

UML

UML(Unified Modeling Language)

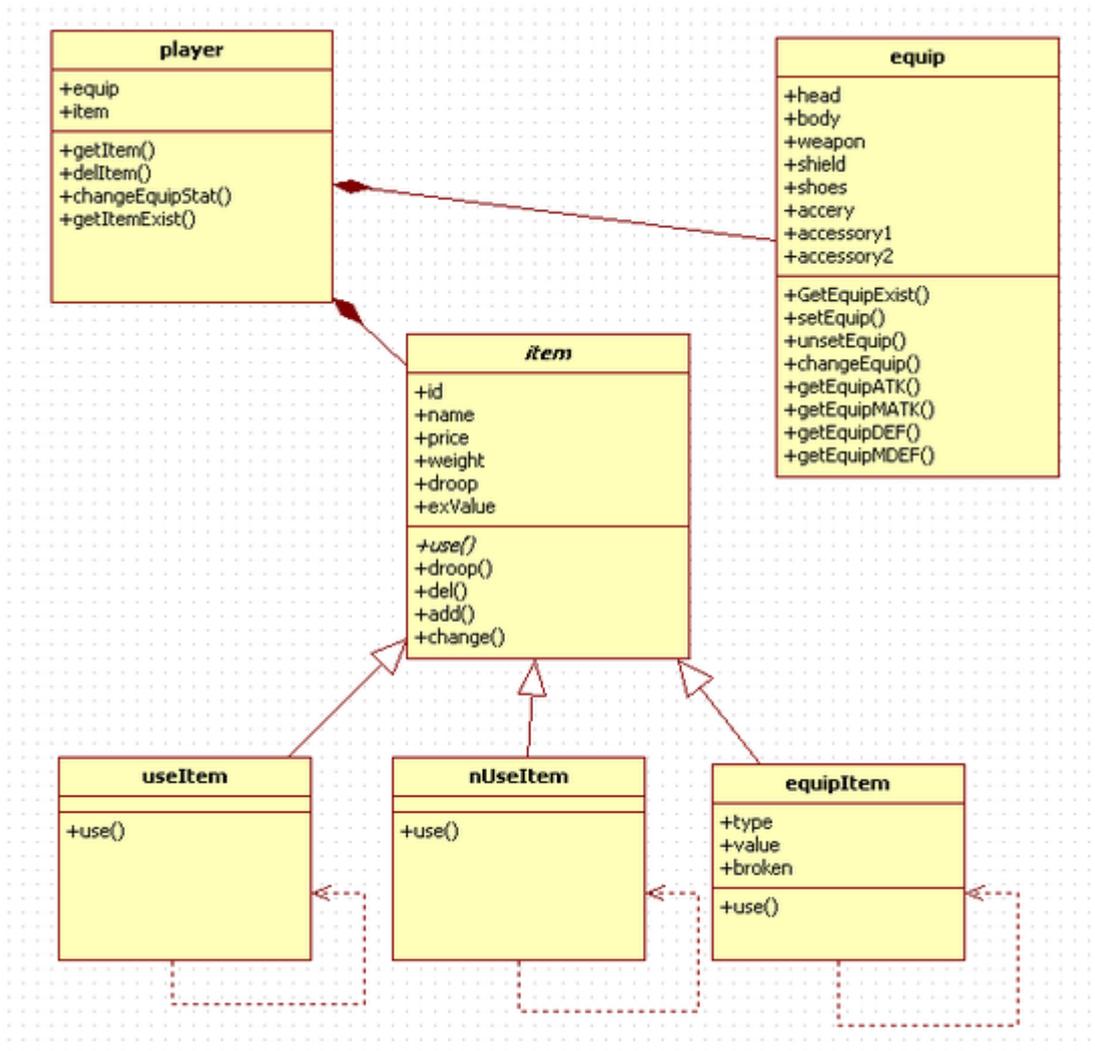
Sequence Diagram

가 가

Class Diagram

. UML

Class Diagram, Usecase Diagram,



UML

StarUML NS

가

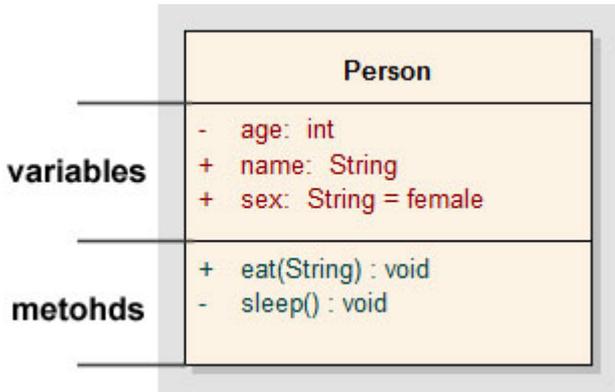
StarUML

가

- <http://sourceforge.net/projects/staruml/files/staruml/5.0/>

Class Diagram

UML 가



type 가 . (ActionScript 3.0 ,)
 variable method

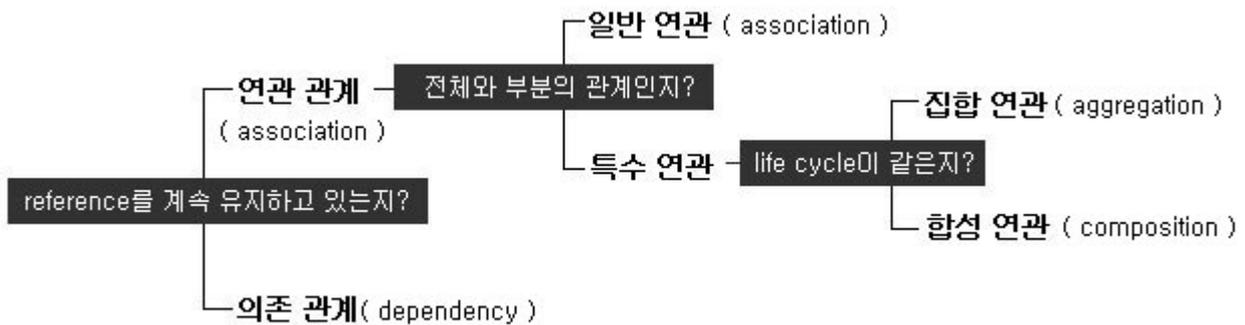
+	public
-	private
#	protected

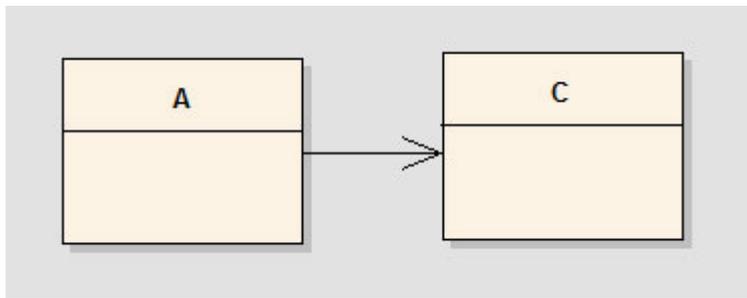
가

suppress

가

가





```

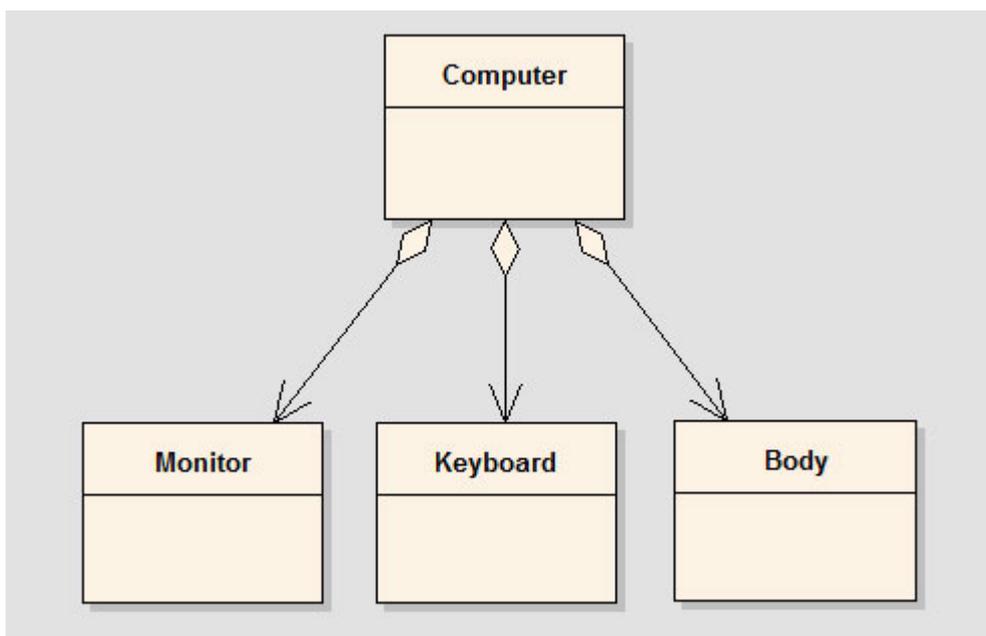
package classes
{
  public class A
  {
    private var c:C;

    public function A()
    {
      // 연관 : 클래스 C 에 대한 reference를 계속 유지하고 있음
      this.c = new C();
    }

    private function methodA():void
    {
      this.c.methodC();
    }
  }
}
  
```

aggregation, composition life cycle(,)

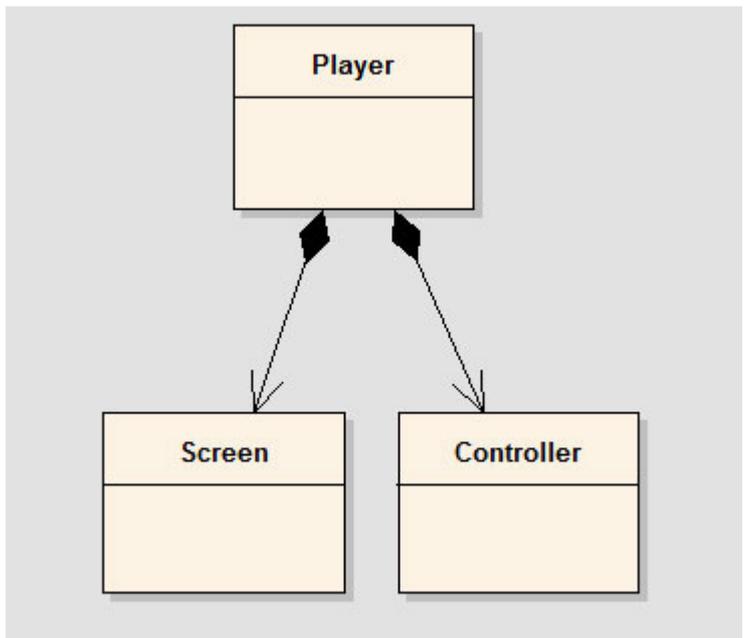
aggregation



```
package classes
{
    public class Computer
    {
        private var monitor:Monitor;
        private var body:Body;
        private var keyboard:Keyboard;

        public function Computer(_monitor:Monitor, _body:Body, _keyboard:Keyboard)
        {
            // 집합 연관 : 부분과 전체의 관계
            // 부분이 되는 객체를 외부에서 생성하며 넘겨받음
            // 따라서 computer 클래스가 없어도 부분이 되는 객체들은 사라지지 않음
            this.monitor = _monitor;
            this.body = _body;
            this.keyboard = _keyboard;
        }
    }
}
```

composition

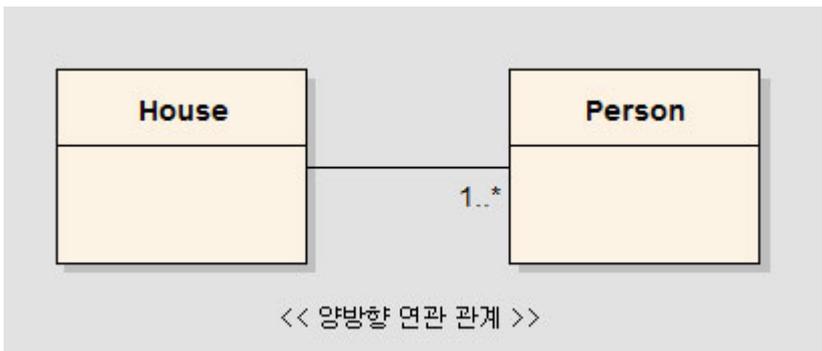


```

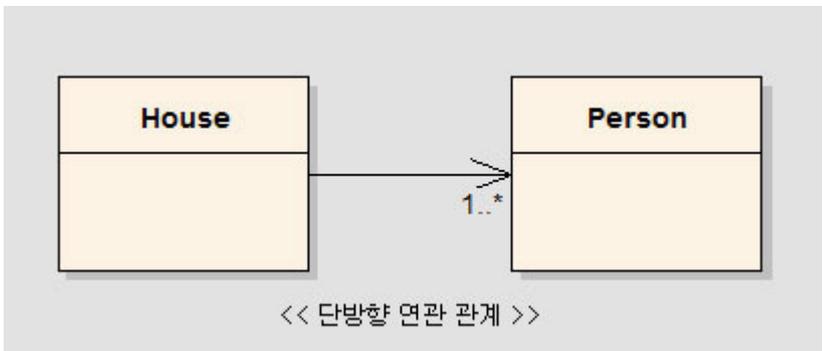
package classes
{
    public class Player
    {
        private var screen:Screen;
        private var controller:Controller;

        public function Player()
        {
            // 합성 : 집합 관계 중에서도 강한 집합체의 의미를 가짐
            // 부분을 이루는 객체가 없이는 전체가 아무 의미를 갖지 못함
            // Player 클래스가 사라져 버리면 내부에서 생성된 screen, controller도 같이 사라짐
            this.screen = new Screen();
            this.controller = new Controller();
        }
    }
}

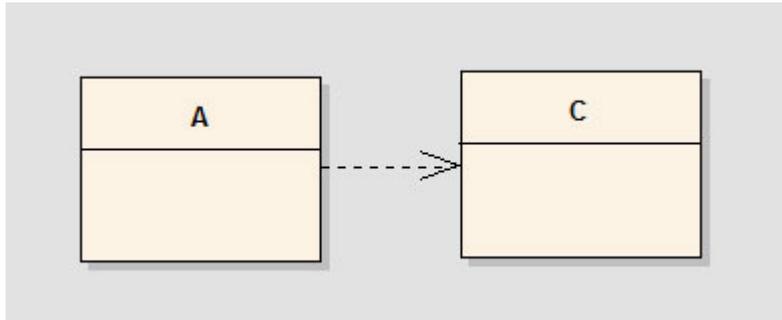
```



UML
House House Person



House Person , Person House
House . House Person 가 , Person
House 가 .



```
package classes
{
    public class A
    {
        private var c:C;

        public function A()
        {
        }

        private function methodA():void
        {
            // 의존 관계 : 이 메소드 내부에서만 c의 레퍼런스를 유지함
            // 함수 실행이 끝나면 c 의 참조도 사라짐
            var objC:C = new C();
            c.methodC();
        }
    }
}
```

StarUML 5.0 가
UML - ()
UML:

From:
<http://obg.co.kr/doku/> - **OBG Wiki**

Permanent link:
http://obg.co.kr/doku/doku.php?id=programming:design_pattern:uml

Last update: **2020/11/29 14:09**

