* Top Level Class
  ↳ 패키지에 소속된 클래스

com.eomcs.oop.ex11
↳ a
- sub
  - A
  - B
- Exam0110
- Exam0210
- Exam0310
- Exam0311

패키지에 직접 소속된 클래스
"Top Level Class"
(패키지 멤버 클래스)

* 접근 범위
  - public
  - (default) ⇒ package-private

* Nested 클래스

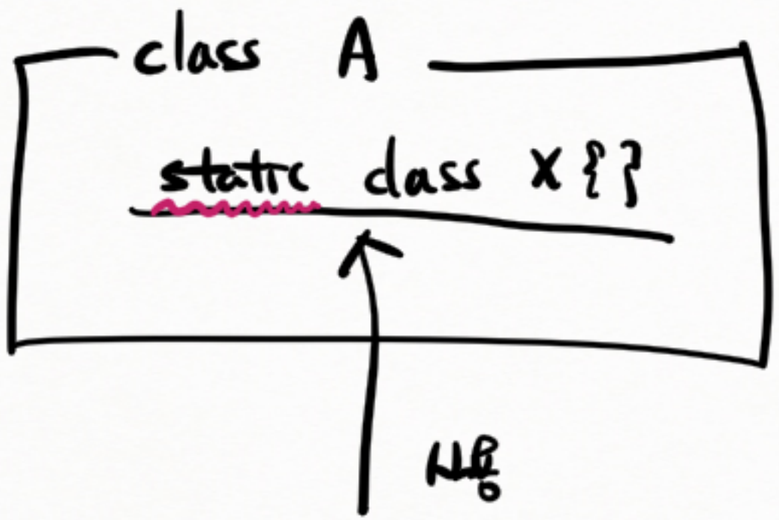
Exam0210

```
static class A{}   ← static nested class
class B{}          ← non-static nested class = inner class

main() {
    class C {}     ← local class
}
```

nested class (중첩 클래스)

* static nested class 내용



① 레퍼런스 선언 →

```
A.X obj;
```

A: 바깥클래스, X: 중첩클래스

② 객체 생성 (인스턴스)

```
obj = new A.X();
```

A: 바깥클래스, X: 중첩클래스

* nested 클래스에서 바깥클래스의 멤버 사용

[ex11.b.Exam02.10]

class B

static int v1;
static void m1(){-}

int v2;
void m2(){-}

static class X {}

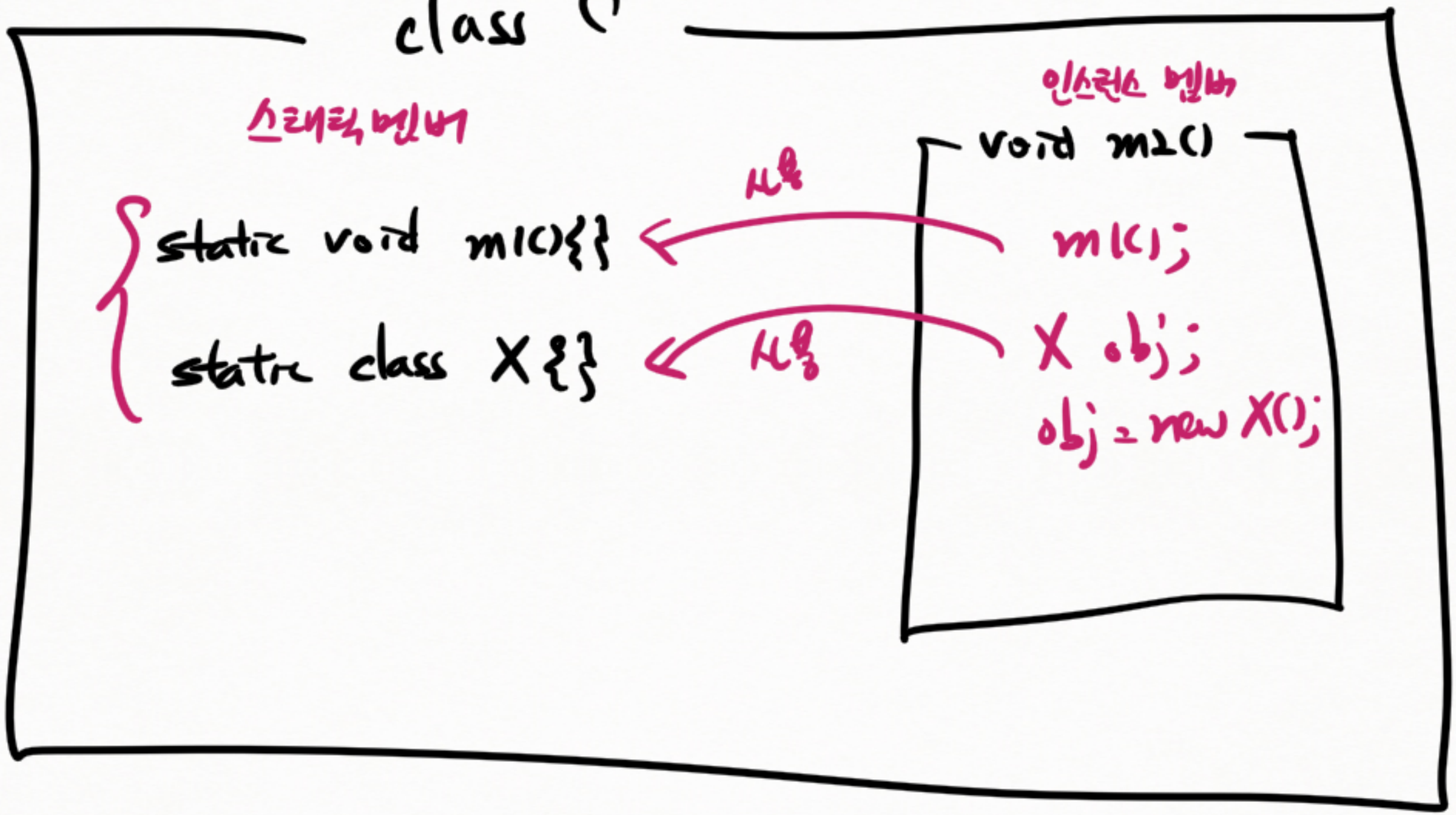
static class X

B.v1 = 100;
B.m1();
↑ 바깥클래스의 static 멤버 생략 가능

v2 = 100;
m2();

↑ 다른 스태틱 메서드 처럼
스태틱 클래스도 인스턴스 멤버 사용불가!
↳ B 객체를 가리켜야 사용 할 수 있다.

* 다른 멤버가 스태틱 nested 클래스 사용



* 스태틱 멤버 import 하기

[ex11.6.Exam0410]

class D

static int v1;

static void m1(){-]

static class X{}

다른 패키지의 class M

static int v2;

static vorit m2()

static class Y{}

class Exam0410

D.v1=100;

D.m1();

D.X obj=new D.X();

패키지를 import 한 후

M.v2=100;

M.m2();

M.Y obj=new M.Y();

* 스태틱 멤버 import 하기 : import static

import static com.eomcs.oop.ex11.b.E.v1;

class E
static int v1;
static void m1(){-}
static class X{}

다른 패키지의 class M
static int v2;
static void m2()
static class Y{}

class Exam0420
← 클래스명 생략가능
~~E.~~ v1 = 100;
~~E.~~ m1();
~~E.~~ X obj = new E.X();

import static 사용후
~~M.~~ v2 = 100;
~~M.~~ m2();
~~M.~~ Y obj = new M.Y();

클래스명 생략 가능