

De werking van Kotlin Native

Onderzoeksvoorstel Bachelorproef

Van Wassenhove Ilias

Samenvatting

Nog niet zo lang geleden had men bij het bouwen van mobiele applicaties (Android, iOS en Windows) enkel en alleen de mogelijkheid om drie aparte applicaties te bouwen, native applicaties. Dit vroeg zeer veel werk en was heel kostelijk voor veel bedrijven. Als reactie hierop zijn de hybride frameworks uitgevonden, denk maar aan: React, Xamarin en Angular. Bij deze frameworks moest er maar één maal code geschreven worden en het framework zorgde er voor dat je applicatie op elk besturingssysteem kon draaien. We mogen namelijk nog een framework toevoegen aan het lijstje, namelijk Kotlin Native.

Kotlin is een nieuwe programmeertaal die geïntroduceerd werd in 2011 door JetBrains. Origineel is het een programmeertaal die draait op de Java Virtual Machine (JVM). Maar JetBrains is veel verder gegaan dan toestellen die een JVM kunnen draaien. Met Kotlin Native hebben ze zich met Kotlin gericht tot alle platformen en besturingssystemen. In dit onderzoek zal onderzocht worden hoe Kotlin Native werkt. Dit is belangrijk omdat Kotlin Native nog heel jong is en er zijn al veel ontwikkelaars die reeds Kotlin gebruiken en staan te springen om Kotlin Native te gebruiken. De enige beperking die zij momenteel ervaring is documentatie. Van dit onderzoek wordt verwacht dat aan de hand van een aantal experimenten het duidelijk is hoe men Kotlin Native kan gebruiken. De conclusies die men uit dit onderzoek zal kunnen trekken is dat Kotlin Native zeker en vast geschikt is voor de ontwikkeling van web-, desktop- en mobiele applicaties. Voor de toekomst kan ik zeggen dat Kotlin en Kotlin Native enkel en alleen maar groter en populairder zal worden. Het is natuurlijk afhankelijk van de programmeur naar welk framework zijn voorkeur gaat. Google heeft reeds volledige ondersteuning aangeboden voor Kotlin en Android heeft met 85% het grootste marktaandeel van de mobiele applicaties.

Sleutelwoorden

Applicatieontwikkeling. Kotlin – JetBrains – Cross-platform – Google – Desktop – Mobile – Web – Kotlin Native

Contact: ilias.vanwassenhove.w9579@student.hogent.be

Inhoudsopgave

1	Introductie
2	State-of-the-art
3	Methodologie
4	Verwachte resultaten
5	Verwachte conclusies
	Referenties

1. Introductie

Maar wat betekent nu cross-platform? Met cross-platform bedoelt men een systeem/software dat op verschillende platformen en/of besturingssystemen kan draaien. Er zijn al heel wat cross-platform frameworks op de markt. Denk maar aan Xamarin, React Native, Angular in combinatie met het Ionic framework. Momenteel is JetBrains bezig met het ontwikkelen van Kotlin Native waarmee het zou mogelijk zijn om zowel voor web, dekstop als voor mobile (Android, iOS) te ontwikkelen.

Kotlin is voor vele programmeurs nieuw. Sinds dit jaar biedt Google volledige ondersteuning voor het gebruik van Kotlin bij Android applicatieontwikkeling. Maar het gaat nog veel verder. JetBrains, de ontwikkelaars van Kotlin, werkt momenteel aan Kotlin Native, het framework dat moet zorgen voor cross-platform ontwikkeling. Maar Kotlin Native is nog heel jong. De huidige versie is 0.6 en er bestaat nog maar weinig documentatie rond Kotlin Native. Het probleem echter bij dit framework is dus momenteel dat er heel weinig documentatie en programmeurs dus zelf moeten ontdekken hoe alles momenteel werkt. Dit kan via conferences, voorbeeldprojecten of youtube (kanaal van JetBrains). Maar wat zijn nu de mogelijkheden van Kotlin Native? Zijn er beperkingen? Hoe wordt de GUI opgebouwd? Hoe werkt LLVM, de compiler die alle code omzet naar machinetaal die op elk platform en/of besturingssysteem kan draaien en dus zonder het gebruik van de Java Virtual Machine?

2. State-of-the-art

Kotlin, de nieuwe programmeertaal. In 2011 voor het eerst aangekondigd door JetBrains. Zes jaar later biedt Google

volledige ondersteuning voor Android applicatieontwikkeling. De voorbije jaren was er nog maar weinig bekend rond Kotlin. Er waren zo goed als geen boeken of andere literatuur op de markt. Hier is het laatste jaar verandering in gekomen. Er worden steeds meer en meer boeken geplubliceerd.

Op het internet zijn er reeds enkele reviews, onderzoeken en boeken beschikbaar. Hierin wordt er verteld waarom men juist wel of niet voor Kotlin moet kiezen. Zo vindt je her en der ook wel eens een vergelijkende studie tussen Java en Kotlin, maar meestal zijn deze niet zo sterk uitgewerkt.

Echter is er op het internet een studie beschikbaar (Magnus, 2017), die aangeeft waarom programmeurs Kotlin écht moeten gebruiken. Hieruit blijkt dat Kotlin volledig compatibel zou zijn met Java. Bestaande projecten die geschreven zijn in Java, kunnen dus zonder enig probleem verdergezet worden in Kotlin, dit noemt men de automigration. Bestaande Java frameworks kan men verder gebruiken indien men wenst te programmeren in Kotlin. De taal zou makkelijk te begrijpen zijn voor iedereen die ervaring heeft met OOP (Object-Oriented Programming). Volgens ontwikkelaars die ervaring hebben met Swift 4 zou Kotlin bijna een kopie zijn van Swift 4. Voorbeelden hiervan zijn: geen puntkomma's om de regels af te sluiten en declaratie van variabelen gebeurd op identieke manier.

```
var number: Integer = 5
var text: String = "text"
```

Bovenstaande code toont beide voorbeelden aan.

Een andere studie (AJ, 2016), beschrijft dan de nadelen van Kotlin. Kotlin zou geen gebruik maken van namespaces. De static modifier zou verdwenen zijn, maar een alternatieve manier is dan weer mogelijk via @JvmField. Het compileren van projecten in Android Studio zou veel tijd in beslag nemen en het gebruik van annotations zou nog niet volledig geoptimaliseerd zijn. Zo blijkt dat er toch nog wat werk is aan Kotlin. We zitten dan ook nog maar aan versie 1.2.

Wat betreft het gebruik van Kotlin als cross-platform programmeertaal (DZone, 2015), wordt er sterk gewerkt aan Kotlin Native. Momenteel is dit nog in development, JetBrains zelf biedt reeds enkele projecten aan waar ontwikkelaars met de slag kunnen. Zo geven ze de programmeurs de mogelijkheid om reeds kennis te maken met de werking van Kotlin Native ook al is er nog geen documentatie over dit framework. Er is echter ook een kleine tutorial op de website van JetBrains aanwezig om een simpele applicatie te bouwen die 'Hello World' op het scherm print.

Op het internet zijn er voldoende bronnen te vinden om applicaties te bouwen met Kotlin voor één platform. Dit kan gaan over een webapplicatie die gebruik maakt van het Spring framework, waarbij Spring gebruik maakt van de taalfeatures van Kotlin. Wat Android betreft kan men gewoon in Android

Studio een applicatie geschreven in Android converteren naar Kotlin. Tenslotte is er de mogelijkheid om voor een desktopapplicatie, KotlinFX te gebruiken. KotlinFX is een library die dezelfde functionaliteit biedt als JavaFX, een library om interfaces te bouwen.

Zoals te lezen is, is er op het internet voldoende informatie te vinden om applicaties te bouwen voor één platform. Maar is het nu ook mogelijk om via één codebase verschillende platformen aan te spreken? Kan je door het eenmalig schrijven van code, een applicatie ontwikkelen die zowel op Android als op iOS kan draaien? Dit is het grote verschil met mijn onderzoek.

3. Methodologie

Vooraleer er aan de slag wordt gegaan met het uittesten van Kotlin Native, of het analyseren van voorbeeldprojecten van JetBrains, is het belangrijk om te onderzoeken hoe het nu komt dat Kotlin Native op verschillende platformen kan draaien. Kotlin is oorspronkelijk, net zoals Java, een programmeertaal die gebruik maakt van de Java Virtual Machine. Het zal daarom dus belangrijk zijn om te onderzoeken waarop Kotlin Native draait, wat de taak is van de LLVM compiler. Dit zal gebeuren aan de hand van een literatuurstudie.

Daarna worden de voorbeeldprojecten van JetBrains, geschreven in Kotlin Native, geanalyseerd om de werking van Kotlin Native te achterhalen. De bedoeling is om te kijken naar bepaalde aspecten. Hoe wordt de user interface opgebouwd, wordt er per platform een UI opgebouwd of is er slechts één UI? Hoe kunnen specifieke native modules gebruikt worden?

Eenmaal de werking van Kotlin Native duidelijk is en voldoende gedocumenteerd is, is het het doel om zelf aan de slag te gaan met Kotlin Native en een kleine cross-platform applicatie te schrijven.

4. Verwachte resultaten

Ten eerste wordt verwacht om uit de literatuurstudie te kunnen besluiten wat de werking is van de LLVM compiler. Hierdoor is het duidelijk hoe Kotlin Native er voor zorgt dat Kotlin voor verschillende platformen kan gebruikt worden.

Anderzijds wordt een zeer duidelijke documentatie verwacht over de werking van Kotlin Native zodat mensen die geïnteresseerd zijn om Kotlin Native in de toekomst te gebruiken deze documentatie kunnen gebruiken.

5. Verwachte conclusies

Uit dit onderzoek verwacht ik te kunnen concluderen dat Kotlin meer dan geschikt is als cross-platform programmeertaal. Dit aangezien Kotlin volledig ondersteund wordt door Google wat betreft de ontwikkeling van Android applicaties. Er wordt

nog steeds verder gebouwd aan Kotlin Native. Maar natuurlijk, Kotlin is een nieuwe taal, waar nog veel aan gesleuteld zal moeten worden. De toekomst ziet er enkel maar veel belovend uit. De kans is groot dat een groot aantal programmeurs zal overschakelen naar deze taal. Kotlin heeft zeker en vast de kracht om programmeurs, die al jaren in dit vak zitten en zich hebben vastgehecht aan een bepaalde programmeertaal, mee te slepen in het Kotlin-avontuur. Ook voor Swift programmeurs zal de aanpassing naar Kotlin niet groot zijn wegens de grote gelijkenissen tussen de twee programmeertalen.

Referenties

- AJ, A. (2016). Why you shouldn't switch to Kotlin. <https://medium.com/keepsafe-engineering/kotlin-the-good-the-bad-and-the-ugly-bf5f09b87e6f>.
- DZone. (2015). Kotlin and cross-platform. <https://dzone.com/articles/kotlin-for-cross-platform-mobile-app-development>.
- JetBrains. (2017a). Kotlin for server-side development. <https://kotlinlang.org/docs/reference/server-overview.html>.
- JetBrains. (2017b). Kotlin official website FAQ. <https://kotlinlang.org/docs/reference/faq.html>.
- JetBrains. (2017c). Kotlin official website, Kotlin Native. <https://kotlinlang.org/docs/reference/native-overview.html>.
- Magnus, V. (2017). Switch to Kotlin. <https://medium.com/@magnus.chatt/why-you-should-totally-switch-to-kotlin-c7bbde9e10d5>.