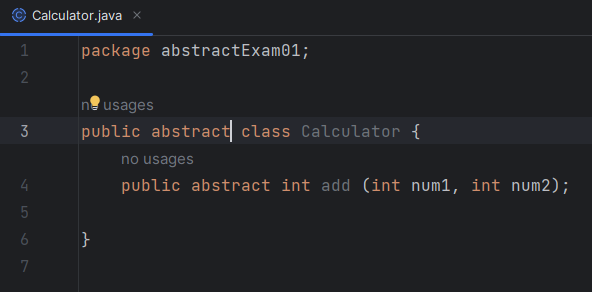
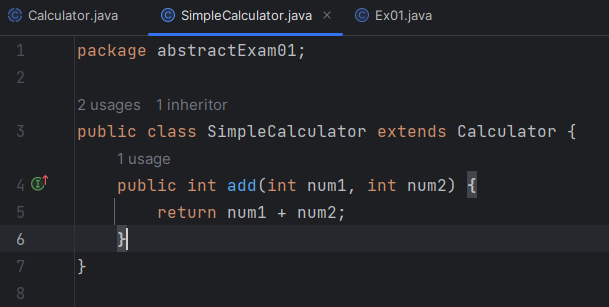
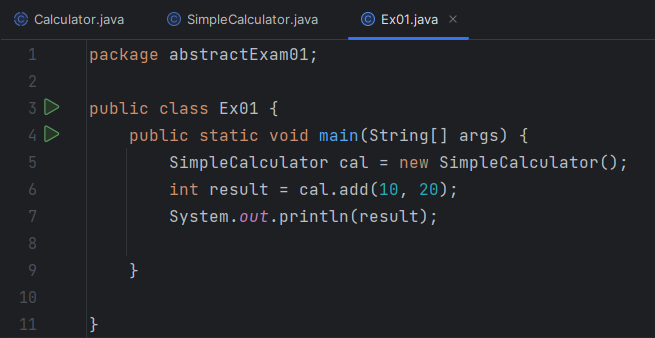
추상 클래스(abstract class)

하나 이상의 추상 메서드(메서드의 구현체가 구현되지 않은 것)를 포함하는 클래스. 상속받는 클래스에서 반드시 오버라이딩 해서 구현, 사용해야 한다(사용하는 이유).

* **추상 클래스 문법 : abstract class 클래스명( ) { };**
* **추상 메서드 문법 : abstract 반환형 메서드명( );**

먼저 상속을 통해 자식 클래스를 만들고, 만든 자식 클래스에서 추상 클래스의 모든 추상 메서드를 오버라이딩(Ctrl + O)하고 나야 자식 클래스의 인스턴스를 생성할 수 있게 된다. -> 상속을 통해 하위 클래스가 객체가 된다.

하위 클래스가 따라야 할 원칙, 규칙(설계 원칙, 설계의 틀)을 제공. 추상클래스는 객체가 되지 않으나 내부적으로 공간을 가짐(다형성). 하위 클래스가 제공하는 공통 기능, 변수를 공유할 때 쓰임.

final 예약어

변경이 불가하다. 제한하다. 최종적이다.

1. 변수 : 수정할 수 없기에 초기화 값이 필수다. 상수가 된다.
2. 메서드 : 오버라이드를 제한한다. 상속받은 클래스에서 final이 붙은 메서드를 수정해서 사용하지 못한다(재정의 할 수 없다.).
3. 클래스 : 상속 불가능한 클래스가 된다. 다른 클래스에서 상속해서 재정의 할 수 없다.

인터페이스(interface)

설계 목적으로 사용되는 추상 클래스의 일종. 추상 메서드만 정의할 수 있다. 하나의 클래스가 여러 인터페이스를 상속할 수 있다.

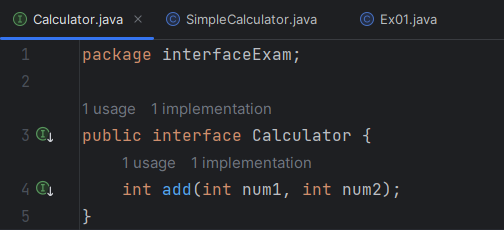
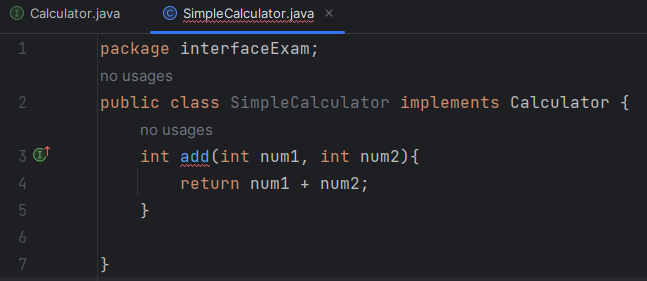
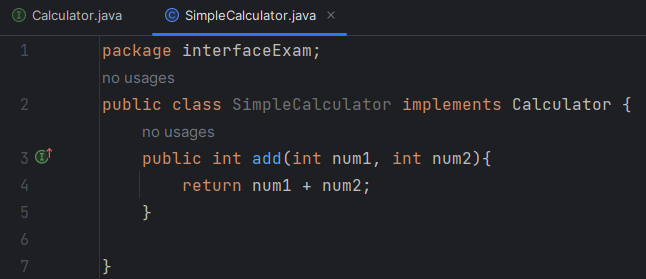
* **인터페이스 선언 : 접근제어자 interface 인터페이스명{**

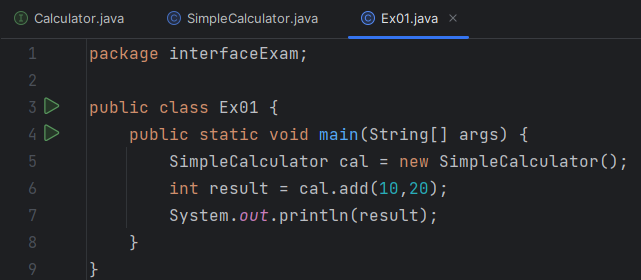
**Public static final 타입 상수명 = 값;**

**Public abstract 메서드명( );** // <- 추상클래스

**}**

* **인터페이스 구현 : class 하위클래스명 implements 인터페이스명{…}**

인터페이스의 모든 필드는 public static final이어야 하며, 모든 메서드는 public abstract이어야 한다. 이 부분은 모든 인터페이스에 공통으로 적용되는 부분이므로 이 제어자는 생략할 수 있다. 이렇게 생략된 제어자는 컴파일 시 자바 컴파일러가 자동으로 추가해 준다.

추상 클래스와 마찬가지로 내부적으로 객체인 것이다. 구현한 클래스 안의 객체인 것이다(다형성).