

1

생성자 : 인스턴스가 생성될 때 호출되는 '인스턴스 초기화'를 실행하며 클래스 멤버 변수의 초기화를 목적으로 한다. 생성자 역시 메소드처럼 클래스 내에 선언되며, 구조도 메소드와 유사하다.

- 생성자 규칙

- ① 생성자는 객체를 생성할 때 자동으로 호출되며 목적은 멤버변수 초기화이다.
- ② 생성자도 접근 지정자를 가진다.
- ③ overload 할 수 있으며 this()라는 키워드로 내부 호출 된다.
- ④ 생성자는 user가 명시하지 않으면 default 생성자가 내부적으로 호출되어 메모리 할당 시 변수를 초기화한다.
- ⑤ 생성자를 하나라도 명시하게 되면 명시된 생성자를 호출하면서 메모리를 할당한다.
- ⑥ 생성자는 외부, 내부 메소드처럼 호출할 수 없고, 객체 생성 시 단 한 번만 호출된다.
- ⑦ 생성자라는 클래스 이름과 같지만 반환형은 없다.

2

상속 : 객체지향개념의 가장 중요한 코드의 재사용을 프로그램으로 구현하는 부분이다.
프로그램을 전체를 새로 개발한 것이 아니므로 프로그램 개발기간의 단축과, 비용의 감소를 가져올 수 있고, 이미 실 시스템에서 안정적으로 돌아가는 코드부분을 사용하기 때문에 시스템이 더 안정적이다.

- 새로 작성해서 기존의 코드를 가져다 쓰는 클래스가 상속을 받는 클래스
 - ➔ 서브 클래스(sub class)또는 자식 클래스(child class) 또는 파생 클래스(derived class)
- 기존에 이미 만들어져서 상속해주는 클래스
 - ➔ 슈퍼 클래스(super class)또는 베이스 클래스(base class)또는 부모 클래스(parent class)
- 자바 프로그램에서 상속은 extends라는 키워드를 써서 표현한다.

2

• 상속 특징

- 1 상속은 단일 상속을 원칙으로 한다. 즉 후손이 슈퍼 클래스는 하나밖에 가질 수 없다.
- 2 서브 클래스는 몇 단계로 만들 수 있다.
- 3 멤버 변수, 메소드는 상속된다.
- 4 후손 클래스의 변수와 메소드는 사용할 수 없다. 즉, 인스턴스화 한 클래스가 슈퍼 클래스에서 상속된 기능밖에 사용할 수 없다.
- 5 생성자는 상속되지 않는다. 클래스 고유의 것이다.
- 6 후손의 멤버 변수 이름, 메소드 이름이 겹치면 인스턴스화 된 클래스에서 가장 계층이 가까운 선조 클래스의 메소드가 호출된다. 즉, 상속 시에 기존 멤버 변수 방법을 변경할 수 있다.
- 7 후속이 선조의 멤버 혹은 생성자를 호출할 때 super.멤버, super()로 호출한다.

3

오버라이딩

- 상속관계에서 발생하는데, 슈퍼 클래스에서 상속받은 메소드를 그대로 사용하는 것이 아니라, 서브클래스에 맞도록 메소드의 내용을 수정해서 사용하는 것을 말한다.
- 슈퍼 클래스와 서브 클래스에 같은 이름을 가진 메소드가 존재할 때, 슈퍼 클래스의 메소드를 무시하고(override), 서브 클래스의 메소드를 사용하는 것으로, 메소드를 재정의하는 것이다.
- 오버라이딩 시 주의점
 - 1 슈퍼 클래스의 메소드를 오버라이딩하는 서브 클래스의 메소드는 메소드의 시그니처는 물론 리턴 타입까지 일치해야 한다. 일치하지 않으면 오버라이딩이 되지 않는다.
 - 2 서브 클래스에서 재정의 되는 메소드의 접근제어자는 슈퍼 클래스의 오버라이드 되는 메소드의 접근제어자와 같거나, 제한범위가 넓은 접근제어자를 사용한다.