

# Other Class Types



**Kevin Jones**

@kevinrjones

# Overview

**Enum classes**

**Sealed classes**

**Nested and inner classes**



# Sealed Classes

**Used to create restricted class hierarchies**

**Provide more control over inheritance**

**All subclasses are known at compile time**

**All subclasses in same module as sealed class**

**Similar to enums**

- But can have instances of a sealed class
- With its own state



```
class Outer {  
    class Nested {  
        val name: String = "Kevin";  
  
        fun doSomething() {  
            println(name)  
        }  
    }  
}  
  
val nested = Outer.Nested()  
  
nested.doSomething()
```

◀ Class definition nested inside Outer

◀ No access to outer data

◀ Inner scoped to Outer ‘namespace’



```
class Outer {  
    val name: String = "Kevin";  
    inner class Inner {  
        fun doSomething(){  
            println(name)  
        }  
    }  
}
```

```
val nested = Outer().Inner()  
nested.doSomething()
```

◀ Use ‘inner’ keyword

◀ Access outer classes data

◀ Inner create on instance of Outer



```
window.addMouseListener(  
    object : MouseAdapter() {  
        override fun mouseClicked(e:  
            MouseEvent) { ... }  
  
        override fun mouseEntered(e:  
            MouseEvent) { ... }  
    })
```

◀ **addMouseListener expects an instance of 'MouseAdapter'**

◀ **Anonymous inner class derives from MouseAdapter**

◀ **Implements the methods needed**



# Summary

## Enum classes

- Provide type-safe enumerations
- Are instance of the enumerated class type
  - Eg ‘Unknown’ was an instance of ‘ErrorCode’
- Have name and ordinal
- Can also provide a value



# Summary

## Sealed classes

- All subclasses derive from 'sealed class'
- All direct subclasses in the same module
- Like enums but can create instances
- Instances have state
- Used in 'when..is' to differentiate



# Summary

## Nested and inner classes

- Similar but
- Nested classes are scoped
- Inner classes created with instance of outer
- Inner classes have access to outer's data and methods
- Most often used for anonymous inner classes as event handlers



# Objects and Companions

---

