

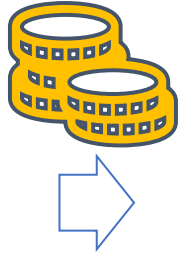
Java Fundamental

상속

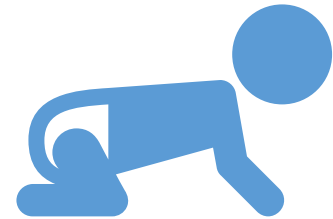
현실 세계의 상속



부모



자식
(부모)



자식

상속(Inheritance)

- 다른 클래스의 요소를 물려 받는 것
- 코드의 재사용 및 기능의 확장(추가)
- `extends` 키워드로 상속 관계 적용
- 상위 클래스(Superclass) = 부모 클래스 = 물려줌
- 하위 클래스(Subclass) = 자식 클래스 = 물려 받음

상속과 생성자

- 자식 클래스의 생성자는 따로 명시하지 않으면 자동으로 부모 클래스의 기본 생성자를 호출함

super 키워드와 this 키워드

- **super**: 부모 클래스를 가리키는 키워드
- **this**: 자기 자신을 가리키는 키워드
- 상속 관계에서 특정 클래스의 요소를 선택할 때 사용
- `super()`: 부모 클래스의 생성자를 호출
- `super()`는 반드시 생성자 블록에서 첫 번째로 작성

메서드 오버라이딩(Overriding)

- 상속받은 메서드를 재정의하는 것
- @Override 어노테이션으로 표시
 - Annotation: 데이터의 특징을 나타내기 위한 데이터
- 오버라이딩 메서드의 시그니처를 바꾸지 않음
 - Method signature: 메서드 정의 부분에서 메서드명과 매개변수 목록을 일컫는 말
- 메서드 호출 시 오버라이딩 된 메서드 우선 호출

오버로딩 vs 오버라이딩

정의 메서드명 매개변수 반환형	오버로딩	오버라이딩
	중복 정의	재정의
	동일	동일
	다름	동일
	무관	동일

클래스 다이어그램

```
public class Car {  
    private String color;  
    private int speed;  
  
    public String whatColor() {  
        return color;  
    }  
    public void accel(int speed) {  
        this.speed += speed;  
    }  
}
```

Car	
-	color :String
-	speed :int
+	whatColor() :String
+	accel(int) :void

다이어그램에서 상속 표현

```
public class Car {
```

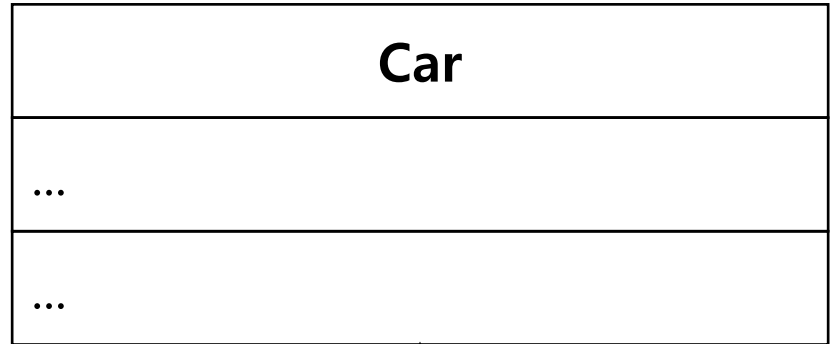
```
    ...
```

```
}
```

```
public class SportsCar extends Car{
```

```
    ...
```

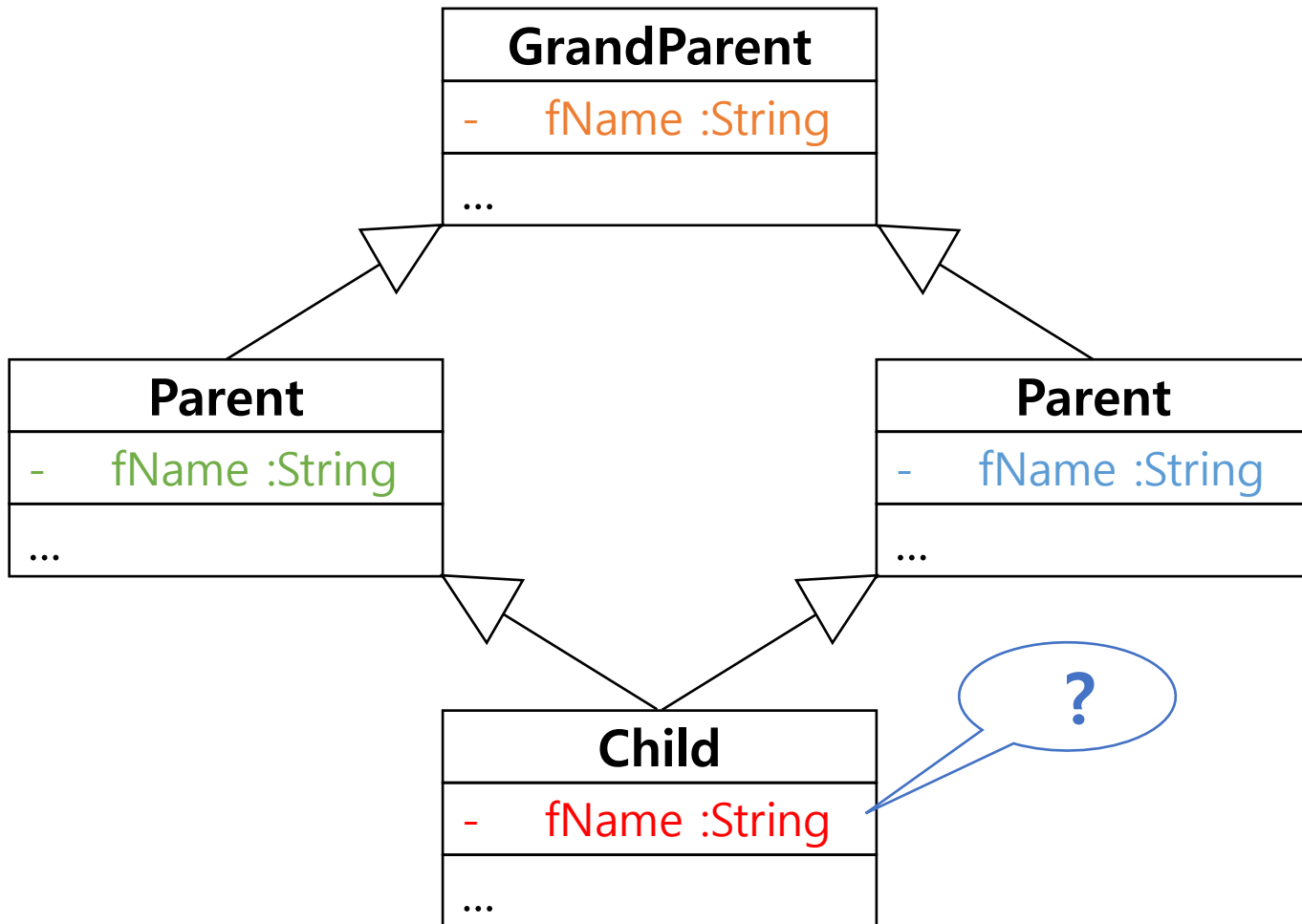
```
}
```



다중 상속

- 두 개 이상의 클래스와 상속 관계를 만드는 것
- 자바에서는 다중 상속을 지원하지 않음

다이아몬드 문제



instanceof 연산자

- 객체의 자료형을 검사하는 연산자
- 연산 결과 `true`, 혹은 `false` 반환

protected 접근 제한자

- 같은 패키지 내에 있는 클래스에 대해 접근을 허용
- 상속 관계에 있는 클래스에 대해 접근을 허용

Object 클래스

- 최상위 클래스
- 모든 클래스는 상속 관계를 설정하지 않아도 암시적으로 Object 클래스를 상속받음
- equals(), getClass(), toString() 등의 메서드를 갖고 있음

final 클래스, final 메서드

- final 클래스: 상속 불가
ex) String 클래스
- final 메서드: 오버라이드 불가
 - 주로 생성자에서 호출되는 메서드에서 사용