

05

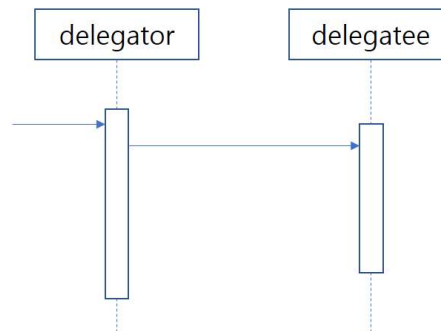
05-5. Delegate

다양한 기법

델리게이션

델리게이션 클래스

- 위임 패턴



```
class MyDelegatee {
    fun print(str: String){
        println("i am delegatee : $str")
    }
}
class MyDelegator {
    val delegatee: MyDelegatee = MyDelegatee()

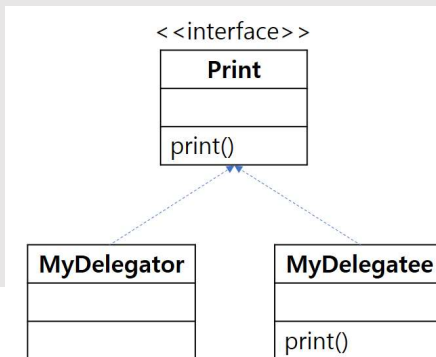
    fun print(str: String){
        delegatee.print(str)
    }
}
fun main(args: Array<String>) {
    val obj: MyDelegator = MyDelegator()
    obj.print("hello kkang")
}
```

델리게이션

델리게이션 클래스

- 코틀린에서는 위임을 by 예약어를 이용

```
interface Print {  
    fun print(arg: String)  
}  
  
class MyDelegatee: Print {  
    override fun print(arg: String) {  
        println("i am delegatee : $arg")  
    }  
}  
  
class MyDelegator(obj: MyDelegatee): Print by obj  
  
fun main(args: Array<String>) {  
    val obj: MyDelegatee = MyDelegatee()  
    MyDelegator(obj).print("hello kang")  
}
```



델리게이션

델리게이션 프로퍼티

- 프로퍼티의 위임도 by에 의해 제공
- 프로퍼티의 위임을 위한 클래스는 getValue() 와 setValue() 함수를 가지고 있어야 한다.
- 프로퍼티의 get() 에 의해 getValue() 함수가, set() 에 의해 setValue() 함수가 호출

```
class MySumDelegate {  
    var result: Int = 0  
    operator fun getValue(thisRef: Any?, property: KProperty<*>): Int {  
        println("getValue call... ref : $thisRef, property : '${property.name}'")  
        return result  
    }  
    operator fun setValue(thisRef: Any?, property: KProperty<*>, value: Int) {  
        result=0  
        println("setValue call... value : $value, '${property.name}'")  
        for(i in 1..value){  
            result += i  
        }  
    }  
}  
class Test {  
    var sum: Int by MySumDelegate()  
}
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare