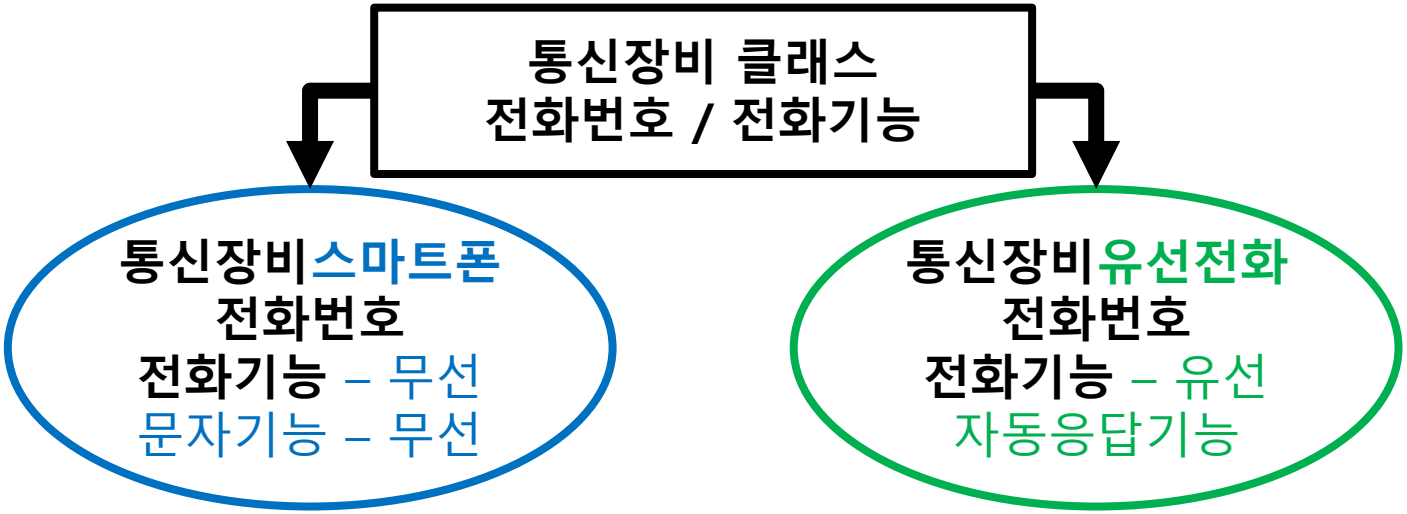

K G 아 이 티 뱅 크
JAVA
J A V A

추상 클래스

추상 클래스

- 객체가 공유하고 있는 개념만 정의하는 상속용 클래스
  - 객체들을 동일할 특성과 동작을 가지도록 하는 개념
  - 동작은 고유의 동작이 되도록 재정의를 강요하는 것
  - 통신장비는 휴대폰과 일반전화기가 될 수 있다.
    - 큰 개념은 같지만, 세부속성/동작은 모두 다름
- 큰 틀만 준비하며, 상속받아 구체화가 되어야 함



추상 클래스

❖ 추상클래스의 주의사항 : 상속받고, 재정의를 해야 함

❖ 추상클래스

```
public abstract class Super {
    protected int data1;
    public Super( .. ) { .. }
    public abstract void method1();
}
public class Sub extends Super {
    private int data2;
    public Sub( .. ) {
        super( ... ); data2 = ..;
    }
    public void method1() { .. }
}
```

❖ 추상 부모클래스

abstract 키워드 사용
해당 클래스로는
객체 생성 불가

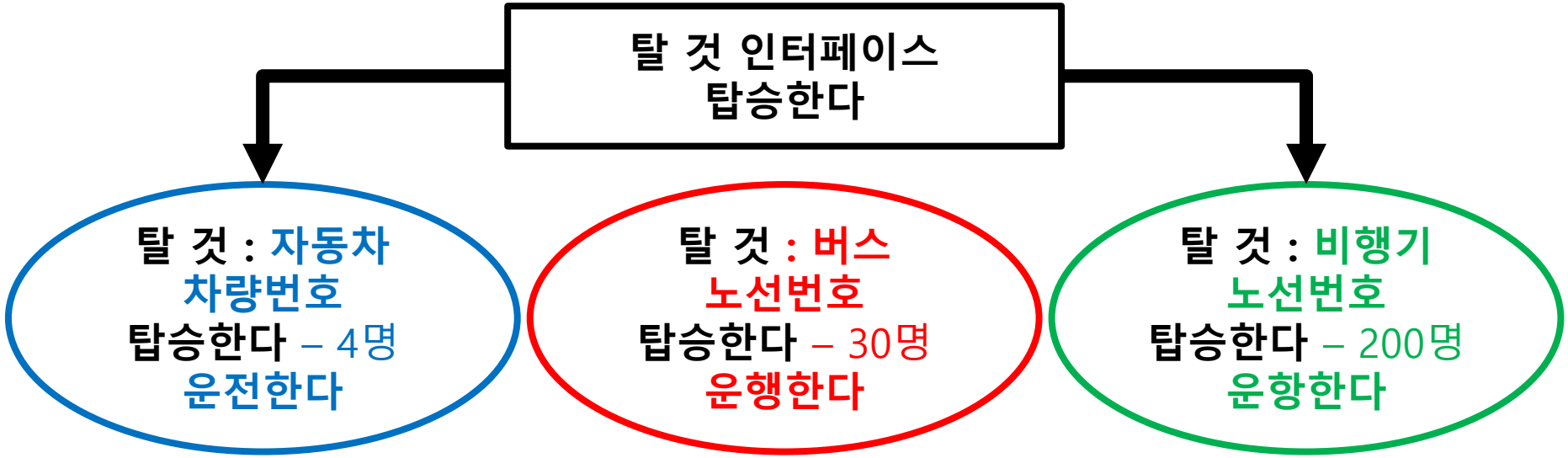
❖ 자식클래스

구체화시키는 개념
추상화된 부분은
전부 작성되어야 함

인터페이스

인터페이스

- **만들어진 객체들의 사용성만 보장하는 상속용 클래스**
  - 서로 다른 클래스의 객체의 메서드를 통일하기 위함
  - 서로 다른 객체여도 동일하게 다룰 수 있음
- **기본적인 개념은 상속이나 상속과의 차이점이 있음**
  - 상속 – **특성 및 동작을 최대한 통일시켜 이용**하는 것
  - 인터페이스 – 특성보다 **동일한 동작**만 보장하는 것



인터페이스

❖ 인터페이스의 주의사항 : 전부 추상 메서드로 취급함

❖ 인터페이스 상속

```
public Interface Action {
     int value = 0;
     void method1();
     default void method2(..) { .. }
}
public class Sub implements Action
{
    int data;
    public Sub( .. ) { n3 = ..; }
    public void method1( .. ) { .. }
}
```

❖ 인터페이스

추상클래스와 유사
생략된 접근제어 및
속성키워드가 존재함

❖ 상속한 클래스

반드시 재정의를 함
붙어있는 키워드에
따라서 안하기도 함


익명 객체

익명 객체

❖ **부모 클래스를 이용해 임시 자식 객체를 만드는 문법**

- ➢ 모든 객체를 위한 클래스를 정의하는 것은 불가능
- ➢ 특정 영역에서 필요한 1회성 객체를 만들어 쓰는 기술

❖ **일반 클래스로 만들면 융통성이 떨어져서 힘듬**

- ➢ **주로 추상클래스 / 인터페이스를 이용하여 생성하게 됨**

통신장비 클래스 전화번호 / 전화기능	탈 것 인터페이스 탑승한다
통신장비 : 실전화기 전화번호 없음 전화기능 – 유선(실)	탈 것 : 코끼리 특징 : 거대함 탑승한다 – 등에 탑승가능
통신장비 : 전서구 전화번호 없음 전화기능 – 편지(오래걸림)	탈 것 : 이카로스의 날개 날개수량 : 2개 탑승한다 – 팔에 부착함

익명 객체

❖ 익명 객체의 주의사항 : 필드추가 및 오버라이딩만 가능

❖ 익명 객체의 생성

```
public class Super {
    protected int n1;
    public Super( .. ) { .. }
    public void method1( .. ) { .. }
}
익명객체가 필요한 특정 메서드 {
 Super super1 = new Super(){
    private int n2 = 100;
    public void method1( .. ) { .. }
    public void method2( .. ) { .. }
    };
}
```

❖ 부모클래스

상속할려는 클래스
기본적으로
상속받는 구조임

❖ 익명클래스

급조하는 객체
super()와 super를
이용할 수 있음

❖ 새로운 메서드

정의 자체는 가능
단, 호출하려고 하면
상당히 번거로움