YALOVA UNIVERSITY

HUAWEI

Android Programming with Huawei Mobile Services

Berk Ozyurt
Mehmet Yozgatli
Cengiz Toru

W3:
KOTLIN BASICS & OOP

- **Kotlin Basics**
  1. Functions
  2. Classes
- **OOP in Kotlin**

Kotlin Basics

Kotlin Basics - Functions

A function is a program that we have written, broken down into small parts.

Functions, which are a structure that saves from writing code repeatedly, are used in many software languages (almost all).

The fun keyword is used to define a function in Kotlin. Then the function name is written, the parenthesis is opened and the parameters it will take are written.

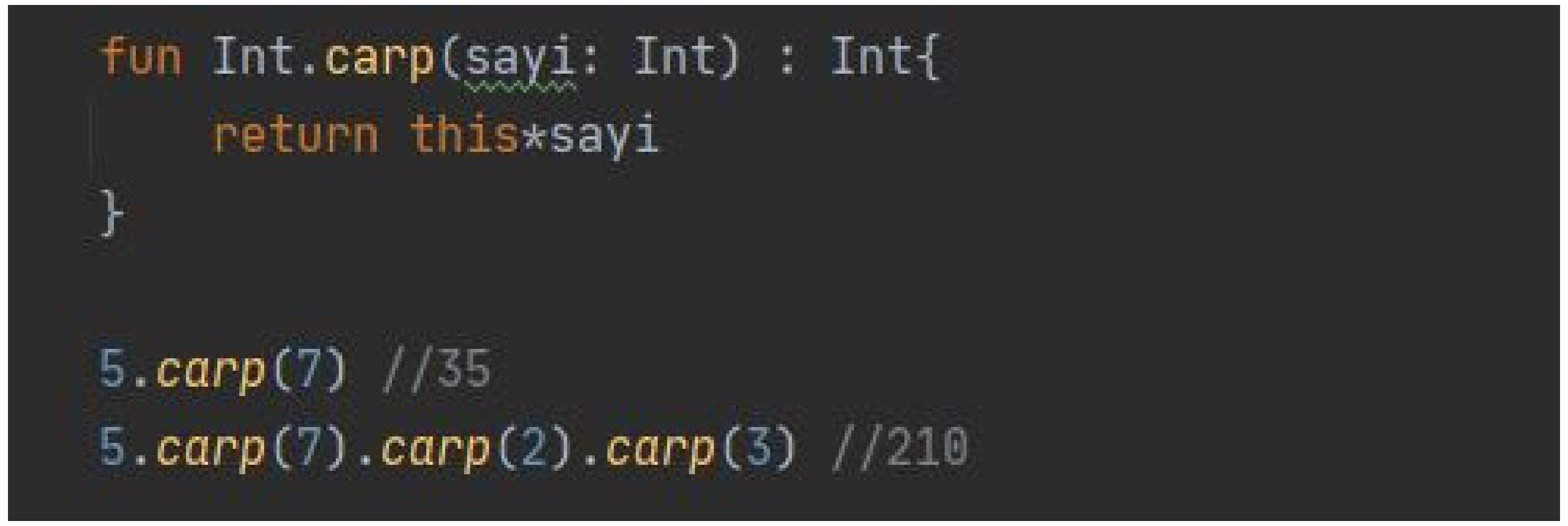
If there is no parameter, the parentheses will remain empty. Then the colon (:) is put on top of each other and the return type is written.

```
fun double(x: Int): Int {
    return 2 * x
}
```

Kotlin Basics - Extension

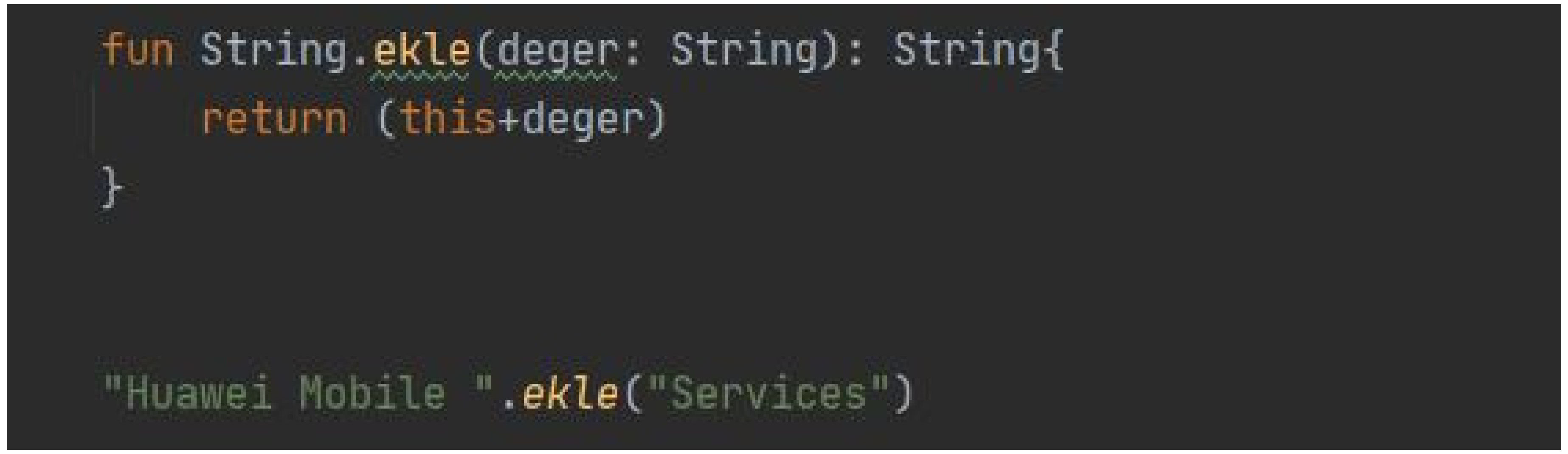
Extension function is a function for extending the functions of classes without having to touch their code.

Extension functions in Kotlin help extend the functionality of a class by adding new functions.



```kotlin
fun Int.carp(sayi: Int) : Int{
    return this*sayi
}

5.carp(7) //35
5.carp(7).carp(2).carp(3) //210
```



```kotlin
fun String.ekle(deger: String): String{
    return (this+deger)
}


"Huawei Mobile ".ekle("Services")
```

Kotlin Basics - Classes

Classes in Kotlin are declared using the keyword class:

```
class Person { /*...*/ }
```

The class declaration consists of the class name, the class header (specifying its type parameters, the primary constructor, and some other things), and the class body surrounded by curly braces. Both the header and the body are optional; if the class has no body, the curly braces can be omitted.

Kotlin Basics - Classes

Constructors

A class in Kotlin can have a primary constructor and one or more secondary constructors. The primary constructor is a part of the class header, and it goes after the class name and optional type parameters.

If the primary constructor does not have any annotations or visibility modifiers, the constructor keyword can be omitted

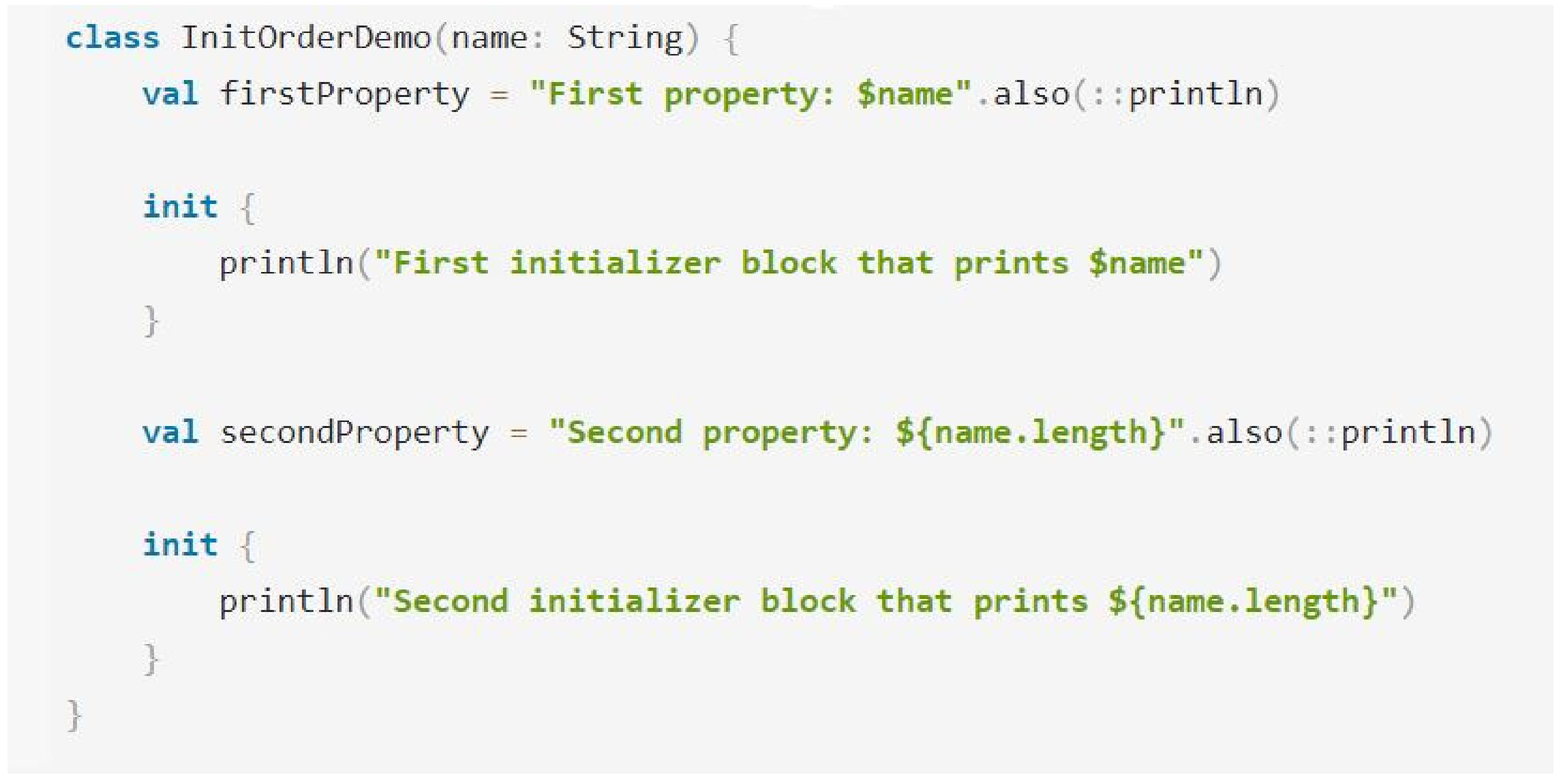
```
class Person constructor(firstName: String) { /*...*/ }
```

Kotlin Basics - Classes

Init

The primary constructor cannot contain any code. Initialization code can be placed in initializer blocks prefixed with the init keyword.

During the initialization of an instance, the initializer blocks are executed in the same order as they appear in the class body, interleaved with the property initializers:



```
class InitOrderDemo(name: String) {
    val firstProperty = "First property: $name".also(::println)

    init {
        println("First initializer block that prints $name")
    }

    val secondProperty = "Second property: ${name.length}".also(::println)

    init {
        println("Second initializer block that prints ${name.length}")
    }
}
```

Kotlin Basics - Classes

Data classes

It is not unusual to create classes whose main purpose is to hold data. In such classes, some standard functionality and some utility functions are often mechanically derivable from the data. In Kotlin, these are called data classes and are marked with data:

```kotlin
data class User(val name: String, val age: Int)
```

```kotlin
data class Person(val name: String) {
    var age: Int = 0
}
```

```kotlin
val person1 = Person("John")
val person2 = Person("John")
person1.age = 10
person2.age = 20
```

Kotlin Basics - Classes

Enum classes

The most basic use case for enum classes is the implementation of type-safe enums:

```kotlin
enum class Direction {
    NORTH, SOUTH, WEST, EAST
}
```

Each enum constant is an object. Enum constants are separated by commas.
Since each enum is an instance of the enum class, it can be initialized as:

```kotlin
enum class Color(val rgb: Int) {
    RED(0xFF0000),
    GREEN(0x00FF00),
    BLUE(0x0000FF)
}
```

OOP in Kotlin

OOP in Kotlin - What is OOP?

OOP

Object Oriented Programming

Object-oriented programming has been the longest used structure in the software community since 1960. Java, C#, C++, Python, PHP and many languages use object-oriented programming. Kotlin programming language is an object-oriented language.

The general purpose in object-oriented programming is to destroy the structure that causes confusion while writing code. A hierarchy is provided between the codes written with the class structure.

OOP in Kotlin - What is OOP?

OOP

Benefits

- It ensures the code-reusability of the code we have written.
- It gives us extensibility as we can add properties and methods to a class we write.
- Since the code written in Procedural Programming time is a little easier to turn into spaghetti code, the maintenance of projects has become easier thanks to OOP.
- It provides sustainability. In this way, when the codes are maintained, we can update our codes by updating a small number of classes without having to play on many functions.
- It becomes easier for us to save time (timely). Thanks to OOP, more than one developer can easily work on more than one piece of code.

OOP in Kotlin - What is OOP?

CLASS

These are the fields where we define the variables and methods that are object properties. Class is used in languages that support object-oriented programming. A class is not an object. Classes do not consist of objects. Class and objects work interdependently, but they are different structures from each other.

OBJECT

Objects are more physical structures than classes. When we compare the classes to the house plan, as we mentioned in the example above, we can compare the objects to the houses built by adhering to the plan. We can generate one or more objects from within a class.

OOP in Kotlin - What is OOP?

THIS

The "this" keyword is used in the programming language Kotlin as well as in Java, C# languages. It is a reference variable. It is used to refer to an existing object. Although it is used in the class, it refers to the object created after the object is created.

PROPERTIES

It is a concept that has a relationship with class and object. It is used to specify properties of objects. The following elements will be added when the class is created:

- Variable
- Constant
- Method
- Class defined inside the class

OOP in Kotlin - What is OOP?

METHODS

They are the elements that perform the tasks specified in the application. The fun keyword is used to define a method. Methods are used to specify the functions of objects.

INTERFACE

They are the concepts that we define the properties and functions of objects as lists. In interfaces, only the names of the objects should be written when the properties and functions are to be given to the objects.

OOP in Kotlin - What is OOP?

INHERITANCE

Inheritance, which is inspired by the inheritance structure found in real life, can transfer its properties to other objects through inheritance. The Object class used in Android programming is the ancestor of all classes. Other classes are derived from this Object class. In Kotlin, Any class is the parent class. All other classes are derived from it.
You can avoid code duplication by using inheritance. You write shorter and cleaner code.

ABSTRACT

It is the overriding of the inherited features within the derived classes to gain different features. We can give different functions to the same method in different classes.

OOP in Kotlin - What is OOP?

ENCAPSULATION

In short, we can say that encapsulation is used to limit external interference to the object we have created. We use the prefix private at the beginning of the property that we define for the class, so that private properties of this class cannot be accessed from the outside.

POLYMORPHISM

In short, Polymorphism is the ability to perform different operations using the same name. Polymorphism is divided into 2 in itself. Performing different operations with the same name in the same class is called static polymorphism. Performing different operations with the same name in different classes is called dynamic polymorphism.

HUAWEI

LET'S TALK
CONTACT INFORMATION

Berk Ozyurt
berk.ozyurt1@huawei.com

Mehmet Yozgatli
mehmet.yozgatli1@huawei.com

Cengiz Toru
cengiz.toru@huawei.com

Telegram Channel
will be created

Official Website
https://developer.huawei.com