

# Kotlin Sealed Class – Example

## Kotlin Sealed Class

Kotlin Sealed Class represent restricted class hierarchies, where a value can have a type from a restricted set.

### Sealed Class Example

Following is an example of sealed class and sub-classes that extent sealed class.

```
sealed class ArithmeticOperation

class Add(var a: Int, var b: Int): ArithmeticOperation()
class Subtract(var a: Int, var b: Int): ArithmeticOperation()
class Multiply(var a: Int, var b: Int): ArithmeticOperation()
class Divide(var a: Int, var b: Int): ArithmeticOperation()
```

A value of type `ArithmeticOperation` is restricted to `Add`, `Subtract`, `Multiply` and `Divide`.

This situation may look similar to that of [Kotlin Enum](#). But the difference between Enum and Sealed Classes is that, in Enum there is only one value for each class type, whereas in Sealed Class, there could be multiple object instances for a class type (whose parent is a sealed class).

### About Sealed Classes

- A Sealed Class have specific number of sub-classes.
- Location
  - All the classes that directly inherit a sealed class should be present in the same Kotlin file.
- Constructors
  - Sealed Class cannot have non-private constructors.
  - Private is the default access modifier for constructors in Sealed Class.
- Abstract
  - By default, a Sealed Class is abstract.
  - Hence, a Sealed Class cannot be instantiated and all other properties pertaining to Abstract Class hold for a Sealed Class.

### Example – Kotlin Sealed Class with when expression

In the following example, a Sealed class with four sub-classes is considered and when expression is used for demonstrating the usage of Sealed Class.

```
fun main(args:
Array<String>){
```

```
fun main(args: Array<String>) {
    var operation = Add(4, 3)

    var result = execute(operation)

    print("Result : "+result)
}

sealed class ArithmeticOperation

class Add(var a: Int, var b: Int): ArithmeticOperation()
class Subtract(var a: Int, var b: Int): ArithmeticOperation()
class Multiply(var a: Int, var b: Int): ArithmeticOperation()
class Divide(var a: Int, var b: Int): ArithmeticOperation()

fun execute(op: ArithmeticOperation) = when (op) {
    is Add -> op.a + op.b
    is Subtract -> op.a - op.b
    is Multiply -> op.a * op.b
    is Divide -> op.a / op.b
}
```

This example could be considered a special case, where [Kotlin When](#) is used as an expression. In the `execute()` function, Kotlin Compiler makes sure that all the classes from the allowed sub-class set are addressed.

## Kotlin Java

- [Kotlin Tutorial](#)

## Getting Started

- [Setup Kotlin\(Java\) Project](#)

- [Kotlin Example Program](#)

- [Convert Java to Kotlin](#)

- [Kotlin Main Function](#)

- [Kotlin Loops](#)

- [Kotlin For Loop](#)

- [Kotlin While, Do While Loops](#)

- [Kotlin Repeat](#)

- [Kotlin Ranges](#)

- [Kotlin When](#)

↳ Kotlin when

## Object Oriented Concepts

### Classes

↳ Kotlin - Class, Primary and Secondary Constructors

↳ Kotlin Sealed Class

↳ Kotlin Data Class

↳ Kotlin Enum

↳ Kotlin - Extension Functions

### Inheritance

↳ Kotlin Inheritance

↳ Kotlin Override Method of Super Class

### Abstraction

↳ Kotlin Abstraction

↳ Kotlin Abstract Class

↳ Kotlin - Interfaces

↳ Kotlin Null Safety

### Exception Handling

↳ Kotlin Try Catch

↳ Kotlin Throw Exception

↳ Kotlin Custom Exception

### Fix Compilation Errors

↳ Kotlin - Variable must be initialized

↳ Kotlin - Primary Constructor call expected

↳ Kotlin - Null can not be a value of a non-null type String

↳ Kotlin - Cannot create an instance of an abstract class

### Kotlin - String Operations

↳ Kotlin - Compare Strings

↳ Kotlin - Replace String

↳ Kotlin - Split String

↳ Kotlin - Split String to Lines

↳ Kotlin - String Capitalize

### Kotlin - Functions

↳ Kotlin Function - Default Arguments

↳ Kotlin - Use Function

## Kotlin Collections

### Kotlin List

▸ [Kotlin List](#)

▸ [Kotlin List forEach](#)

## Kotlin File Operations

▸ [Kotlin - Create File](#)

▸ [Kotlin - Read File](#)

▸ [Kotlin - Read File as List of Lines](#)

▸ [Kotlin - Write to File](#)

▸ [Kotlin - Append Text to File](#)

▸ [Kotlin - Check if File Exists](#)

▸ [Kotlin - Copy a File to Other](#)

▸ [Kotlin - Iterate through all files in a directory](#)

▸ [Kotlin - Delete Recursively](#)

▸ [Kotlin - Get File Extension](#)

## Kotlin Interview Q/A

▸ [Kotlin Interview Questions](#)

## Kotlin Android

▸ [Kotlin Android Tutorial](#)

## Useful Resources

▸ [How to Learn Programming](#)