Présentation de Kotlin

Kotlin

* Historique

JetBrains est un éditeur tchèque d’outils pour développeurs et plus spécifiquement dédié aux langages HTML, CSS, JavaScript, PHP, Java, C#, Python et Ruby. Ces outils sont réputés fiables, pratiques et d’une grande qualité. Son IDE le plus connu est IntelliJ IDEA.

En 2010, JetBrains décide de créer un langage s’approchant du C#, mais qui reste basé sur la JVM (machine virtuelle Java) et pouvant pallier les lacunes de Java. Le projet Kotlin naît. Kotlin est le nom d’une île (comme Java) près de Saint-Pétersbourg. Cette île contient de nombreuses sociétés modernes (ainsi que des bureaux JetBrains) et a une histoire révolutionnaire. Cela reflète bien le langage qu’est Kotlin : en lien avec Java, moderne et révolutionnaire.

La raison d’être de Kotlin est de faciliter la vie des développeurs. Tout ce qui peut être développé en Java peut l’être aussi avec Kotlin.

Voici les dates clés :

* Création de la société JetBrains en 2000.
* Naissance du langage Kotlin en 2010.
* Présentation de Kotlin à la conférence JVM Langage Summit en 2011.
* Entre 2011 et 2017 : fort engouement pour Kotlin.
* Android Studio (basé sur IntelliJ IDEA) devient l’IDE officiel pour les développements Android en 2014.
* Officialisation du support de Kotlin pour la plateforme Android lors de la célèbre conférence du Google i/o 2017.
* Début 2018 : 20 % des projets Android sont développés en Kotlin.

Le futur semble bien prometteur pour ce nouveau langage.

* Philosophie

Jetbrains a développé Kotlin avec une certaine philosophie, voici les différents points qui la définissent :

* Pragmatique : désigne la capacité à s’adapter à la réalité. C’est ce qu’a fait JetBrains en créant Kotlin : grâce à l’expérience qu’a acquise JetBrains dans le développement d’IDE, ils ont imaginé un langage pratique répondant aux besoins réels des développeurs.
* Concis : le langage Kotlin s’efforce de rendre implicites nombre de fonctionnalités afin que le code ait du sens et soit le plus explicite possible, et ainsi réduire au mieux la dette technique des projets.
* Robuste : le langage Kotlin s’efforce d’augmenter la sécurité du code afin que le nombre d’erreurs soit le plus petit possible. Par exemple, de nombreux mécanismes ont été mis en place afin d’éviter au maximum les NullPointerException.
* Interopérable : avoir un nouveau langage c’est bien, mais cela risque de nous freiner si l’on doit tout redévelopper. Avec Kotlin, vous n’avez pas à vous en faire, car il est totalement interopérable. C’est-à-dire que du code Java peut totalement interagir avec du Kotlin et inversement.
* Avantages

Voici les principaux avantages de ce langage :

* Moins verbeux : le nombre de lignes de code d’un programme Kotlin est nettement plus faible qu’un même programme développé en Java.
* Plus explicite : moins de lignes de code avec un code qui s’autocommente davantage, c’est-à-dire que le rôle des instructions est plus clair.
* Pratique : moins verbeux, moins explicite, donc plus pratique.
* Moins d’erreurs avec un système de gestion des NullPointerException séduisant.

En résumé, il est plus productif.

* Environnement de développement
* Présentation

L’utilisation d’IntelliJ IDEA est idéale pour découvrir Kotlin. Cet IDE est complet et innovant et c’est surtout l’IDE venant des créateurs de Kotlin. De plus, Android Studio est basé sur IntelliJ IDEA, une façon de ne pas être perturbé lors du passage sur Android Studio. Une version gratuite sur 30 jours est disponible sur le site internet de JetBrains ce qui donne suffisamment de temps pour découvrir Kotlin.

Il est donc conseillé d’aller télécharger l’IDE IntelliJ IDEA pour tester les exemples présentés dans ce livre, à l’adresse suivante :   
https://www.jetbrains.com/idea/

* Premiers pas avec IntelliJ IDEA

Pour tester Kotlin, il est nécessaire de créer un nouveau projet de type Kotlin/JVM :



Une fois le projet créé, l'interface suivante apparaît. Pour tester vos programmes, il est nécessaire de :

* savoir afficher l'explorateur de projet : View - Tool Windows - Project,
* exécuter le programme : Run - Run.



* Premier exemple Kotlin

L'exemple de ce chapitre démontre l'efficacité et la simplicité de Kotlin.

Ci-dessous, une classe Personne (2 lignes) écrite avec Kotlin.

Classe Personne écrite avec Kotlin

import java.util.\*   
   
data class Personne(var id:Int, var nom:String, var   
prenom:String,var dateDeNaissance:Date)

L’exemple ci-dessous est une classe Personne (53 lignes) écrite avec Java. Cette classe ne fait absolument rien de plus que l’exemple précédent.

Classe Personne écrite avec Java

import java.util.Date;   
   
public class Personne {   
    public int id;   
    public String nom;   
    public String prenom;   
    public Date dateDeNaissance;   
   
    public Personne() {   
    }   
   
    public int getId() {   
        return id;   
    }   
   
    public void setId(int id) {   
        this.id = id;   
    }   
   
    public String getNom() {   
        return nom;   
    }   
   
    public void setNom(String nom) {   
        this.nom = nom;   
    }   
   
    public String getPrenom() {   
        return prenom;   
    }   
   
    public void setPrenom(String prenom) {   
        this.prenom = prenom;   
    }   
   
    public Date getDateDeNaissance() {   
        return dateDeNaissance;   
    }   
   
    public void setDateDeNaissance(Date dateDeNaissance) {   
        this.dateDeNaissance = dateDeNaissance;   
    }   
   
    @Override   
    public String toString() {   
        return "Personne{" +   
                "id=" + id +   
                ", nom='" + nom + '\'' +   
                ", prenom='" + prenom + '\'' +   
                ", dateDeNaissance=" + dateDeNaissance +   
                '}';   
    }   
}

Cet exemple permet d’imaginer ce que Kotlin peut offrir aux développements Android…