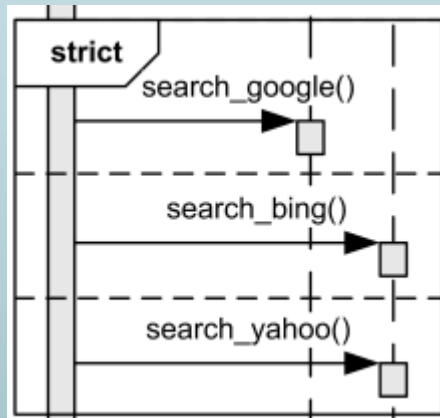
- Seq

- 실행 순서가 중요하지 않은 경우



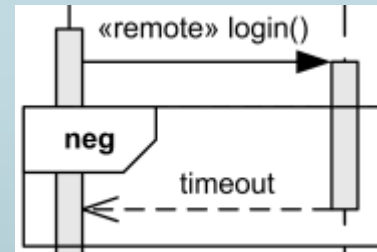
- Strict

- 실행 순서가 중요한 경우



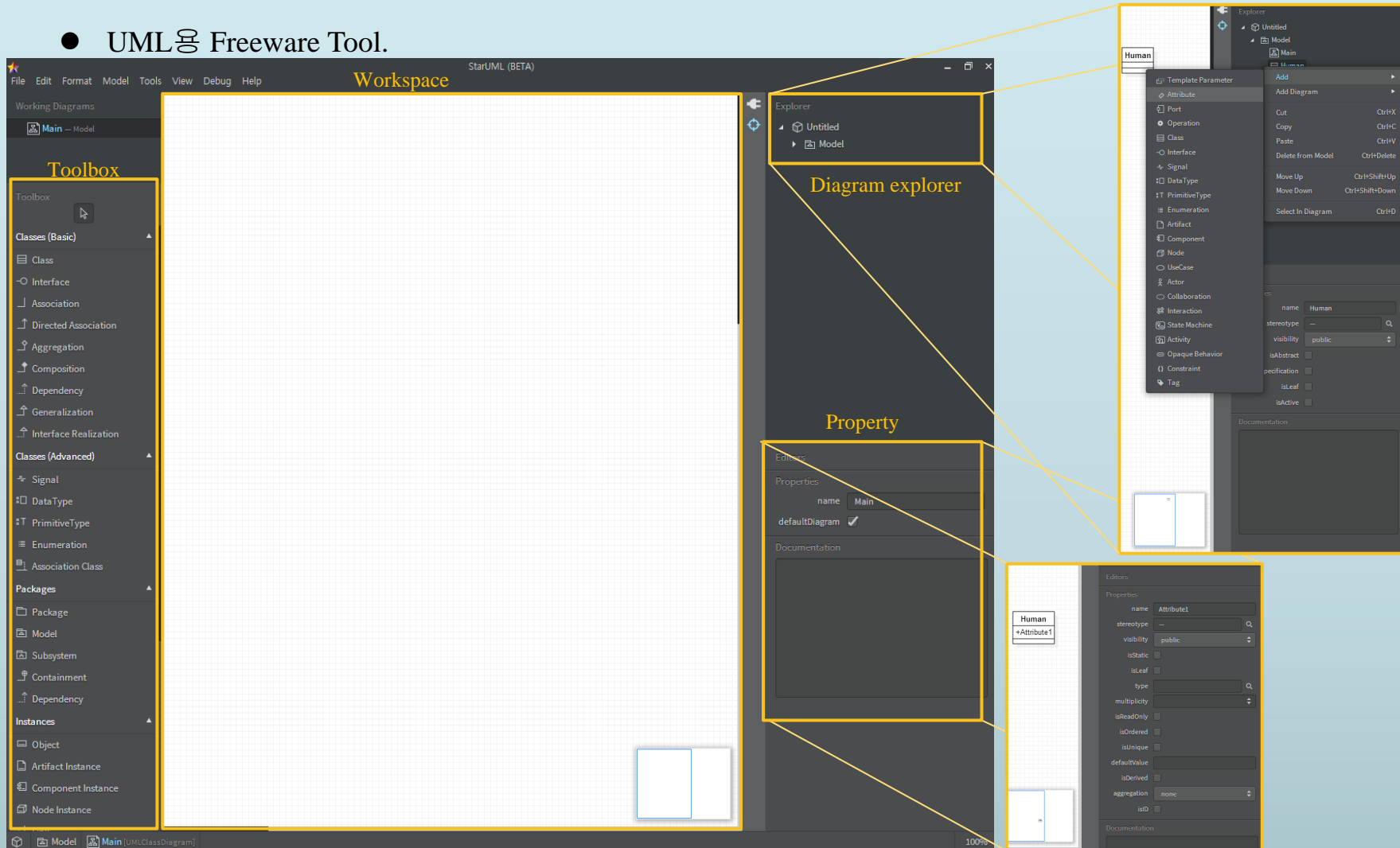
- Neg

- 절대로 발생하면 안되는 경우



# StarUML

## ● UML용 Freeware Tool.



# Conclusion

---

- 발표 자료에서 설명된 Diagram 이외에도 다양한 Diagram이 있으나, Sequence Diagram과 의미 하는 바가 겹치거나(State, Activity, Communication), 우리의 업무와 상관이 없는 Diagram(Component, Deployment)은 생략하였다.
- UML은
  - 사전에 제작된 알고리즘과 그에 해당하는 코드의 재활용성
  - 타 기관과의 협력 과제 수행 시 업무 분담 효율 및 작업 능력 향상
  - 업무 인수 인계 효율성 향상의 장점이 있다.



Q&A

