Inheritance

1. 상속기능 사용하기

|  |
| --- |
| **class** Parent{    }  **class** Child Parent{  **int** childVar = 20;  **void**  {  System.***out***.println("child Method");  }  }  **public** **class** ExtendsTest {    **public** **static** **void** main(String[] args) {    System.***out***.println("child.parentVar = " + child.parentVar);  System.***out***.println("child.childVar = " + child.childVar);  child.parentMethod();  child.childMethod();  }  } |
| child.parentVar = 10  child.childVar = 20  parent Method  child Method |

상위클래스를 상속받은 클래스를 다른 크래스가 다시 상속한 경우

|  |
| --- |
| **class** Super{  **int** x = 90;  **void** superMethod(){  System.***out***.println("super Method");  }  }  **class** Sub **extends** Super{  **int** y = 10;  **void** subMethod(){  System.***out***.println("sub method");  }  }  **class** Subsub **extends** Sub{  **int** z = 20;  **void** subSubMethod(){  System.***out***.println("subSub method");  }  }  **public** **class** ExtendsTest2 {    **public** **static** **void** main(String[] args) {    Subsub ss = **new** Subsub();  System.***out***.println("ss.x = " + ss.x);  System.***out***.println("ss.y = " + ss.y);  System.***out***.println("ss.z = " + ss.z);    ss.subMethod();  ss.subMethod();  ss.subSubMethod();  }  } |
| ss.x = 90  ss.y = 10  ss.z = 20  sub method  sub method  subSub method |

2. Overriding

①자식 클래스에서 메소드를 재정의한 경우

|  |
| --- |
| **class** OverridingParent{  **void** parentMethod(){  System.***out***.println("parent method");  }  }  **class** OverridingChild **extends** OverridingParent{  **void** parentMethod(){  System.***out***.println("overridng method");  }  }  **public** **class** OverrdingTest {  **public** **static** **void** main(String[] args) {      }  } |
| overridng method |

3. super & super( )

① super 레퍼런스 변수 사용하기

|  |
| --- |
| **class** Sawon{  String name;  String dept;  **int** salary;    }  **class** Sales **extends** Sawon{  **int** commition;  /\*String displayInfo(){  return "이름 : " + name + ", 부서 : " + dept + ", 연봉 : " + salary  + ", commition = " + commition;  }\*/  String displayInfo(){  **return** **super**.displayInfo() + ", 수당 : " + commition;  }  }  **public** **class** SuperTest {      }  } |
| 이름 : null, 부서 : null, 연봉 : 0, 수당 : 0 |

②super( ) 용도 살펴보기

|  |
| --- |
| **class** Sawon2{  String name;  String dept;  **int** salary;      }  **class** Sales2 **extends** Sawon2{  **int** commition;      /\*String displayInfo(){  return "이름 : " + name + ", 부서 : " + dept + ", 연봉 : " + salary  + ", commition = " + commition;  }\*/  String displayInfo(){  **return** **super**.displayInfo() + ", 수당 : " + commition;  }  }  **public** **class** SuperConstructorTest {  **public** **static** **void** main(String[] args) {    Sales2 sales2 = **new** Sales2("홍길동","영업부",100000000,70000000);  System.***out***.println(sales2.displayInfo());  }  } |
| 이름 : 홍길동, 부서 : 영업부, 연봉 : 100000000, 수당 : 70000000 |

② 상위 클래스의 변수 호출

|  |
| --- |
| **class** Sawon3{  String name = "홍길동";  }  **class** Sales3 **extends** Sawon3{  String name = "샤이니";  String displayInfoLocal(){  String name = "이정";  **return** name;  }  String displayInfoThis(){  String name = "이정";  **return** **this**.name;  }  String displayInfoSuper(){  String name = "이정";  **return** **super**.name;  }  }  **public** **class** SuperVariableTest {  **public** **static** **void** main(String[] args) {      }  } |
| local.name = 이정  this.name = 샤이니  super.name = 홍길동 |