



**HoGent**

Faculteit Bedrijf en Organisatie

Kotlin als cross-platform programmeertaal

Ilias Van Wassenhove

Scriptie voorgedragen tot het bekomen van de graad van  
professionele bachelor in de toegepaste informatica

Promotor:  
Johan Van Schoor  
Co-promotor:  
Sander Goossens

Instelling: Endare

Academiejaar: 2017-2018

Tweede examenperiode



Faculteit Bedrijf en Organisatie

Kotlin als cross-platform programmeertaal

Ilias Van Wassenhove

Scriptie voorgedragen tot het bekomen van de graad van  
professionele bachelor in de toegepaste informatica

Promotor:  
Johan Van Schoor  
Co-promotor:  
Sander Goossens

Instelling: Endare

Academiejaar: 2017-2018

Tweede examenperiode



# Woord vooraf

Graag neem ik even de tijd om mijn onderwerpkeuze te motiveren.

Ik heb dit onderwerp gekozen omdat ik een grote passie heb voor programmeren. Tijdens het eerste semester van het derde academiejaar toegepaste informatica heb ik kennis mogen maken met Android en iOS. Hierbij heb ik ontdekt dat dit de richting is waarin ik wil verder gaan, namelijk mobiele applicaties. Tijdens de lessen Android werd er af en toe eens gediscussieerd waarom we nog steeds Java gebruikten in plaats van Kotlin. Ik had persoonlijk nog niet veel van Kotlin gehoord maar toen is mijn interesse ontstaan. Ik ben onmiddellijk dingen beginnen opzoeken over deze programmeertaal. Toen het tijd was om een bachelorproefvoorstel in te sturen wou ik zeker iets doen met mobiele applicaties en wanneer mijn stagebedrijf nog eens een onderwerp over Kotlin en cross-platformiteit doorstuurde, was de keuze snel gemaakt.

Tenslotte wil ik nog enkele personen bedanken.

Ten eerste wil ik mijn promotor, Johan Van Schoor, bedanken voor de hulp, raad en opbouwende commentaar die ik gekregen hebben bij het opstellen van deze bachelorproef. Mede dankzij zijn begeleiding ben ik tot een mooi resultaat gekomen.

Ten tweede zou ik graag mijn co-promotor, Sander Goossens, willen bedanken, wie ook mijn stagementor was. Hij stond altijd paraat om mijn vragen te beantwoorden en indien ik wat hulp nodig had was het nooit een probleem om even uitleg te geven.

Tenslotte zou ik iedereen waarmee ik heb samengewerkt willen bedanken voor de vlotte samenwerking. Er werd steeds samengewerkt op een zeer toffe en ontspannen manier waarbij iedereen gerespecteerd werd.

Hierdoor kan ik met een voldaan gevoel dit werk afgeven. Ik wens u veel plezier tijdens het lezen van mijn werkstuk.

Ilias Van Wassenhove

## Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus.

Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.



# Inhoudsopgave

0.1	Probleemstelling	13
0.2	Onderzoeksvraag	14
0.3	Onderzoeksdoelstelling	14
0.4	Opzet van deze bachelorproef	14
<b>1</b>	<b>Stand van zaken .....</b>	<b>15</b>
1.1	Wat is Kotlin	15
1.2	Kotlin en Android	16
1.3	Kotlin web en back-end	16
1.4	Kotlin Native	16
1.5	LLVM	16
1.6	Het gebruik van Kotlin	16
<b>2</b>	<b>Methodologie .....</b>	<b>19</b>

<b>3</b>	<b>Conclusie .....</b>	<b>21</b>
<b>A</b>	<b>Onderzoeksvoorstel .....</b>	<b>23</b>
<b>A.1</b>	<b>Introductie</b>	<b>23</b>
<b>A.2</b>	<b>State-of-the-art</b>	<b>24</b>
<b>A.3</b>	<b>Methodologie</b>	<b>25</b>
<b>A.4</b>	<b>Verwachte resultaten</b>	<b>25</b>
<b>A.5</b>	<b>Verwachte conclusies</b>	<b>25</b>

## Lijst van figuren

- 1.1 Hoeveelheid Github code in Kotlin **JetBrains12** ..... 17
- 1.2 User groups in de wereld **JetBrains12** ..... 17



## Lijst van tabellen



# Inleiding

De ontwikkeling van mobiele applicaties is een zeer belangrijke niche in de IT. Zo kunnen we drie verschillende besturingssystemen onderscheiden bij mobiele applicaties: Android, iOS en Windows. Die laatste wordt steeds minder en minder gebruikt. Volgens een artikel van **Statista2018** een portaal voor statistieken en onderzoek, heeft Android een marktaandeel van 85%. Dit wil dus zeggen dat 85% van de mensen die een smartphone heeft, een toestel is met Android.

De mobiele applicaties die draaien op Android werden allemaal geschreven in Java. Maar sinds kort is er een nieuwe mogelijkheid, Kotlin.

Kotlin is een programmeertaal die ontworpen is door JetBrains in 2011. Sinds 2017 biedt Google volledige ondersteuning om Android applicaties te programmeren in Kotlin.

## 0.1 Probleemstelling

Vanuit dit onderzoek zal blijken hoe goed Kotlin is als cross-platform ontwikkelingstaal en wat de mogelijkheden zijn van Kotlin om te programmeren voor verschillende platformen. Deze vraag kwam vanuit mijn stagebedrijf (Endare BVBA). Een developer bij Endare is reeds bezig met het aanleren van Kotlin en heeft reeds enkele Android projecten gedaan met Kotlin. Bij Endare wordt er zeer veel aan cross-platform ontwikkeling gedaan. Er worden maar weinig native applicaties gebouwd. Zo gebruiken ze Angular in combinatie met Ionic framework, React Native en heel af en toe eens Xamarin. Daarom is het belangrijk om te kijken of Kotlin een mogelijkheid biedt tot een nieuwe cross-platform programmeertaal.

## 0.2 Onderzoeksvraag

Mijn onderzoeksvraag is dus: In hoeverre kan Kotlin gebruikt worden voor cross-platform applicatieontwikkeling?

## 0.3 Onderzoeksdoelstelling

De criteria voor succes van deze bachelorproef zijn enerzijds een overzicht kunnen geven van de belangrijkste criteria van een cross-platform programmeertaal.

Anderzijds een succesvolle proof-of-concept kunnen opstellen waarin de huidige cross-platform mogelijkheden van Kotlin worden verduidelijkt.

## 0.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 1 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 2 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

**REFERENTIES AANPASSEN!**

In Hoofdstuk 1 zal onderzocht worden, via een literatuurstudie, wat de belangrijkste criteria zijn voor een cross-platform programmeertaal en in hoever Kotlin aan deze criteria voldoet.

In Hoofdstuk 2 zal ik een kleine proof-of-concept maken waarmee ik zal kunnen duidelijk maken hoe Kotlin er momenteel voor staat bij cross-platform development.

In Hoofdstuk 3, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.



# 1. Stand van zaken

Dit hoofdstuk is een literatuurstudie over de huidige stand van zaken rond Kotlin. Wat is Kotlin, Kotlin en Android, Kotlin en iOS, Kotlin en cross-platform development, Kotlin Native, LLVM en het gebruik van Kotlin. Na het lezen van dit hoofdstuk bent u volledig op de hoogte van de laatste updates van Kotlin.

## 1.1 Wat is Kotlin

Kotlin is een open-source programmeertaal die object-georiënteerde en functioneel programmeer features combineert. Kotlin is ook een static type programmeertaal. Dit betekent dat variabelen niet moeten gedefinieerd worden alvorens men deze gebruikt. Anders gezegd, het type van de variabele is gekend wanneer de code wordt gecompileerd.

Deze programmeertaal is ontworpen door JetBrains. JetBrains is een software ontwikkeling bedrijf dat gesticht is in het jaar 2000. Hun hoofdkantoor is gevestigd in Praag, Tjsechië. Hun core-business is het ontwikkelen van tools die gebruikt kunnen worden door verschillende types van software ontwikkelaars. Zo hebben zij IDE's <sup>1</sup> ontwikkeld voor Java, Ruby, Python, PHP, SQL, Objective-C, C++, C# en JavaScript.

---

<sup>1</sup>Integrated Development Environment

## 1.2 Kotlin en Android

In 2017 heeft Google bekend gemaakt dat het Kotlin volledig zou ondersteunen voor Android. Een Kotlin project maken in Android Studio is dan ook heel gemakkelijk. Je begint best met het installeren van de Kotlin plugin. Daarna maak je een nieuw Android Studio project aan. Origineel zal dit nog geschreven zijn in Java. Automigration

## 1.3 Kotlin web en back-end

<https://kotlinlang.org/docs/tutorials/httpservlets.html> <https://kotlinlang.org/docs/reference/server-overview.html> <https://medium.com/bcgdv-engineering/building-a-full-stack-web-app-in-kotlin-af8e8fe1f5dc>

## 1.4 Kotlin Native

<https://kotlinlang.org/docs/reference/native-overview.html>

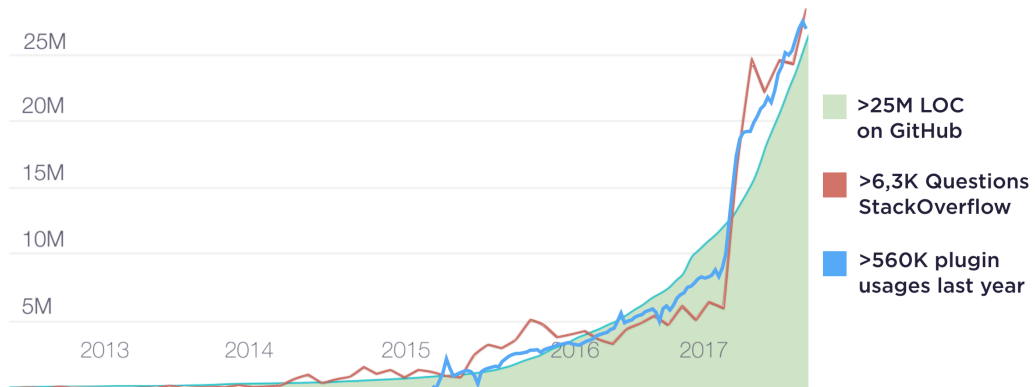
## 1.5 LLVM

<https://blog.jetbrains.com/kotlin/2017/04/kotlinnative-tech-preview-kotlin-without-a-vm/>

## 1.6 Het gebruik van Kotlin

Het gebruik van Kotlin is gedurende de jaren zeer sterk gestegen. Op de blog van JetBrains zijn grafieken te vinden waarmee men aantoont dat de populariteit van Kotlin enkel maar stijgt.

Deze tweede figuur toont de verschillende user groups over de volledige wereld. Het toont aan dat Kotlin toch zeer veel gebruikt worden in onze streken.

Figuur 1.1: Hoeveelheid Github code in Kotlin **JetBrains12**Figuur 1.2: User groups in de wereld **JetBrains12**



## 2. Methodologie

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Maecenas non massa. Vestibulum pharetra nulla at lorem. Duis quis quam id lacus dapibus interdum. Nulla lorem. Donec ut ante quis dolor bibendum condimentum. Etiam egestas

tortor vitae lacus. Praesent cursus. Mauris bibendum pede at elit. Morbi et felis a lectus interdum facilisis. Sed suscipit gravida turpis. Nulla at lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent nonummy luctus nibh. Proin turpis nunc, congue eu, egestas ut, fringilla at, tellus. In hac habitasse platea dictumst.

Vivamus eu tellus sed tellus consequat suscipit. Nam orci orci, malesuada id, gravida nec, ultricies vitae, erat. Donec risus turpis, luctus sit amet, interdum quis, porta sed, ipsum. Suspendisse condimentum, tortor at egestas posuere, neque metus tempor orci, et tincidunt urna nunc a purus. Sed facilisis blandit tellus. Nunc risus sem, suscipit nec, eleifend quis, cursus quis, libero. Curabitur et dolor. Sed vitae sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla facilisi.

### 3. Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit

lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consetetur libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem.

Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.



# A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

## A.1 Introductie

Ontwikkelaars van mobiele applicaties, webapplicaties en desktopapplicaties gebruiken vaak steeds dezelfde programmeertalen waarmee zij vertrouwd zijn. Android, iOS en eventueel Windows voor mobiele applicaties. Java, C# en Visual Basic voor desktopapplicaties. Angular, React en Javascript/jQuery voor webontwikkeling. Natuurlijk zijn er nog vele andere programmeertalen. Maar sinds enkele jaren geleden is er een nieuwe programmeertaal bijgekomen in dat lijstje, genaamd Kotlin.

Kotlin is voor vele programmeurs nieuw. Sinds dit jaar biedt Google volledige ondersteuning voor het gebruik van Kotlin bij Android applicatieontwikkeling. Waarom kan deze taal wel of niet gebruikt worden voor cross-platform ontwikkeling? Zo ja, wat zijn dan de eigenschappen van een goede cross-platform programmeertaal en waarom zou het een goede keuze zijn om Kotlin te gebruiken voor cross-platform ontwikkeling? Zou Kotlin Native mogelijk zijn in de toekomst? In dit onderzoek wordt er een antwoord geformuleerd op deze onderzoeksvragen.

## A.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. Steeds meer en meer boeken worden gepubliceerd.

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.

Op het internet is er een studie beschikbaar (**Pros**), 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. 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). Het gebruik van puntkomma's om de regel af te sluiten is optioneel. Een meer technische specificatie, zo zou je bij het vergelijken van objecten, niet meer `equals()` moeten gebruiken, maar kan je gebruik maken van de `==` operator. Deze zou vanaf nu controleren op structurele gelijkheid. Dit zijn features die toch allemaal héél positief klinken.

Een andere studie (**Cons**), 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.

Wat betreft het gebruik van Kotlin als cross-platform programmeertaal (**dzone**), wordt er sterk gewerkt aan Kotlin Native. Momenteel is dit nog niet beschikbaar. Er wordt echter binnenkort een alpha release verwacht. Ook is het nog niet mogelijk om Kotlin te compileren naar iOS. Het zou reeds al mogelijk zijn om Kotlin te gebruiken als programmeertaal voor desktop- en webapplicaties. Er bestaat een library genaamd KotlinFX, dit is een library die dezelfde functionaliteit aanbiedt als JavaFX, enkel geoptimaliseerd voor Kotlin. Voor webapplicaties kan men gebruik maken van het Spring framework, dat gebruik maakt van de taalfeatures van Kotlin.

Het grote verschil met al deze onderzoeken is dat er nooit een antwoord wordt geformuleerd of Kotlin een goede cross-platform programmeertaal is. Er wordt enkel en alleen maar onderzoek gedaan op één platform en meestal zijn dit de mobiele apparaten. Op het einde van mijn onderzoek zal ik een volledig antwoord kunnen geven of Kotlin nu wel of niet een goede cross-platform programmeertaal is.

Je mag gerust gebruik maken van subsecties in dit onderdeel.

## A.3 Methodologie

Om een goed antwoord te kunnen formuleren of Kotlin een goede cross-platform ontwikkelingstaal is, moet er natuurlijk onderzocht worden wat de eigenschappen zijn van een goede cross-platform programmeertaal. Aan welke criteria moet een programmeertaal precies voldoen om cross-platform te kunnen zijn? Zijn er bepaalde eisen?

Eens we deze onderzoeksvraag hebben kunnen beantwoorden, kunnen we dit gaan aftoetsen op Kotlin zelf. Enerzijds kunnen we dit theoretisch doen, anderzijds kunnen we dit doen via enkele testen. Zo kunnen we zowel voor desktopapplicaties, webapplicaties als mobiele applicaties enkele kleine programma's schrijven waarbij Kotlin vergeleken wordt met een andere vaak gebruikte cross-platform programmeertaal. We kunnen Kotlin vergelijken met Java, gebruikt bij desktop- en mobiele applicaties. Kotlin zou namelijk gebaseerd zijn op Java. Maar we kunnen ook gaan kijken naar specifiek bestaande cross-platform methodes zoals Xamarin, React, Angular, Titanium, ... Ervaren we sterke verschillen tussen deze programmeertalen? Hoe is het hergebruik van code bij Kotlin? Hoe is de toegang tot hardware van de verschillende platformen? Allemaal vragen die we aan de hand van theoretische en praktische testen kunnen beantwoorden.

## A.4 Verwachte resultaten

Enerzijds denk ik uit mijn resultaten te kunnen besluiten wat de eigenschappen zijn van een goede cross-platform programmeertaal. Dit is noodzakelijk om mijn onderzoek goed te kunnen uitvoeren.

Anderzijds verwacht ik uit mijn resultaten de conclusie te kunnen trekken dat Kotlin zowel voor mobiele, desktop- en webapplicaties uitermate geschikt zal zijn. Voor het gebruik van Kotlin op Apple toestellen zal het waarschijnlijk nog te vroeg zijn. Kotlin is nog niet compileerbaar naar Swift, tenzij dat Kotlin Native de komende maanden wordt uitgebracht.

## A.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 reeds gewerkt aan Kotlin Native. Ook zou het reeds mogelijk zijn om Kotlin te gebruiken om een webapplicatie uit te werken. 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