

&gt; programmez!

Recherche

[Se connecter](#) | [S'inscrire](#)
[Actualités](#)   [Avis d'experts](#)   [Boutique](#)   [Emploi](#)   [Agenda](#)   [Meetup #53 : Platform Engineering](#)   [DevCon #25 : informatique quantique](#)


## JetBrains veut créer un nouveau langage avec une (très) haute abstraction

Par: francoistonic | jeu, 21/08/2025 - 07:23

[Jetbrains \(/actualites/jetbrains\)](#)

Partager 0

Post

Partager

JetBrains, créateur de Kotlin, travaille sur un nouveau langage, pour le moment sans nom, avec un haut niveau d'abstraction. Le but : favoriser le développement multiplateforme et pouvoir faire de la génération de code avec l'IA de meilleure qualité et plus facile à maîtriser par l'IA.

C'est le patron de l'éditeur qui a dévoilé le projet dans une interview à InfoWorld. Il a rappelé qu'au départ, nous codions en assembleur puis les langages de plus haut de niveau sont apparus tels que C, C++ puis des langages toujours plus haut, avec Java et C#. Maintenant, il s'agit d'aller plus loin dans l'abstraction. Il estime que les langages actuels sont complexes, avec des paradigmes parfois très difficiles à maîtriser.

Ce nouveau langage serait un dérivé de Kotlin. Ce langage sera sans doute en Anglais. « Ainsi, au lieu d'écrire trois applications, vous les écrivez dans un langage de programmation spécial, qui est essentiellement l'anglais, qui décrit comment vous voulez voir cette application d'une manière très spécifique, puis les agents d'IA, avec les outils JetBrains, généreront le code de toutes ces plateformes », a déclaré Skrygan. Divers projets sur une génération de code depuis un texte en langage naturel ont été lancés depuis 15 ans. Il existe aussi des langages spécifiques comme les DSL.

Aucun détail ni aucune roadmap n'ont pas donné.

**Notre avis :** Aujourd'hui, nous avons des langages avec tous les niveaux d'abstraction, du natif, de l'interprété, etc. Nous connaissons les problèmes avec les langages avec un haut niveau d'abstraction. Le risque est d'introduire une nouvelle lourdeur pour son exécution notamment via un JIT. Le constat que fait Skygan n'est pas faux notamment quand le code est généré par de l'IA. Faut-il pour autant réinventer la roue ? Car finalement, ce langage sans nom réinventera Kotlin sans doute en supprimant telle complexité, telle paradigme ou en introduisant une surcouche pour masquer la "plomberie".

Partager 0

Post

Partager