

Aprende más sobre cómo programar aplicaciones con Kotlin

 jotajotavm.com/android

OPERADORES LÓGICOS

El comportamiento de los operadores AND y OR son muy diferentes.

Con el siguiente ejemplo espero que lo recuerdes con facilidad

El operador OR



El operador **OR** se comporta igual que uno de esos carros del supermercado al que debes meter una moneda para poder usarlo. Acepta cualquiera de las monedas que te indica. No exige una moneda específica. Con cualquier de las que te indica es suficiente. Tampoco te obliga a meter una moneda de cada opción. **Con una es suficiente.** Si tu dispones de todas las opciones...ok. Pero no es necesario

En la figura ves que el carro acepta monedas de 1€ o de 0,5€. Esa es la clave. Apliquemos esto al código

El operador OR da por buena cualquier combinación de condiciones simplemente con que una de ellas sea cierta, pero al menos tiene que haber una que sí sea cierta

Si pudiera hablar, el operador OR diría esto:

- Tienes una moneda de 1€?? Ok, adelante
- Tienes una moneda de 0,5€? Ok, adelante
- Tienes una moneda de 1€ y también tienes otra de 0,5€? Ok, adelante
- No tienes ninguna?? en serio?? Ninguna?? mmm entonces lo siento, pero... NO puedes acceder al carro

En código esto sería

CondiciónFalsa OR CondiciónVerdadera = OK
 CondiciónVerdadera OR CondiciónFalsa = OK
 CondiciónVerdadera OR CondiciónVerdadera = OK
 CondiciónFalsa OR CondiciónFalsa = NO

False || True = True
 True || False = True
 True || True = True
 False || False = False

El operador AND



El operador **AND** se comporta como un policía que no te deja pasar ni una. ¿Llevas todo perfecto excepto una sola cosa? Multa. ¿Circulabas con el carnet, sin hablar por teléfono, con el cinturón puesto, sin beber alcohol, pero había una luz rota? Multa **Tienes que cumplir TODAS las condiciones para que todo vaya bien**





En código esto sería

CondiciónFalsa AND CondiciónVerdadera = NO
 CondiciónVerdadera AND CondiciónFalsa = NO
 CondiciónFalsa AND CondiciónFalsa = NO
 CondiciónVerdadera AND CondiciónVerdadera = OK

Es decir:

False && True = False
 True && False = False
 False && False = False
 True && True = True

Cuándo cada uno de estos operadores devuelve un OK, es decir, un true, accederíamos al código del if, for, while, o cualquier otra estructura de control en la que se esta estableciendo la condición. En caso contrario tal bloque de código no se ejecuta. Por eso estos operadores son tan importantes porque influyen en qué vaya a hacer o no la aplicación que se programe

 jotajotavm
 @JoseCodFacilito
 jotajotavm
 jotajotavm