

# this와 super



# 매개변수가 멤버변수를 가린다

this

- this는 자기 자신 객체를 지정할 때 사용하는 키워드입니다.
- this. 을 사용하면 동일 클래스 내의 멤버(멤버변수, 메서드)를 참조할 수 있습니다.
- this()를 사용하면 생성자 내부에서 자신의 다른 생성자를 호출할 수 있습니다.

```
public class Person {
```

```
    String name;  
    int age;
```

```
    Person(String name, int age) {  
        name = name;  
        age = age;  
    }
```

```
}
```

같은 이름의 매개변수가  
멤버변수의 사용을 가린다

이 때  
나 자신의 멤버로 지칭하는 키워드가 this이다

# this 예제

inheritance/this\_/Person.java

```
1: package inheritance.this_;
2:
3: public class Person {
4:
5:     public String name;
6:     public int age;
7:
8:     public Person(String name, int age) {
9:         this.name = name;
10:        this.age = age;
11:    }
12:    public Person(String name) {
13:        this(name, 1);
14:    }
15:    public Person() {
16:        this("이름없음", 1);
17:    }
18:
19:    public String getDetails() {
20:        return "이름: " + name + "\t나이: " + age;
21:    }
22: }
```

this. 로 나의 멤버변수의 접근 가능하다

this() 로 나의 생성자의 접근이 가능하다



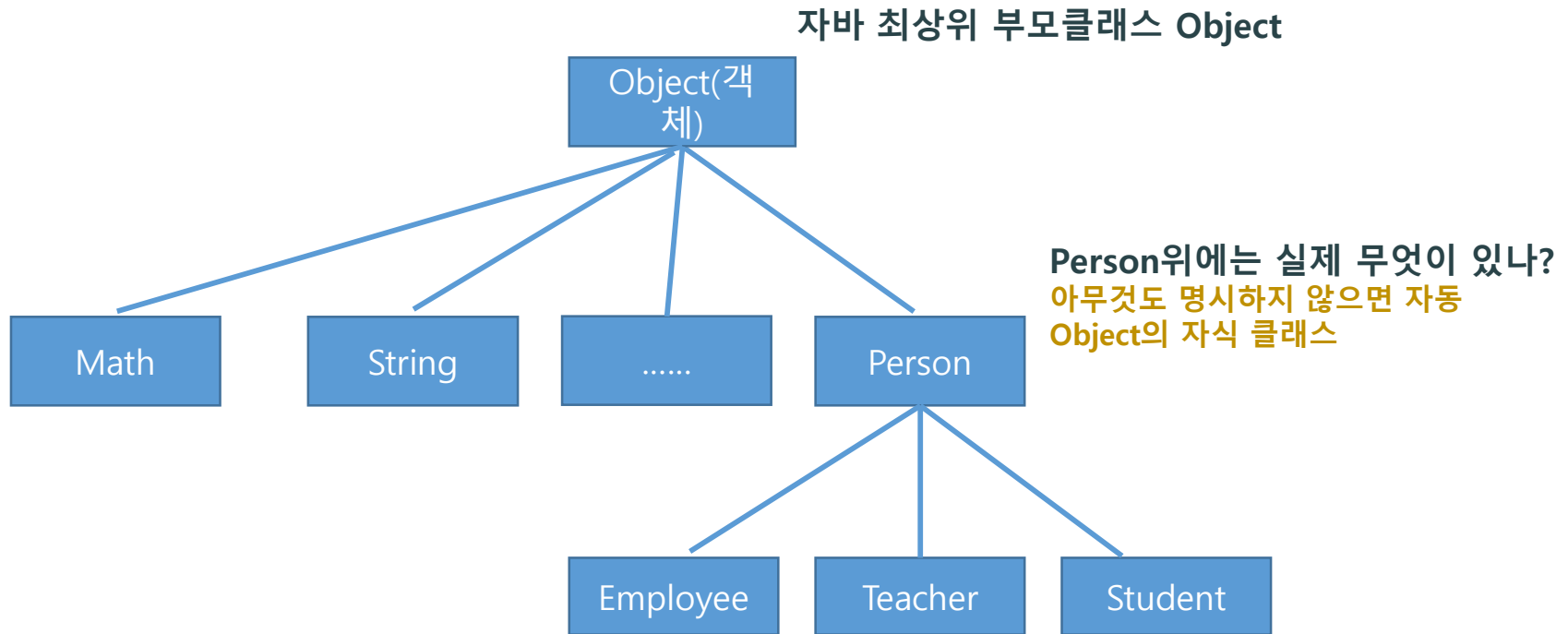
# super

## super

- super는 **한단계 위 부모클래스**의 객체를 지정할 때 사용하는 키워드입니다.
- super. 을 사용하면 **부모클래스의 멤버**를 참조할 수 있습니다.
- super()는 생성자 내부에서만 사용이 가능하며, **부모클래스의 생성자**를 호출하는 데 사용합니다.
- 생성자의 **첫 라인에는 반드시 this(), super()가 있어야 하는데** 이를 기술하지 않으면 묵시적으로 super()가 삽입되어 부모클래스의 기본 생성자를 자동으로 호출합니다.
- 생성자 내부에서 또다른 생성자를 호출할 때는(this(), super()) 반드시 생성자 블록 내부의 첫 라인에 기술해야 합니다.

# 실제 자바는 상속 계층 구조이다!

-자바의 모든 클래스는 **Object** 클래스를 상속받고 있습니다. Object클래스는 **자바의 최상위 클래스**입니다.



계층 구조는 **super()**를 통한 호출로 연결되어 있다!

# this와 super 예제

inheritance/this\_/Person.java

```
1: package inheritance.this_;
2:
3: public class Person {
4:
5:     public String name;
6:     public int age;
7:
8:     public Person(String name, int age) {
9:         this.name = name;
10:        this.age = age;
11:    }
12:    public Person(String name) {
13:        this(name, 1);
14:    }
15:    public Person() {
16:        this("이름없음", 1);
17:    }
18:
19:    public String getDetails() {
20:        return "이름: " + name + "\t나이: " + age;
21:    }
22: }
```

inheritance/super\_/Student.java

```
1: package inheritance.super_;
2:
3: public class Student extends Person{
4:
5:     private String studentId;
6:
7:     public Student(String name, int age, String studentId) {
8:         super(name, age);
9:         this.studentId = studentId;
10:    }
11:
12:    public String getDetails() {
13:        return super.getDetails() + "\t학번: " + studentId;
14:    }
15: }
```

# super예제

- super 키워드는 자식 클래스의 오버라이딩 된 메서드에서 부모의 메서드가 필요할 때 사용 할 수도 있습니다

```
public class Airplane {  
  
    void takeOff() {  
        System.out.println("비행기가 이륙합니다.");  
    }  
  
    void fly() {  
        System.out.println("일반 모드로 비행합니다.");  
    }  
  
    void land() {  
        System.out.println("비행기가 착륙합니다.");  
    }  
}
```

```
public class SuperSonicAp extends Airplane {  
  
    int flyMode = 1;  
  
    void fly() {  
        if(flyMode == 1) {  
            System.out.println("초음속 모드로 비행합니다.");  
        }else {  
            super.fly();  
        }  
    }  
}
```



# Chapter 12

## 수고하셨습니다