

자바의 상속
<p>1. 상속이란 자식이 부모가 가지고 있는 재산이나 권력을 물려받는다는 의미이다.</p> <p>2. 상속문법  형식) class 자식클래스명 extends 부모클래스명{    }  extends 키워드로 부모클래스로부터 상속을 받는다.  클래스 상속은 하나의 부모로 부터만 단일 상속만 가능하다.</p> <p>3. 자바 클래스의 최상위 조상클래스는 Object 이다. 다른 클래스로부터 명시적인 상속을 받지 않으면 자동적으로 Object 클래스로부터 상속을 받는다.</p>

메서드 오버라이딩 이란
<p>1. 자손 클래스에서 부모 클래스의 기존 메서드와 시그니처 즉, 이름, 전달인자의 자료형과 전달인자의 개수,반환타입을 동일하게 정의한다. 부모클래스로 부터 상속받은 메서드 내용을 자식 클래스에 맞게 변경하는 것을 오버라이딩 이라 한다.</p> <p>2. 메서드 오버라이딩 전제 조건  반드시 부모 자식간의 상속관계를 만들어야 한다.</p>

super
<p>1. 상속관계에서 자식클래스 영역에서 부모클래스의 메서드가 오버라이딩 된 경우 부모클래스의 메서드를 호출하려면  super.메서드(); 로 접근한다.</p> <p>2. super는 자손클래스에서 조상 클래스로부터 상속받은 멤버 변수를 참조하는 데 사용되는 참조변수이다.</p> <p>3. super는 정적메서드내에서는 사용할 수 없고,인스턴스 메서드에서만 사용한다.</p> <p>4. 상속관계에서 자식클래스에서 부모클래스부터 상속받은 멤버변수에 접근하려면  super.변수명; 으로 접근한다. 즉 조상클래스 멤버 변수 이름과 동일한 이름을 자손클래스에서 정의 했을 경우 자손 클래스에서 조상클래스로부터 상속받은 멤버 변수에 접근할 때 super.변수명으로 접근해서 값을 가져온다. 자손 클래스 멤버변수에 접근할 때는 this.변수명으로 값을 가져온다.</p>

상속에서의 생성자 호출 문제
<ol style="list-style-type: none"> <li>1. 전달인자가 없는 생성자를 기본생성자라 한다.</li> <li>2. 생성자가 오버로딩 되면 자바는 기본생성자를 묵시적으로 제공하지 않는다.</li> <li>3. 그러므로 생성자가 오버로딩 될 때 자식클래스에서 묵시적으로 제공하지 않는 부모클래스 기본생성자를 호출하고자 할때 문제가 발생한다. 그러므로 부모클래스에서 생성자가 오버로딩 되면 상속에서의 생성자 호출문제가 발생할 수 있으므로 명시적인 기본생성자를 정의하는 것이 좋다.  라. 생성자는 상속되지 않는다. 그러므로 자식클래스에서 new 클래스명();에 의해서 생성자를 호출하면 자바는 먼저 부모클래스 기본생성자를 먼저 호출하도록 설계되어져 있다.</li> <li>4. 자식클래스에서 부모클래스 기본생성자를 호출하고자 할때 <code>super();</code>로 한다. 이 부분은 기본으로 생략 되어져 있다.</li> <li>5. <code>super()</code>는 조상 클래스 생성자를 호출할 때 사용한다.</li> <li>6. <code>this()</code>는 같은 클래스 내의 다른 생성자를 호출할 때 사용한다.</li> <li>7. <code>super(인자값,인자값);</code> 으로 조상클래스 오버로딩 된 생성자를 호출하도록 인위적으로 경로를 바꿀 수 있다.</li> </ol>

package와 import
<ol style="list-style-type: none"> <li>1. package 는 클래스 묶음을 뜻한다. 그러므로 이 키워드로 클래스 묶음을 저장할 수 있는 폴더를 생성할 수 있다.</li> <li>2. import 키워드는 외부 패키지 클래스를 읽어올때 사용한다.  이클립스 임포트 단축키는 <code>ctrl+shift+o</code>(영문) 이다.</li> <li>3. 자바 기본 패키지 경로는 <code>java.lang</code>이다. 이 패키지 경로는 기본값으로 생략해도 된다.</li> </ol>

자바의 접근 지정자
<ol style="list-style-type: none"> <li>1. <b>private</b> : 내 자신 클래스 내에서만 접근 가능하다. 자바의 보안성</li> <li>2. <b>기본 접근 지정자(생략)</b>: 같은 패키지의 내 자신클래스 와 다른 클래스에서 접근 가능하다.</li> <li>3. <b>protected</b> : 같은 패키지 내의 내 자신 클래스와 다른 클래스 또는 상속받은 자식 클래스에서 접근 할수 있다. 특히 상속받은 자식클래스에서 접근할 때는 외부 패키지에서도 접근할 수 있다.</li> <li>4. <b>public</b> : 같은 패키지 뿐만 아니라 외부 패키지 다른 클래스에서도 접근 할 수 있다. 누구나 다 접근 할 수 있다.</li> </ol>