Companion Objects

Kevin Jones

@kevinrjones www.rocksolidknowledge.com

Static methods

Kotlin does not have static methods

However can have 'singletons'

Use 'object' keyword

Use 'companion object' to get 'statics'

object Keyword

Create a singleton

Actually defines a class and creates an instance

objects

Can have properties, methods, initializers

Cannot have constructors

'object' Keyword

```
object Meetings {
    var allMeetings = arrayListOf<Meeting>()
}

Meetings.allMeetings.add(Meeting(...))
```

Demo

Using the 'object' keyword

objects

Can derive from other classes/interfaces

Can be used where any 'instance' is used

'object' Inheritance

```
object CaseInsensitiveComparator : Comparator<File> {
    override fun compare(...) : Int {}
}
```

objects

Can be declared inside another class

Nested 'object'

```
class Person {
    object CaseInsensitiveComparator : Comparator<File> {
       override fun compare(...) : Int {}
    }
}
```

Kotlin Classes

Do not have static members

Companion Objects

Used for

Factory objects and static members

'static' Methods

```
class Person {
    companion object {
       fun main(args:Array<String>) {...}
    }
}
```

'factory' Methods

```
class Student {
    companion object {
       fun createUndergrad(name:String):Undergraduate{...}
       fun createPostgrad(name:String):Postgraduate{...}
    }
}
```

Companion Object Inheritance

```
class Student {
    companion object : Factory <Student> {
       fun createUndergrad(name:String):Undergraduate{...}
       fun createPostgrad(name:String):Postgraduate{...}
    }
}
```

Summary

'object' defines a standard class

- Can implement interfaces
- Can derive from classes

'object' Lets us create singletons

'companion object'

- 'Static' methods
- Factory methods