\*Interface ppt8장,

-interface : 개발할때 규칙을 미리세워두는 장치

-상수필드만 선언가능 public static final이 자동으로 붙음

-인터페이스 안의 메소드는 추상메소드 public abstract가 자동으로 붙음

-defalt 형 메소드를 붙여 일반메소드를 추가할수있다(java 8부터 추가됨) 메소드 구현됨

-static 정적 메소드를 만들수있다(java 8부터 추가됨), 객체없이 인터페이스 호출가능하며 메소드구현

-직접객체를 생성하진 못하지만 implement한 class의객체를 생성할수있다 해당클래스에서 @Override하여야함

Ex) interface Remote, class Tv, class Audio가 있다면

Remote remote, remote1;

remote = new Tv();, remote1 = new Audio();로 각각의 implement한 class의 객체를 만들수있다

객체의 다형성 7장 ->부품화하여 교체할수있다

-익명 구현 객체 : 해당클래스에서만 일회성으로 사용할경우 주로사용(모바일할 때 주로사용)

new 인터페이스(){

// 인터페이스에 선언된 추상메소드를 실제로 구현하면 new생성자로 인터페이스 객체를 만들수있다

}.메소드();//일회용으로 사용할려는 목적

-상속과 마찬가지로 다형성이 적용되어 자동타입변환, Override,

Interface Vehicle, class Bus가있고 Vehicle에는 run(), Bus에는 run(),checkFare()메소드가있을 때

Vehicle vehicle = new Bus(); //Vehicle형 변수 vehicle을 만들고 그안에 객체 Bus를 만든다

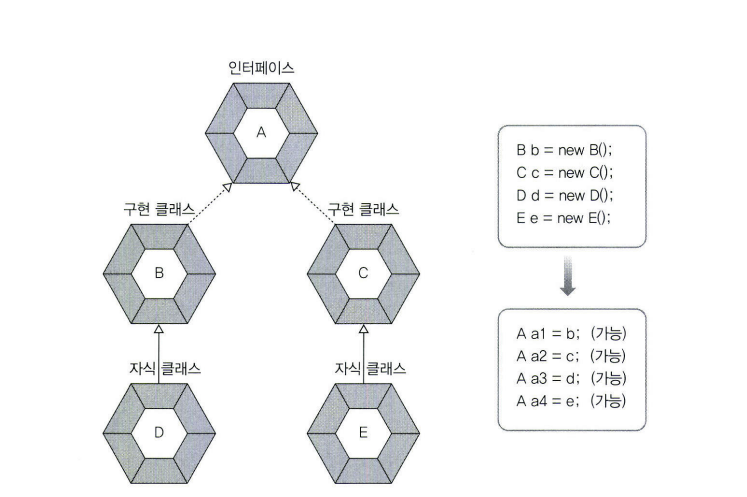
vehicle.run();

// vehicle.checkFare(); 사용불가

Bus bus = (Bus) vehicle; // 강제 타입변경

Bus.run();

Bus.checkFare();



-구현한 객체를 배열로 관리할시 제어문에서 가장많이 혜택을본다

For(Tire tire : tires){ // 전체타이어의 roll메소드를 호출한다

Tire.메소드();

}

-interface도 다른 interface를 상속받을수있으며 다중 interface를 허용한다

default메소드를 자식인터페이스에서 추상메소드로 재정의할수있다

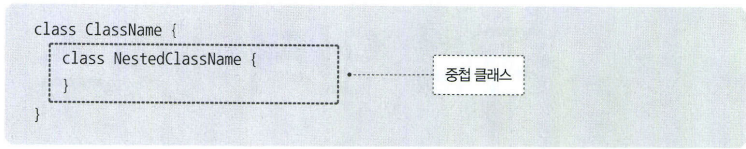
-다중인터페이스 구현 : 여러게의 인터페이스를 implements 할수있다

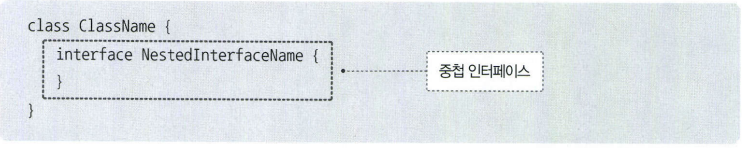
Ex) Interface C extends A,B라면

Class D implement C-> 추상 메소드 A,B,C를 구현 클래스D에서 모두 구현해야한다

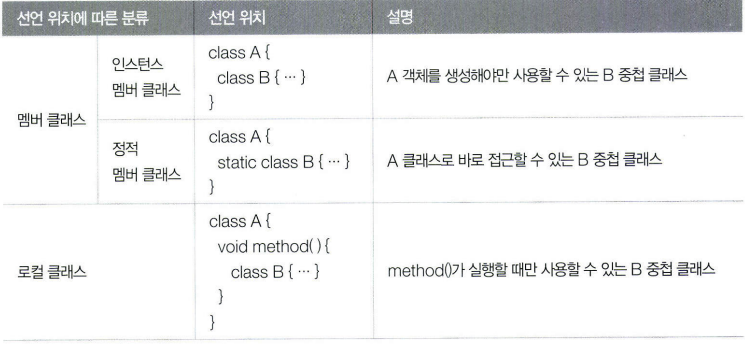
\*chapter 9 중첩클래스와 중첩 인터페이스(Nested Class, Interface)

- 특정 클래스와 관계를 가질때에 클래스, 인터페이스 내부에 클래스,인터페이스를 선언하는 것이 좋다 두클래스의 멤버들을 서로 쉽게 접근할수있으며(긴밀한관계) 외부에는 불필요한 클래스간의 관계를 감춤으로 코드의 복잡성을 낮춘다, 중첩인터페이스는 주로 UI프로그래밍에서 이벤트를 처리할 목적으로 많이 활용된다





-중첩클래스



🡪

-compile시 멤버클래스는 bin 폴더에 들어가보면 A$B.class로 변환되고(A바깥 클래스, B 멤버class

-로컬클래스는 A$1B.class로 complile된다

-인스턴스 멤버클래스는 바깥클래스 객체가 생성되어야 인스턴스 멤버클래스의 객체를 생성할수있다