



Amit Shekhar

22 May 2017

Learn Kotlin — Sealed Classes



What are Sealed Classes?

From the documentation: Sealed classes are used for representing restricted class hierarchies, when a value can have one of the types from a limited set, but cannot have any other type. They are, in a sense, an extension of enum classes: the set of values for an enum type is also restricted, but each enum constant exists only as a single instance, whereas a subclass of a sealed class can have multiple instances which can contain state.

How to declare sealed class?

```
sealed class Car {  
  
    data class Maruti(val speed: Int) : Car()  
    data class Bugatti(val speed: Int, val boost: Int) : Car()  
    object NotACar : Car()  
  
}
```

The key benefit of using sealed classes comes into play when you use them in a when expression. If it's possible to verify that the statement covers all cases, you don't need to add an else clause to the statement.

```
fun speed(car: Car): Int = when (car) {  
    is Car.Maruti -> car.speed  
    is Car.Bugatti -> car.speed + car.boost  
    Car.NotACar -> INVALID_SPEED  
    // else clause is not required as we've covered all the case  
}
```



So , whenever you get the situation like this, consider using sealed class.

Share and spread the knowledge



Facebook Comments Plugin

Mindorks Updates



Correctify App

Janishar Ali

English editing and learning app, a product of Mindorks.

Android Interview Questions

Amit Shekhar

Your cheat sheet for Android interview.

Better yourself

Dalai Lama

The goal is not to be better than the other man, but your previous self.

© 2017 Copyright

Mindorks Nextgen Private Limited

GitHub

[MindorksOpenSource](#)

[Amit Shekhar](#)

[Janishar Ali](#)



Connect With Mindorks

[Facebook](#)

[Linkedin](#)

[Medium](#)



Mindorks Community

[Get Community Invite](#)

[Slack Community](#)

[Subscribe](#)

