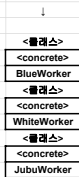
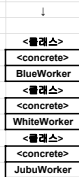




- > 이 클래스를 사용하는 다른 클래스는 변경할 필요가 없다.
- > 위임은 기능 변경과 교체 쉽다.



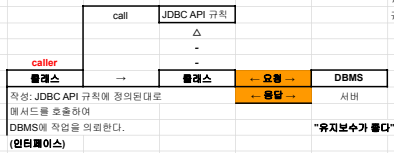
\*인터페이스와 코드 작성



\*인터페이스와 웹 프로그래밍



\*인터페이스와 JDBC 프로그래밍



규칙에 따라 호출될 클래스를 만들 것이지.  
규칙에 따라 호출한 클래스를 만들 것이지.

\*메서드 파라미터와 인터페이스

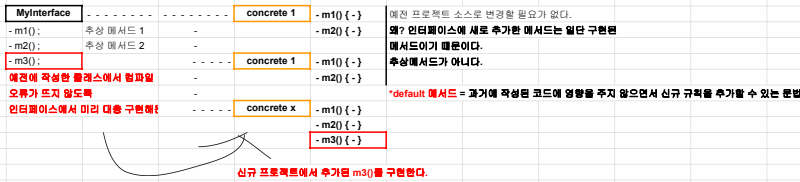
Exam.a1 -----> use(Tool use(Tool A) 객체  
- main() { ~ }  
use (toolB) { ~ } -----> Tool B 객체  
use (toolC) { ~ }



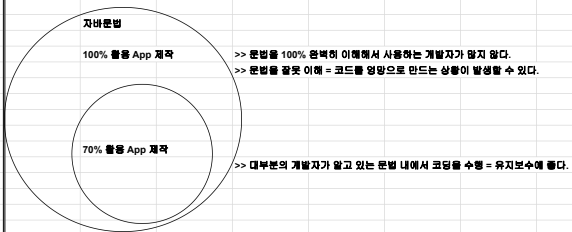
\*default 메서드 등장 전



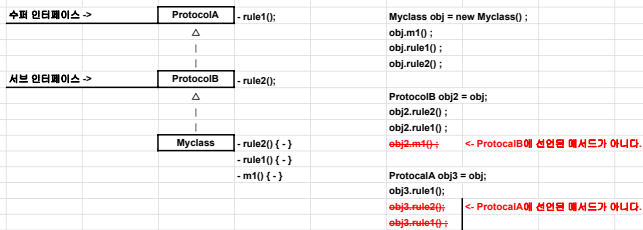
\*default 메서드 등장 후



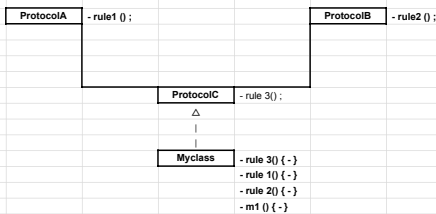
\*자바 문법의 활용



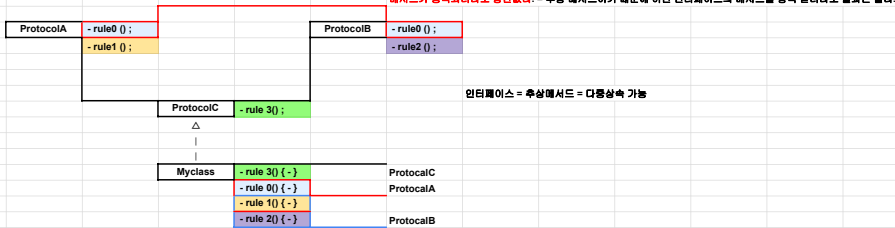
인터페이스의 상속



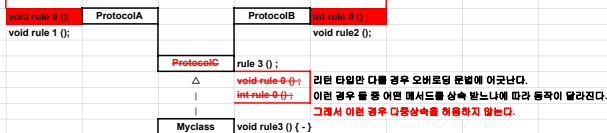
인터페이스의 상속과 구현 2



다중 상속과 메서드 중복



다중 상속 불가



void rule0 () { - }  
int rule0 () { - }  
void rule1 () { - }  
void rule2 () { - }

리턴타입이 다른 경우 "오버로딩"이 불가능기때문에 다중상속을 할 수 없음.

"다중 인터페이스 구현 불가"

ProtocolA

void rule0 ()  
rule1 ()

ProtocolB

int rule0 ()  
rule2 ()

Myclass

리턴 타입만 다른 경우  
"오버로딩" 불가  
> 컴파일 오류

void rule0 () { - }  
void rule1 () { - }  
int rule0 () { - }  
void rule2() { - }

ProtocolA

ProtocolB

상속

class A

↑

↑

class C

↑

class D

Interface

Interface A

△

class B

△

class B

△

class B



















