이너클래스, 이너인터페이스

이너클래스 (inner class)

이너클래스(inner class)

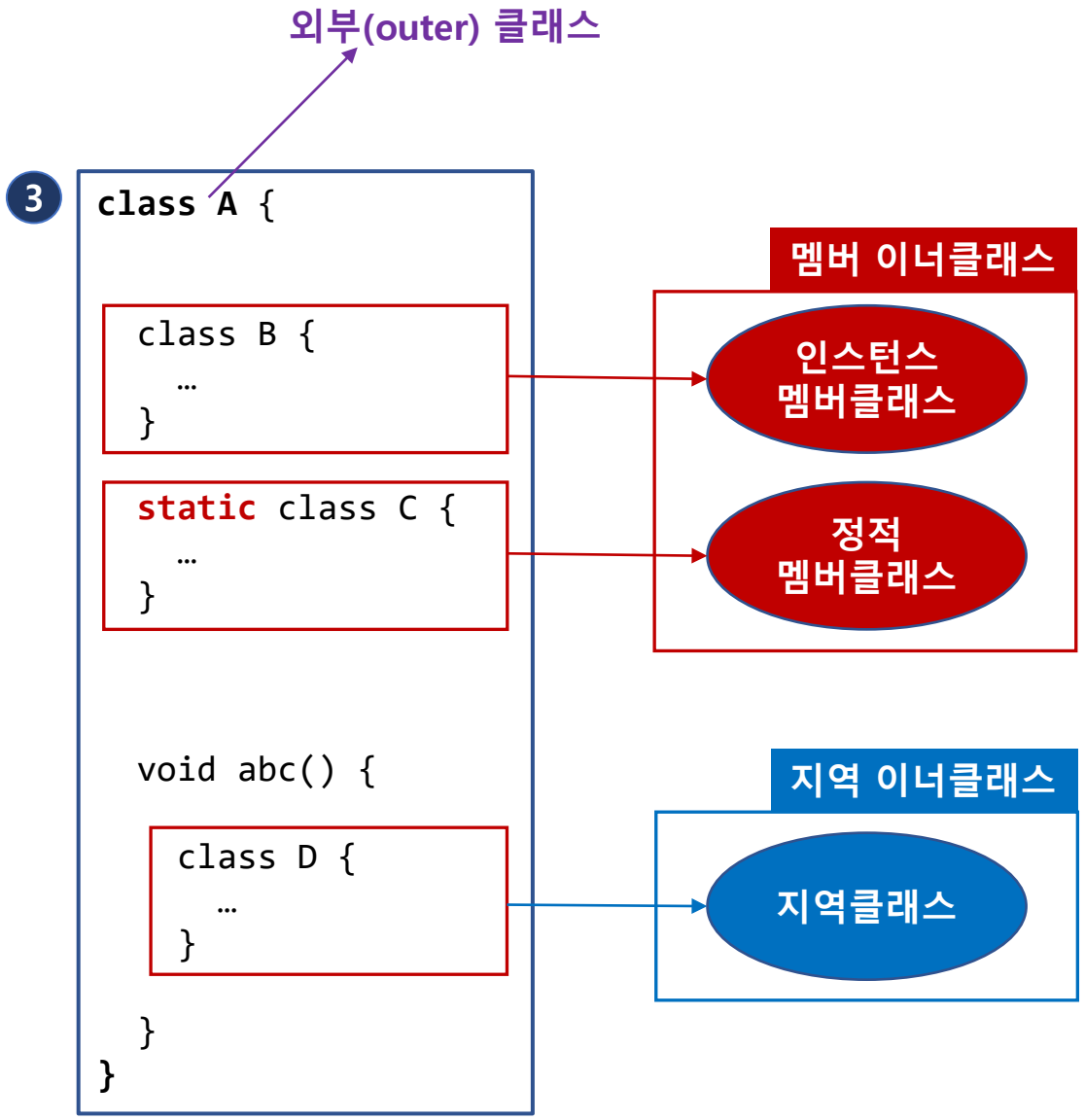
☞ 이너클래스

1 - 정의: 클래스 내부에 포함된 클래스

- 종류

2 **멤버클래스**
#1. 인스턴스 이너 클래스
#2. 정적 이너 클래스

#3. 지역클래스



이너클래스(inner class)

☞ **#1**. 인스턴스 멤버 이너클래스 - **객체생성 및 멤버 사용**

1 - **특징**: 외부(outter)클래스의 모든 접근지정자의 멤버 접근 가능

3 - **생성클래스명**: **A.class, A$B.class**

```
외부클래스.class
외부클래스$이너클래스.class
```

- **객체생성방법**

4 **Step#1.** 외부클래스 객체 생성
Step#2. 이너클래스 객체 생성

```
외부클래스 a = new 외부클래스();
외부클래스.이너클래스 b = 외부클래스객체.new 이너클래스();
```

5

```
A a = new A();
A.B b = a.new B();

b.bcd();
```

```
3
4
5
6
```

외부(outer) 클래스

2

```
class A {

  public int a = 3;
  protected int b = 4;
  int c = 5;
  private int d = 6;

  class B {

    void bcd(){
      System.out.println(a);
      System.out.println(b);
      System.out.println(c);
      System.out.println(d);
    }

  }

}
```

이너클래스(inner class)

☞ **#1**. 인스턴스 멤버 이너클래스 - **객체생성 및 멤버 사용**

1

```
class A {
  public int a = 3;
  protected int b = 4;
  int c = 5;
  private int d = 6;

  void abc() {
    System.out.println("A 클래스  메서드");
  }

  class B {
    void bcd() {
      //#1. outer class 필드 사용
      System.out.println(a);
      System.out.println(b);
      System.out.println(c);
      System.out.println(d);
      //#2. outer class 메서드 사용
      abc();
    }
  }
}
```

```
외부클래스 a = new 외부클래스();
외부클래스.내부클래스 b = 외부클래스객체.new 내부클래스();
```

A 클래스 외부에서 객체 생성 및 사용시

2

```
public static void main(String[] ar) {

  //#3. instance inner 클래스 객체 생성
  //@3.1. outer class 객체 생성
  A a = new A();

  //@3.2 outer 참조변수로 부터 inner클래스 객체 생성
  A.B b = a.new B();
  b.bcd();

}
```

```
3
4
5
6
A 클래스  메서드
```

이너클래스(inner class)

☞ **#1**. 인스턴스 멤버 이너클래스 - **외부 클래스의 객체 참조**

1 - **외부 클래스의 객체 참조**

외부클래스명.this

3

```
A a = new A();
A.B b = a.new B();

b.bcd();
```

```
5
6
3
4
```

4

TIP

- 인스턴스 멤버이너클래스는 힙메모리내의 외부클래스 객체내에 생성되기 때문에 외부 클래스 객체를 먼저 생성 하여야 함

2

```
class A {

  int a = 3;
  int b = 4;

  class B {
    int a = 5;
    int b = 6;

    void bcd(){
      System.out.println(this.a);//5
      System.out.println(this.b);//6

      System.out.println(A.this.a);//3
      System.out.println(A.this.b);//4
    }

  }

}
```

이너클래스(inner class)

☞ **#1**. 인스턴스 멤버 이너클래스 - **외부 클래스의 객체 참조**

```
class A {
  int a = 3, b = 4;
  void abc() {
    System.out.println("A 클래스 메서드");
  }
  class B {
    int a = 5, b = 6;
    void abc() {
      System.out.println("B 클래스 메서드");
    }
    void bcd() {
      // #1. outer class 필드/메서드 사용
      System.out.println(a);
      System.out.println(b);
      abc();
      // #2. outer class 필드/메서드 사용
      System.out.println(A.this.a);
      System.out.println(A.this.b);
      A.this.abc();
    }
  }
}
```

1

3

4

```
외부클래스 a = new 외부클래스();
외부클래스.내부클래스 b = 외부클래스객체.new 내부클래스();
```

A 클래스 외부에서 객체 생성 및 사용시

2

```
public static void main(String[] ar) {

  //#3. instance inner 클래스 객체 생성
  //@3.1. outer class 객체 생성
  A a = new A();

  //@3.2 outer 참조변수로부터 inner클래스 객체 생성
  A.B b = a.new B();
  b.bcd();

}
```

```
5
6
B 클래스 메서드
3
4
A 클래스 메서드
```

이너클래스(inner class)

☞ **#2**. 정적 멤버 이너클래스

1 - **특징**: 외부(outter)클래스의 static 멤버만 접근가능 (static의 특징)

3 - **생성클래스명**: **A.class, A$B.class**

외부클래스.class
외부클래스$이너클래스.class

- **객체생성방법**

4 **Step#1.** 내부클래스 생성자로 직접 객체 생성

```
외부클래스.이너클래스 a = new 외부클래스.이너클래스();
```

```
A.B b = new A.B();

b.bcd();
```

정적멤버이너클래스

외부(outter) 클래스

2

```
class A {

  int a = 3;
  static int b = 4;

  static class B {

    void bcd(){
      System.out.println(a); (X)
      System.out.println(b); (O)
      System.out.println("정적멤버이너클래스");
    }

  }

}
```

이너클래스(inner class)

☞ **#2**. 정적 멤버 이너클래스

```
class A{
  int a=3;
  static int b=4;
  void method1() {
    System.out.println("Instance Method");
  }
  static void method2() {
    System.out.println("Static Method");
  }

  static class B{
    void bcd() {
      //#1. outer 클래스 필드 접근
      //System.out.println(a); //(불가능)
      System.out.println(b);

      //#2. outer 클래스 메서드 접근
      //method1(); //(불가능)
      method2();
    }
  }
}
```

외부클래스.내부클래스 a = new 외부클래스.내부클래스();

A 클래스 외부에서 객체 생성 및 사용시

```
public static void main(String[] ar) {

  //#3. static inner class 객체생성
  A.B b = new A.B();
  b.bcd();

}
```

```
4
Static Method
```

이너클래스(inner class)

☞ #3. 지역 이너클래스

1 - **특징**:

특징1. 메서드 내부에서 정의된 클래스
특징2. 외부(outter)클래스의 필드는 모두 접근 가능
특징3. 메서드 지역변수는 final만 사용가능
(final로 지정하지 않은 경우 컴파일러가 자동으로 지정)

3 - **생성클래스명**: **A.class, A$1B.class**

동일한 클래스 명이 있을 때 번호증가

메서드가 종료되면 모든 지역변수는 사라지기 때문

```
외부클래스.class
외부클래스$+번호내부클래스.class
```

- **객체생성 및 사용**

4 메서드 내에서만 생성 및 활용 가능

```
A a = new A();
a.abc();
```

→

```
3
5
```

지역 클래스에서 사용되는 경우 자동으로 (final) 추가

2

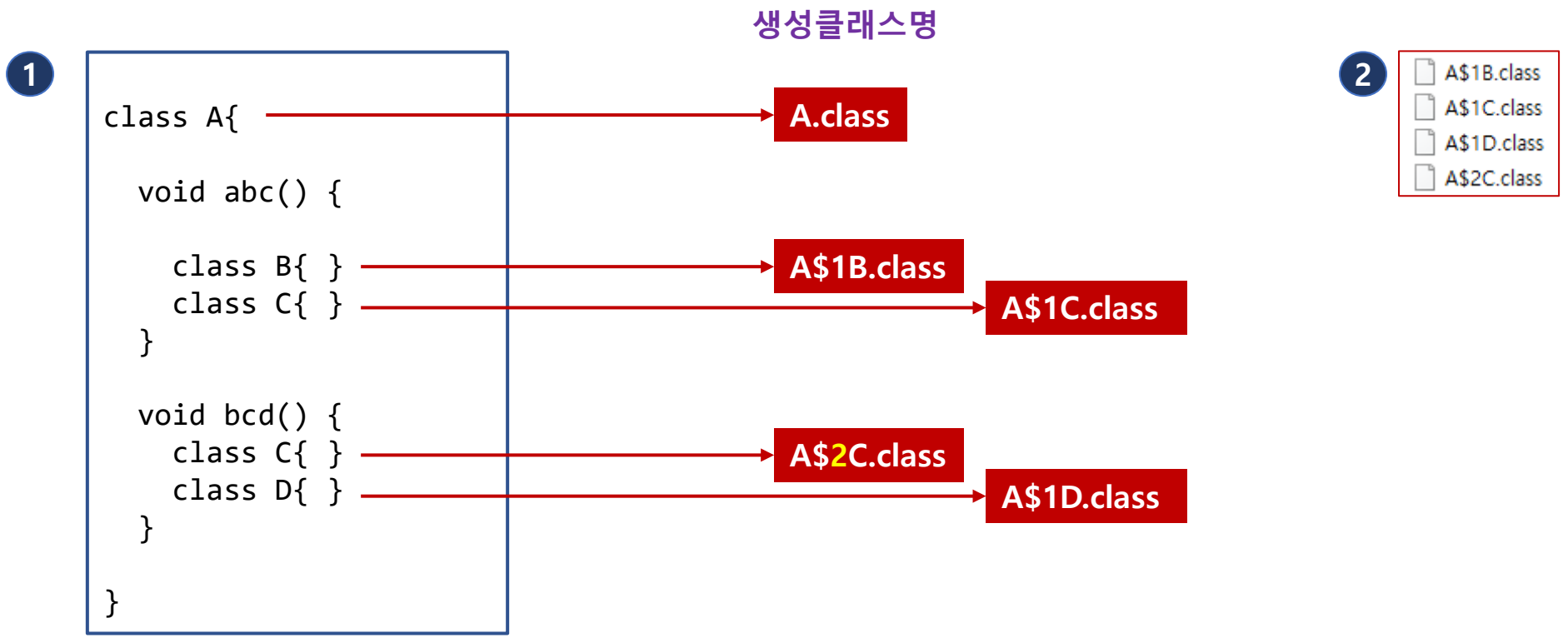
```
class A {

  int a = 3; //필드

  void abc(){
    int b = 5; //지역변수

    class B {
       void bcd(){
         System.out.println(a); //필드 (O)
         System.out.println(b); //지역변수 (O)
         a=7; //필드값 변경 (O)
         //b=7; //지역변수값 변경 (X)
      }
    }

    B bb = new B();
    bb.bcd();
  }
}
```

5 TIP

- 지역 이너클래스는 static 지정 불가능

이너클래스(inner class)

☞ #3. 지역 이너클래스



The End

익명이너클래스(Anonymous class)

익명이너클래스(Anonymous class)

1 ☞ 익명 이너클래스 ← 익명(이름을 알 수 없음) + 이너클래스

2
```
interface C {
  public abstract void bcd();
}
```

3
```
class A {

  C b = new B();

  void abc(){
    b.bcd();
  }

  class B implements C{

    public void bcd(){
      System.out.println(…);
    }

  }
}
```

필드 → `C b = new B();`

메서드 → `void abc(){ b.bcd(); }`

이너클래스 → `class B implements C{ … }`

4
```
class A {

  C b = new C(){
    public void bcd(){
      System.out.println(…);
    }
  };

  void abc(){
    b.bcd();
  }

}
```

C를 상속받아 bcd()메서드를 오버라이딩한 익명 이너클래스의 객체

익명이너클래스(Anonymous class)

TIP

- 익명이너클래스를 사용하는 경우 이름이 없어 한번에 객체를 2개 이상 생성 불가능

☞ 익명 이너클래스 ← **익명(이름을 알 수 없음) + 인스턴스이너클래스**

1
```
interface C {
  public abstract void bcd();
}
```

2
```
class B implements C{

  public void bcd(){
    System.out.println(…);
  }

  public void cde(){
    …
  }
}
```
오버라이딩 메서드 → `public void bcd(){ … }`

추가 메서드 → `public void cde(){ … }`

3
```
B b = new B();
b.bcd(); (O)
b.cde(); (O)
```

4
```
C c = new C(){

  public void bcd(){
    System.out.println(…);
    cde();
  }
  public void cde(){
    …
  }
};
```
오버라이딩 메서드 → `public void bcd(){ … }`

추가 메서드 → `public void cde(){ … }`

`cde();` → 내부적으로 호출가능 (6)

5
```
c.bcd(); (O)
c.cde(); (X) // C 타입에 없기때문
```

7

익명이너클래스(Anonymous class)

4 **방법 1**. 클래스명 O + 참조변수명 O
방법 2. 클래스명 O + 참조변수명 X

☞ 익명이너클래스를 활용한 인터페이스 타입의 매개변수 전달

-방법#1

1
```
class C {

  void cde(A a){
    a.abc();
  }

}
```

2
```
interface A {
  public abstract void abc();
}
```

```
class B implements A {
  public void abc(){
    …
  }
}
```

3
```
C c = new C();

//방법1.
A a1 = new B();
c.cde(a1);

//방법2.
c.cde(new B());
```

익명이너클래스(Anonymous class)

☞ 익명이너클래스를 활용한 인터페이스 타입의 매개변수 전달

-방법#2

4

방법 3. 클래스명 X + 참조변수명 O
방법 4. 클래스명 X + 참조변수명 X

1

```
class C {

  void cde(A a){
    a.abc();
  }

}
```

2

```
interface A {
  public abstract void abc();
}
```

3

```
C c = new C();
//방법#3
A a = new A(){
 public void abc(){ …
 }
};
c.cde(a);
```

```
//방법#4
c.cde(new A(){
  public void abc(){
    …
  }
});
```

이너인터페이스(inner interface)

이너인터페이스(inner interface)

TIP

- 이너인터페이스는 정적(static)이너인터페이스만 가능 (static 생략시 자동 추가)

☞ 내부 인터페이스

- **특징**:

1. **특징1**. 외부 클래스와 밀접한 관계가 있는 경우에 정의
   특징2. UI의 이벤트 처리에 가장 많이 사용 (클릭, 터치 등)
   특징3. static을 생략한 경우 컴파일러는 자동으로 static 삽입

3. - **생성클래스명**: **A.class, A$B.class**

- **객체생성 및 사용**

4. **방법#1.**
인터페이스 구현 클래스 생성 및 객체 생성

```
class C implements A.B{
  public void bcd(){
    …
  }
}

C c = new C();
c.bcd();
```

2.
```
class A {

  …

  static interface B {
    void bcd();
  }

}
```

static을 생략해도 컴파일러가 자동으로 추가

5. **방법#2.**
익명이너클래스 사용

```
A.B a = new A.B(){
  public void bcd(){
    …
  }
};

a.bcd();
```

static 클래스와 유사

이너인터페이스(inner interface)

☞ 내부 인터페이스

1
```
class A {
  interface B {
    public abstract void bcd();
  }
}
```

2
```
//방법 1-1. 이너인터페이스 구현 클래스 생성
class C implements A.B {
  public void bcd() {
    System.out.println("이너인터페이스 구현 클래스 생성");
  };
}
```

3
```
public static void main(String[] ar) {
  //객체 생성 방법 1-2. 구현 클래스로 객체 생성
  C c = new C();
  c.bcd();
```

4
```
  //객체 생성 방법 2. 익명이너클래스로 객체 생성
  A.B b = new A.B() {
    @Override
    public void bcd() {
      System.out.println("익명이너클래스로 객체 생성");
    }
  };
  b.bcd();
}
```

이너인터페이스 구현 클래스 생성
익명이너클래스로 객체 생성

이너인터페이스(inner interface)

☞ 내부 인터페이스

- 대표적 예시

1 API 제공

```
class Button{
  OnClickListener ocl;
  void setOnClickListener(OnClickListener ocl) {
    this.ocl = ocl;
  }
  interface OnClickListener{
    public abstract void onClick();
  }
  void click(){
    ocl.onClick();
  }
}
```

2 개발자 #1 : 클릭했을때 음악 재생

```
public static void main(String[] ar) {
  Button button1 = new Button();
  button1.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick() {
      System.out.println("개발자1. 음악 재생");
    }
  });
  button1.click();
}
```

3 개발자 #2 : 클릭했을때 네이버 접속

```
public static void main(String[] ar) {
  Button button2 = new Button();
  button2.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick() {
      System.out.println("개발자2. 네이버 접속");
    }
  });
  button2.click();
}
```

The End