



JOSUÉ LÓPEZ

16 de mayo de 2021

⋮

¿Cuál es la diferencia al programar Android con Java y con Kotlin?

Como bien conocemos, el rumbo de la tecnología va creciendo y con ello su influencia bajo los requerimientos que los programadores deben manejar en las diferentes áreas, a ello se le suma el tema de la programación de aplicaciones móviles, mismo que ya de por sí es muy sonado en la actualidad y es acá donde converge la pregunta que plantea este caso, ya que prácticamente la hemos de haber planteado nosotros mismos al momento de pasar de programar con Java a iniciar con Kotlin y todo esto mientras desarrollamos nuestras horas de clases. De una forma u otra es necesario que recapitemos un poco acerca de la historia de ambos lenguajes, para que de esta forma podamos describir sus beneficios y diferencias como tal...

Primeramente, tenemos a Java un lenguaje desarrollado por James Gosling y lanzado al mercado en 1995 por la compañía Sun Microsystems, que años después fue adquirida por la empresa Oracle. Si estudiamos sus inicios, podemos notar que fue creado con la premisa de ser un lenguaje de programación cuya estructura fuese sencilla y que a la vez se le permitiese su ejecución en diversos sistemas operativos, entonces, partiendo de ello se llegó a un lenguaje que sería parte de la gamma de los lenguajes orientados a objetos y que a la vez proporcionaría un tipado robusto, siendo catapultado como una gran "plataforma" para el mundo informático...

Por otro lado, tenemos a Kotlin, empezando como un proyecto de la empresa JetBrains en 2011, mismo en el que su código no fue liberado sino hasta 2012 bajo la licencia de Apache 2 y prácticamente su inicio se ve influenciado por Java, ya que nace a partir de la idea de superar a este lenguaje como tal, pero a la vez, sin dejar de operar con el código de este y por lo tanto se le considera un lenguaje orientado a objetos, pero de una calidad más "industrial".

Ahora que ya conocemos parte de sus fundamentos, podemos dar paso a mencionar los beneficios y diferencias que trae consigo la implementación de cada lenguaje como tal:

Java:

1. Puede ser ejecutado en cualquier hardware, lo cual lo convierte en portable...
2. Es un lenguaje orientado a objetos.
3. Es multi-plataforma y cuenta con altos niveles de seguridad.
4. Permite la utilización de funciones y clases, por lo tanto, soporta el uso de operadores lógicos, bloques, expresiones y sentencias.
5. Su curva de aprendizaje es rápida.
6. Posee una amplia biblioteca y una gran comunidad activa en la web, siendo este punto una de las grandes ventajas para los programadores en general.
7. Con Java podemos crear dos tipos de programas como lo son los applets y las aplicaciones independientes.
8. Con Java podemos llevar a cabo dentro del mismo programa de forma simultánea una serie de tareas, haciendo que sea multihilo.

Kotlin:

1. Es considerado como el lenguaje "oficial" para el desarrollo de aplicaciones Android, siendo este a su vez orientado a objetos.
2. La extensión de Kotlin para Android, permite realizar el "import" a una referencia del View en el Activity, por lo tanto, mejora en el uso del findViewById.
3. Kotlin nos permite heredar una clase con nueva funcionalidad, lo cual es ideal si hay una clase a la que le falta un método
4. Kotlin no tiene excepciones del tipo checked, entonces no se requiere capturar o declarar ninguna excepción.
5. Kotlin puede realizar el manejo de los Cast redundantes de forma autónoma...
6. Kotlin no soporta conversiones implícitas.
7. Es intercambiable con Java, entonces, es posible tener clases Kotlin y clases Java trabajando juntas.
8. Tiene una muy sencilla curva de aprendizaje, ya que esta diseñado con el objetivo de ser intuitivo y sencillo de leer.
9. Tiene un soporte de primera para Android Studio.
10. Su código es más conciso y compacto.
11. Provee una fácil migración y un bajo riesgo al llevarla a cabo.
12. Kotlin elimina los NPE (Null Pointer Exception) del código, es por ello que no es posible asignar un valor nulo a una variable independientemente de su tipo.
13. Nos da la opción de importar desde el diseño todas las referencias a las vistas.
14. Para aquellos sistemas que manejen la API RESTfull y analicen objetos de tipo JSON podemos almacenar los valores de las propiedades dentro de un mapa.
15. Es posible trabajar con frameworks o librerías JQuery, React, Angular, entre otros.
16. Trabaja con funciones anónimas y funciones de orden superior.
17. Permite el desarrollo multiplataforma, por lo cual es interoperable.