《 상속을 정리하는 문제 10개 》

출처: www.youtube.com/@weekendcode // https://inf.run/zSrvA

상속에 대해서 헷갈리는 분이 많아서 유형별로 정리했습니다.

1. 부모를 명시적으로 호출하고, 부모 생성자끼리 호출
2. 부모와 자식 클래스의 메서드에서 this로 각각의 인스턴스 변수를 참조
3. 부모 객체와 자식 객체 각각의 메서드를 호출
4. 자식 클래스에서는 this 사용, 부모 클래스에서는 this를 사용하지 않는 경우
5. 부모의 메서드를 오버라이딩하지 않고 자식에서 새로운 메서드를 정의
6. 부모 클래스의 private 메서드는 상속되지 않음
7. private이지만 에러가 발생하지 않는 경우
8. 부모 클래스의 static 메서드는 오버라이딩 되지 않는다. (hiding)
9. 자식 클래스에서 부모 클래스의 메서드를 명시적으로 호출
10. 부모 생성자의 매개변수를 자식 생성자가 전달하지 않았을 경우

개별 코드를 보면서, 상속이 어떻게 이뤄지고 객체가 생성된 이후에 어떤 메서드가 실행되는지 주의 깊게 살펴보세요.

```
class Parent {
    int x = 25;
    Parent() {
        this(5);
        System.out.println("Parent default");
    }
    Parent(int x) {
        System.out.println("Parent: " + x);
    }
}
class Child extends Parent {
    int x = 10;
    Child() {
        super();
        this.display();
        System.out.println("Child default");
    }
    void display() {
        System.out.println(x);
    }
}
public class TestInheritance {
    public static void main(String[] args) {
        Child child = new Child();
    }
}
```

출력값:

```
Parent: 5
Parent default
10
Child default
```

```
class Parent {
    int value = 10;

    void show() {
        System.out.println("Parent: " + this.value);
    }
}

class Child extends Parent {
    int value = 20;

    void show() {
        System.out.println("Child: " + this.value);
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.show();
    }
}
```

출력값:

Child: 20

```
class Parent {
    void method() {
        System.out.println("Parent");
    }
}

class Child extends Parent {
    void method() {
        System.out.println("Child");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Parent parentObj = new Parent();
        Child childObj = new Child();

        parentObj.method();
        childObj.method();
    }
}
```

출력값:

Parent

Child

```
class Parent {
    int y;

    Parent(int a) {
        y = a;
        System.out.println("Parent: " + y);
    }
}

class Child extends Parent {
    int x;

    Child(int x) {
        super(x + 1);
        this.x = x;
        System.out.println("Child: " + this.x);
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Child child = new Child(5);
    }
}
```

출력값:

Parent: 6

Child: 5

```
class Parent {
    void show() {
        System.out.println("Parent show");
    }
}

class Child extends Parent {
    void show(int x) {
        System.out.println("Child show: " + x);
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.show();
    }
}
```

출력값:

Parent show

```
class Parent {
    private void display() {
        System.out.println("Parent display");
    }
}

class Child extends Parent {
    void display() {
        System.out.println("Child display");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.display();
    }
}
```

출력값:

에러 발생.

obj.display(); 에러

```
class Parent {
    private void display() {
        System.out.println("Parent display");
    }
}

class Child extends Parent {
    void display() {
        System.out.println("Child display");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.display();
    }
}
```

출력값:

```
Child display
```

```
class Parent {
    static void display() {
        System.out.println("Parent static display");
    }
}

class Child extends Parent {
    static void display() {
        System.out.println("Child static display");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.display();
    }
}
```

출력값:

Parent static display

```
class Parent {
    void display() {
        System.out.println("Parent display");
    }
}

class Child extends Parent {
    void display() {
        super.display();
        System.out.println("Child display");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.display();
    }
}
```

출력값:

```
Parent display
Child display
```

```
class Parent {
    int value;

    Parent(int x) {
        this.value = x;
        System.out.println("Parent value: " + value);
    }
}

class Child extends Parent {
    Child() {
        System.out.println("Child created");
    }
}

public class TestInheritance {
    public static void main(String[] args) {
        Child obj = new Child();
    }
}
```

출력값:

Child() 부분 오류 발생