

1. 상속 (Inheritance)

□ 정의

상속은 기존 클래스(부모 클래스, 슈퍼 클래스)의 속성과 메서드를 새로운 클래스(자식 클래스, 서브 클래스)가 물려받는 기능이다.

코드 재사용성을 높이고, 유지 보수성과 확장성을 향상시킬 수 있다.

□ 기본 문법

```
class 부모클래스 {  
    // 필드, 생성자, 메서드  
}class 자식클래스 extends 부모클래스 {  
    // 부모의 멤버를 사용할 수 있음  
}
```

□ 특징

자식 클래스는 부모 클래스의 public 및 protected 멤버를 상속받음
부모 클래스의 생성자는 상속되지 않지만, super()를 통해 호출 가능
하나의 클래스는 단일 클래스만 상속 가능(Java는 단일 상속만 지원)

2. 다형성 (Polymorphism)

□ 정의

하나의 객체가 여러 가지 형태로 동작할 수 있는 성질

즉, 부모 타입의 참조 변수로 자식 클래스 객체를 다룰 수 있게 하는 기능

□ 종류

컴파일 타임 다형성 (정적 바인딩): 메서드 오버로딩 (같은 이름, 다른 매개변수)

런타임 다형성 (동적 바인딩): 메서드 오버라이딩 + 업캐스팅

□ 기본 문법

부모클래스 참조변수 = new 자식클래스();

□ 장점

코드의 유연성과 확장성 증가

인터페이스와 함께 사용할 때, 실제 구현 클래스를 몰라도 객체를 사용할 수 있음

3. 추상 클래스 (Abstract Class)

□ 정의

추상 클래스는 객체를 생성할 수 없는 클래스이며, 상속을 통해 기능을 구현 해야 함.

하나 이상의 추상 메서드(본문이 없는 메서드)를 포함

자식 클래스는 이 추상 메서드를 반드시 오버라이딩(재정의) 해야 함

□ 기본 문법

```
abstract class 클래스이름 {  
    abstract 반환타입 메서드이름(); // 추상 메서드  
    void 일반메서드() {  
        // 구현 가능  
    }  
}
```

□ 특징

- 추상 클래스는 생성자, 필드, 일반 메서드를 가질 수 있음
- 강제성 있는 설계 템플릿을 제공
- 다형성과 함께 쓰일 때 유용

비교

항목	상속	다형성	추상 클래스
핵심 개념	기존 클래스 기능 물려받음	하나의 타입으로 다양한 객체 사용	구현 강제 및 공통 기능 제공
객체 생성	가능	가능	직접 생성 불가 (상속 통해 사용)
문법 키워드	extends	업캐스팅 + 오버라이딩	abstract
주요 목적	코드 재사용	유연한 설계 및 확장성	공통 설계 제공 + 오버라이딩 강제화