

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Teze diplomové práce**

**Využití programovacího jazyka Kotlin pro vývoj  
mobilních aplikací na platformě Android**

**Bc. Petr Kuška**

© 2018 ČZU v Praze

## **Souhrn**

Tato práce se zabývá zhodnocením programovacího jazyka Kotlin společnosti JetBrains z hlediska využití pro vývoj mobilních aplikací na platformě Android. Součástí práce je porovnání jazyka Kotlin s programovacím jazykem Java, jak z hlediska funkčnosti a možností jazyků, tak z hlediska výkonu. Práce se zaměřuje zejména na problematiku testování výkonu. Definuje testovací scénáře a navrhuje nástroje vhodné pro jejich měření. Implementuje navržené nástroje a provádí měření, jehož výsledky následně analyzuje a interpretuje. Dále se práce věnuje aktuálním možnostem vývoje mobilních Android aplikací a detailně popisuje funkcionality jazyků Java a Kotlin.

### **Klíčová slova:**

Kotlin, Java, Android, testování výkonu, měření výkonu, benchmark, funkcionality

Diplomová práce je reakcí na pochybnosti komunity vývojářů ohledně nově vznikajícího programovacího jazyka Kotlin vyvíjeného společností JetBrains, předním expertem v oblasti softwarových vývojových prostředí mnoha jazyků.

O programovacím jazyku Kotlin se v posledních uplynulých letech začalo mluvit na konferencích a internetových diskuzních fórech. Přednášky slibují nové přístupy k návrhu a implementaci, vyšší čitelnost, usnadnění údržby a nižší množství zdrojových kódů. Kotlin je Android komunitou dobře přijímán, avšak o jeho budoucnosti a postoji velkých společností, jako je Google, panují obavy.

Práce si klade za cíl zhodnotit využití programovacího jazyka Kotlin pro vývoj mobilních aplikací na platformě Android. Hlavním cílem práce je porovnání jazyka Kotlin s aktuálně nejrozšířenějším jazykem Java z hlediska výkonu a nabízených funkcionalit. Mezi dílčí cíle patří:

- pohled na aktuální možnosti vývoje mobilních aplikací na platformě Android z hlediska programovacích jazyků, jejich historie a vývoje do budoucnosti
- popsat programovací jazyky Kotlin a Java, jejich historii a vývoj, možnosti a metody použití
- pomocí připravených scénářů a nástrojů otestovat výkon obou programovacích jazyků
- porovnat jazyky dle čitelnosti kódu a náročnosti vypracování implementace
- zhodnocení výsledku testování a doporučení
- predikce budoucího vývoje

Metodika řešené problematiky byla založena na studiu a analýze odborných informačních zdrojů. Práce rovněž čerpá ze zkušeností autora získaných při vývoji mobilních aplikací v komerční sféře.

V první fázi bylo provedeno studium a analýza odborných informačních zdrojů. Na základě získaných poznatků jsou v teoretické části popsány aktuální možnosti vývoje mobilních aplikací pro platformu Android z hlediska programovacích jazyků, jejich historie a vývoje. Kapitoly Kotlin a Java charakterizují historii a vývoj, možnosti jazyků a jejich metody použití a detailněji se zabývají nabízenými funkcionalitami. Poslední

kapitola teoretické části je věnována možnostem testování výkonu a nástrojů s ním spojených.

Praktická část práce se zaměřuje na analýzu měřitelných vlastností. Definuje testované vlastnosti a testovací scénáře, pomocí nichž jsou vlastnosti měřeny. Na základě technologií zmíněných v teoretické části byly navrženy nástroje pro testování výkonu. Byla vypracována implementace těchto nástrojů a proběhla příprava testovacího prostředí. Pomocí vytvořených nástrojů bylo provedeno měření, agregace a analýza naměřených dat na základě statistických charakteristik.

Výběr vlastností byl proveden na základě několika kritérií. Prvním kritériem bylo, aby vybrané vlastnosti korespondovaly s vlastnostmi, které jsou ceněny v praxi. Nebylo žádoucí testovat vlastnost, která v konečném důsledku není rozhodující pro praktické využití a není zahrnuta v rozhodovacím procesu při výběru vhodného jazyka. Dalším kritériem byla možnost objektivnímu přístupu k testování a relativně snadné možnosti měřitelnosti.

V praxi mohou být rozhodujícími vlastnostmi a jinými faktory např. rozšířenost jazyka na trhu práce, či oblíbenost jazyka mezi týmem programátorů, kteří byli na projekt přiděleni (často právě ti bývají rozhodujícím faktorem). Tyto vlastnosti a faktory je obtížné kvantifikovat a jejich měření často podléhá humanitním oborům.

Na základě zkušeností autora, byly vybrány následující vlastnosti:

- doba běhu
- doba kompilace (čisté)
- doba kompilace (inkrementální)
- délka zdrojového kódu

Testovací scénář byl definován jako metoda pro otestování a měření vybrané testované vlastnosti. Každý testovací scénář se skládá z posloupnosti kroků, v jakých je testování prováděno, a z testovacích případů, na kterých jsou měřeny testované vlastnosti. Testovací případy mohou sestávat z krátkých spustitelných částí kódu, ze zdrojových kódů celé aplikace, nebo z projektového repozitáře.

Pro každý testovací scénář je vytvořen benchmark. Benchmark slouží jako nástroj pro opakované provádění měření. Implementuje jednotlivé navržené kroky a umožňuje automatizované měření. Z důvodu eliminace vlivů vnějšího prostředí bylo měření provedeno opakovaně pomocí benchmarku a výsledky měření byly následně statisticky vyhodnoceny.

Výsledky testování výkonu prokázaly, že doba běhu programů v Kotlinu je shodná s Javou. Doba čisté kompilace zdrojového kódu Kotlinu je delší než u Javy. Inkrementální kompilace, která je v praxi používána nejčastěji, se ukázala být nepatrně kratší než u Javy. Dalším výsledkem byla délka zdrojového kódu. Bylo zjištěno, že jak počet řádků, tak počet znaků kódu je u Kotlinu nižší než u Javy. Na základě výsledků testování lze Kotlin považovat za jazyk se stejným, případně lepším, výkonem než Java.

Porovnání jazyků na základě funkcionalit nebylo v práci provedeno, jelikož nebyla nalezena metoda objektivního přístupu. Práce se zabývala především funkcionalitami, které jeden z jazyků nabízí a druhý postrádá. Z toho důvodu nebylo možné použít komparativní metody vícekriteriální analýzy variant. Doporučením autora je provedení širší analýzy odborných informačních zdrojů a odpovídajících metodik, případně vypracování případové studie.

Na základě zkušeností autora s vývojem mobilních aplikací v Javě lze prohlásit, že Kotlin oproti Javě žádné podstatné funkcionality nepostrádá. Po zkušenostech získaných studiem odborných informačních zdrojů a vypracováním implementací autor konstatoval, že Kotlin umožňuje stejné přístupy k vývoji aplikací jako Java a nabízí i některé další.

Problematika čitelnosti kódu a náročnosti vypracování byla autorem vyhodnocena jako velice subjektivní a z důvodu nedostatku materiálů pro objektivní posouzení, nebyla samostatně řešena. Nicméně, na základě zjištění, že počet řádků u Kotlinu klesá vzhledem k Javě rychleji než počet znaků, lze v kombinaci s teoretickými východisky o nabízených funkcionalitách usuzovat, že Kotlin je expresivnějším jazykem, který umožňuje zápis složitějších struktur Javy pomocí jednoho slova či výrazu. Takové jazyky mají obvykle složitější pravidla, čímž kladou vyšší nároky na uživatele jazyka, a tedy i na jeho čitelnost. Náročnost vypracování, je-li chápána jako množství stráveného času, tak závisí na zkušenostech a dovednostech programátora, který daný jazyk používá.

Doporučením pro rozšíření práce může být definice dalších testovacích scénářů a odpovídajících benchmarků. Vzhledem k nabízeným funkcionalitám umožňuje Kotlin vytvářet pokročilé architektonické a implementační struktury. Zajímavými testovanými vlastnostmi by tedy mohly být např. počet souborů zdrojového kódu, průměrná délka zdrojového kódu na soubor, počet úrovní v hierarchii tříd nebo množství tzv. *boilerplate* kódu.

Dalším doporučeným postupem je implementace vyššího množství testovacích případů pro testovací scénář doby běhu programu. Zvýšení doby běhu programu by vedlo ke snížení směrodatné odchylky a ke zvýšení přesnosti výsledků.

Řešením problematiky porovnání čitelnosti a náročnosti vypracování kódu je provedení širší analýzy odborných informačních zdrojů a odpovídajících metodik, případně vypracování případové studie.

V průběhu tvorby této práce Android tým na konferenci Google I/O oficiálně oznámil podporu jazyka Kotlin pro vývoj Android aplikací. Tím úspěšně eliminoval obavy komunity o budoucnost jazyka Kotlin. Android Studio, oficiální nástroj vydávaný společností Google, již vývoj v Kotlinu podporuje a dá se očekávat, že podpora jazyka Kotlin bude i nadále pokračovat.

## Seznam použitých zdrojů

- [1] LEIVA, Antonio. *Kotlin for Android Developers: Learn Kotlin the easy way while developing an Android App*. N Charleston SC: CreateSpace Independent Publishing Platform, 2016. ISBN 978-1530075614.
- [2] *Kotlin Programming Language* [online]. JetBrains, 2018 [cit. 27.3.2018] Dostupné z: <https://kotlinlang.org/docs/reference/>
- [3] *Java Platform Standard Edition 7 Documentation. Java SE Documentation* [online]. 500 Oracle Parkway Redwood Shores, CA 94065: Oracle Corporation, 2016 [cit. 2017-08-14]. Dostupné z: <http://docs.oracle.com/javase/7/docs/>
- [4] *Comparison to Java - Kotlin Programming Language* [online]. JetBrains, 2018 [cit. 27.3.2018] Dostupné z: <https://kotlinlang.org/docs/reference/comparison-to-java.html>
- [5] GARGENTA, Marko a NAKAMURA, Masumi. *Learning Android. Second edition*. Sebastopol, CA: O'Reilly Media, 2014. ISBN 978-1-449-31923-6.
- [6] KLAUSEN, Paul. *Java 1: Basic syntax and semantics: Software Development*. 1st edition. Bookboon.com, 2017. ISBN 978-87-403-1689-6
- [7] NAMBIAR, Raghunath, POESS Meikel. *Performance evaluation and benchmarking: First TPC Technology Conference, TPCTC 2009, Lyon, France, August 24-28, 2009 : revised selected papers*. New York: Springer, c2009. Lecture notes in computer science, 5895. ISBN 978-3-642-10423-7.