Higher Order Functions

Kevin Jones

@kevinrjones

Overview

Declaring lambdas

Using lambdas

Strategy pattern as an example

Lambdas in the standard library

- with and apply

What Are Higher Order Functions?

Functions as first class citizens

Can pass to and return from functions

Can store in collections

Strategy Pattern

Allows an algorithms behaviour to be selected at runtime

In OO the strategy patterns is often implemented using an Interface

But can just pass a function

Declaring Functions

Functions have a type

- And can be declared

Type consists of parameters and return type

- Parameter types declared in brackets
- Followed by ‘arrow’ (->)
- Followed by return type

```
(Int) -> Unit

(Int, String) -> Unit

(String) -> Int
```

- Function that takes an Int and returns Unit
- Function takes two parameters, Int and String
- Function that takes a String and returns an Int

Passing Functions

Declare a function as having another function as a parameter

Then call passing a function argument

Functions as Arguments

```
fun foo(bar: Int, action: (Int) -> Unit) {}

fun someFunction(num: Int) {}


foo(23, { a -> someFunction(a) })

foo(23) { a -> someFunction(a) }

foo(23, { someFunction(it) })

foo(23) { someFunction(it) }
```

Functions as Arguments

```
fun foo(bar: Int, action: (Int) -> Unit) {}

foo(23, {a -> /*do something here*/ })

foo(23) {a -> /* do something here */}
```

Functions as Arguments

```
fun foo(bar: Int, action: (Int) -> Unit) {}

foo(23, {/* do something with ‘it’ here*/ })

foo(23) {/* do something with ‘it’ here */}
```

Function References

```
fun foo(bar: Int, action: (Int) -> Unit) {}

class Bar {
  fun someOtherFunction() {}
}

fun someFunction(num: Int) {}

foo(23, ::someFunction)

val b = Bar()

foo(23, b::someOtherFunction)
```

Closures

Can mutate values outside the lambda

Unlike Java

Summary

Declaring lambdas

- (Type1, Type2) -> ReturnType

Using lambdas

- Can pass them into functions

Strategy pattern as an example

- Used a Fibonacci function
- Pass right type of lambda to the function

Lambdas in the standard library

- with and apply
- use, equivalent to ‘try with resources’

Collections and Sequences