



11

COMPUTER PROGRAMMING

INHERITANCE



CONTENTS

- OVERVIEW OF INHERITANCE
- INHERITANCE OF MEMBER VARIABLE
- RESERVED WORD – SUPER
- METHOD INHERITANCE and OVERRIDING
- INHERITANCE and CONSTRUCTOR



Overview of Inheritance

□ Declaration format of Class including Inheritance

```
[public/final/abstract] class ClassName extends SupperClassName {  
    ..... // Declaration of member variable  
    ..... // Constructor  
    ..... // Declaration of method  
}
```

Inheritance of Member Variable

□ Example of Member Variable Inheritance

```
class A {
    int aa = 1;
}
class B extends A {
    int bb = 2;
}
class C extends B {
    int cc = 3;
}
class Dabc {
    public static void main(String[] args) {
        C objc = new C();
        System.out.println("objc객체의 객체속성변수 aa의 값은 " + objc.aa);
        System.out.println("objc객체의 객체속성변수 bb의 값은 " + objc.bb);
        System.out.println("objc객체의 객체속성변수 cc의 값은 " + objc.cc);
    }
}
```

Inheritance of Member Variable

```
class C1 {
    static int x;
    static int y;
}

class C2 extends C1 {
    static String x;
}

class Inheritance3 {
    public static void main(String args[]) {
        C2.x = "알기쉽게 해설한 자바";
        C2.y = 20000;
        C1.x = 30000;
        System.out.println("클래스 변수 C2.x 값 : " + C2.x);
        System.out.println("클래스 변수 C2.y 값(C1으로부터 상속) : " + C2.y);
        System.out.println("클래스 변수 C1.x 값 : " + C1.x);
    }
}
```

Reserved Word - super

```
class D1 {
    int x = 1000;
    void display() {
        System.out.println("상위클래스 D1의 display() 메소드 입니다");
    }
}
class D2 extends D1 {
    int x = 2000;
    void display() {
        System.out.println("하위클래스 D2의 display() 메소드 입니다");
    }
    void write() {
        this.display();
        super.display();
        System.out.println("D2 클래스 객체의 x 값은 : " + x);
        System.out.println("D1 클래스 객체의 x 값은 : " + super.x);
    }
}
class InheritanceSuper {
    public static void main(String args[]) {
        D2 d = new D2();
        d.write();
    }
}
```

Method Inheritance and Overriding

```
class A {
    int i;
    int j;
    void setij(int x, int y) {
        i = x;
        j = y;
    }
}

class B extends A {
    int total;
    void sum() {
        total = i + j;
    }
}

class Inheritance4 {
    public static void main(String args[]) {
        B subOb = new B();
        subOb.setij(10, 12);
        subOb.sum();
        System.out.println("두수의 합계는 : " + subOb.total);
    }
}
```

Method Inheritance and Overriding

```
class A {  
    void show(String str) {  
        System.out.println("상위클래스의 메소드 show(String str) 수행 " + str);  
    }  
}  
  
class B extends A {  
    void show() {  
        System.out.println("하위클래스의 메소드 show() 수행");  
    }  
}  
  
class OverrideExam1 {  
    public static void main(String args[]) {  
        B over = new B();  
        over.show("알기쉽게 해설한 자바");  
        over.show();  
    }  
}
```


Method Inheritance and Overriding

```
class A {  
    void show() {  
        System.out.println("상위클래스의 메소드 show(String str) 수행 ");  
    }  
}  
  
class B extends A {  
    void show() {  
        System.out.println("하위클래스의 메소드 show() 수행");  
    }  
}  
  
class OverrideExam2 {  
    public static void main(String args[]) {  
        B over = new B();  
        over.show();  
    }  
}
```

Method Inheritance and Overriding

```
class A {
    int i, j;
    A(int a, int b) {
        i = a;
        j = b;
    }
    void show() {
        System.out.println("상위클래스의 메소드 show() 수행");
    }
}

class B extends A {
    int k;
    B(int a, int b, int c ) {
        super(a,b); ←
        k = c;
    }
    void show() {
        System.out.println("하위 클래스의 메소드 show() 수행");
        System.out.println("===super를 이용한 상위 클래스 메소드 호출===");
        super.show(); ←
    }
}

class OverrideExam3 {
    public static void main(String args[]) {
        B over1 = new B(10, 20, 30);
        System.out.println("i, j, k의 값 : " + over1.i + " " + over1.j + " " + over1.k);
        over1.show();
    }
}
```

Inheritance and Constructor

```
class A1 {  
    double d1;  
    ① → A1() { ←  
        System.out.println("클래스 A1의 생성자 수행");  
        d1 = 10*10;  
    }  
}  
  
class A2 extends A1 {  
    double d2;  
    ② → A2() { ←  
        System.out.println("클래스 A2의 생성자 수행");  
        d2 = 10*10*10;  
    }  
}  
  
class A3 extends A2 {  
    double d3;  
    ③ → A3() { ←  
        System.out.println("클래스 A3의 생성자 수행");  
        d3 = 10*10*10*10;  
    }  
}  
  
class Constructors1 {  
    public static void main(String args[]) {  
        A3 super1 = new A3(); ←  
        System.out.println("10의 2제곱 : " + super1.d1);  
        System.out.println("10의 3제곱 : " + super1.d2);  
        System.out.println("10의 4제곱 : " + super1.d3);  
    }  
}
```

Inheritance and Overriding

```
class A1 {
    int d1;
    int s;
    A1(int s1) {
        System.out.println("클래스 A1의 생성자 수행");
        s = s1;
        d1 = s * s ;
    }
}
class A2 extends A1 {
    int d2;
    int t;
    A2(int s1, int t1) {
        super(s1);
        System.out.println("클래스 A2의 생성자 수행");
        t = t1;
        d2 = t * t ;
    }
}
class Constructors2 {
    public static void main(String args[]) {
        A2 super2 = new A2(10,20);
        System.out.println("10의 제곱은 : " + super2.d1);
        System.out.println("20의 제곱은 : " + super2.d2);
    }
}
```

Type Conversion of Object

□ Object type conversion in classes of inheritance relation

```
class Acast {
    int a=1;
}
class Bcast extends Acast {
    int b=2;
}
class Ccast extends Bcast {
    int c=3;
}

class TestCasting {
    public static void main(String[] args) {
        Acast refA; ← Acast 타입의 객체/refA 선언
        refA = new Ccast(); ←
        System.out.println("refA.a의 값은 "+refA.a);
    }
}
```

CONCLUDE

- OVERVIEW OF INHERITANCE
- INHERITANCE OF MEMBER VARIABLE
- RESERVED WORD – SUPER
- METHOD INHERITANCE and OVERRIDING
- INHERITANCE and CONSTRUCTOR

