

추상 클래스와 인터페이스

추상 클래스(abstract class)

- 하나 이상의 추상 메소드를 포함하는 클래스
- 추상 메소드는 자식 클래스에서 추상 클래스를 상속받아서 오버라이딩해야만 사용할 수 있는 메소드
- abstract는 abstract 클래스나 메소드를 사용하기 위해서는 반드시 상속해서 사용하도록 강제하는 것

추상 클래스(abstract class)

```
1 abstract class A{
2     public abstract int b();
3 }
4
5 public class App {
6     public static void main(String[] args
7 ){
8     A obj = new A();
9 }
10
```

```
✓ J App.java src (3)
  ✗ A is abstract; cannot be instantiated (errors(1): 7:21-7:22) [Ln 7, Col 21]
  ✗ Cannot instantiate the type A Java(16777373) [Ln 7, Col 21]
  ⚠ Variable obj is never read (hints(1): 7:11-7:14) [Ln 7, Col 11]
```

추상 클래스는 구체적인 메소드의 내용이 존재하지 않기 때문에 인스턴스화시켜서 사용할 수 없다.

추상 클래스(abstract class)

```
1 abstract class A{
2     public abstract int b();
3 }
4
5 class B extends A{
6     public int b() {System.out.println(
7         "Hello World!"); return 1;}
8 }
9 public class App {
10     public static void main(String[] args
11     ){
12         B obj = new B();
13         System.out.println(obj.b());
14     }
15 }
```

```
● PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> c:\
21.0.6.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExcep
Hello World!
1
○ PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> █
```

추상 클래스(abstract class)

- 추상 클래스는 상속을 강제하기 위한 것이다.
- 상황에 따라서 동작 방법이 달라지는 메소드는 추상 메소드로 만들어서 하위 클래스에서 구현하도록 하고 모든 클래스의 공통분모의 경우에는 상위 클래스에 두어서 코드의 중복, 유지보수의 편의성 등의 효과를 얻을 수 있다.

인터페이스(interface)

- 어떤 객체가 있고 그 객체가 특정한 인터페이스를 사용한다면 그 객체는 반드시 인터페이스의 메소드들을 구현해야 한다.
- 인터페이스는 인터페이스를 구현할 클래스가 접근해야 하기 때문에 인터페이스의 멤버는 반드시 public이다.
- 인터페이스도 상속이 가능하다.

인터페이스(interface)

```
1 interface ex1{
2     public void a();
3 }
4
5 class A implements ex1{
6     public void b() {System.out.println(
7         "B");}
8 }
9 public class App {
10     public static void main(String[] args
11     ){
12         A obj = new A();
13         obj.b();
14     }
15 }
```

```
App.java src (2)
  ✖ The type A must implement the inherited abstract method ex1.a() Java(67109264) [Ln 5, Col 7]
  ✖ A is not abstract and does not override abstract method a() in ex1 (errors(1): 5:7-5:8) [Ln 5, Col 7]
```

인터페이스 내부의 메소드들은 abstract를 붙이지 않아도 전부 추상 메소드이다.

인터페이스(interface)

```
1 interface ex1{
2     public void a();
3 }
4
5 class A implements ex1{
6     public void a() {System.out.println(
7         "A");}
8 }
9 public class App {
10     public static void main(String[] args
11     ){
12         A obj = new A();
13         obj.a();
14     }
15 }
```

```
● PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> c:; cd 'c:\Us
21.0.6.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessag
A
○ PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> |
```


인터페이스(interface)

- 협업을 하는 경우 클래스가 가지고 있어야 할 메소드를 인터페이스로 만들어서 제공하면 직접적인 소통이 없어도 구현이 가능하다.
- 자바에서 하나의 클래스가 여러 개의 클래스를 상속받는 것은 불가능하다. 하지만 인터페이스는 여러 개를 받을 수 있으므로 다중 상속의 역할이 가능하다.

인터페이스(interface)

```
1 abstract class ex1{
2     public abstract void a();
3 }
4
5 abstract class ex2{
6     public abstract void b();
7 }
8
9 class A extends ex1 extends ex2{
10     public void a() {System.out.println("A");}
11     public void b() {System.out.println("B");}
12 }
13
14 public class App {
15     public static void main(String[] args){
16         A obj = new A();
17         obj.a();
18     }
19 }
20
```

J App.java src 4

- ❌ '{' expected (errors(1): 9:1-9:33) [Ln 9, Col 1]
- ❌ Syntax error on token "extends", permits expected Java(1610612940) [Ln 9, Col 21]
- ⚠️ ex2 is never used (hints(1): 5:16-5:19) [Ln 5, Col 16]
- ⚠️ Add @Override Annotation (hints(2): 10:17-10:18) [Ln 10, Col 17]

```
1 abstract class ex1{
2     public abstract void a();
3 }
4
5 abstract class ex2{
6     public abstract void b();
7 }
8
9 class A extends ex1, ex2{
10     public void a() {System.out.println("A");}
11     public void b() {System.out.println("B");}
12 }
13
14 public class App {
15     public static void main(String[] args){
16         A obj = new A();
17         obj.a();
18     }
19 }
20
```

J App.java src 6

- ❌ '{' expected (errors(1): 9:1-9:26) [Ln 9, Col 1]
- ❌ Syntax error, insert "ClassBody" to complete ClassMemberDeclaration Java(1610612976) [Ln 9, Col 17]
- ❌ Syntax error, insert ";" to complete Block Java(1610612976) [Ln 9, Col 25]
- ❌ Syntax error on token "}", delete this token Java(1610612968) [Ln 12, Col 1]
- ⚠️ ex2 is never used (hints(1): 5:16-5:19) [Ln 5, Col 16]
- ⚠️ Add @Override Annotation (hints(2): 10:17-10:18) [Ln 10, Col 17]

인터페이스(interface)

```
1 abstract class ex1{
2     public abstract void a();
3 }
4
5 interface ex2{
6     public abstract void b();
7 }
8
9 class A extends ex1 implements ex2{
10     public void a() {System.out.println("A");}
11     public void b() {System.out.println("B");}
12 }
13
14 public class App {
15     public static void main(String[] args){
16         A obj = new A();
17         obj.a();
18         obj.b();
19     }
20 }
21
```

```
PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> c:\
21.0.6.7-hotspot\bin\java.exe' -XX:+ShowCodeDetailsInExcep
A
B
PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> █
```

```
1 interface ex1{
2     public abstract void a();
3 }
4
5 interface ex2{
6     public abstract void b();
7 }
8
9 class A implements ex1, ex2{
10     public void a() {System.out.println("A");}
11     public void b() {System.out.println("B");}
12 }
13
14 public class App {
15     public static void main(String[] args){
16         A obj = new A();
17         obj.a();
18         obj.b();
19     }
20 }
21
```

```
PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> c
21.0.6.7-hotspot\bin\java.exe' -XX:+ShowCodeDetailsInExc
A
B
PS C:\Users\kmjoo\OneDrive\바탕 화면\Java\VScode\Test> █
```