

**Česká zemědělská univerzita v Praze**  
**Provozně ekonomická fakulta**  
**Katedra informačních technologií**



**Teze diplomové práce**

**Programovací jazyk Swift**

**Bc. Marcel Soukeník**

**Vedoucí práce: Ing. Jiří Vaněk, PhD.**

© 2017 ČZU v Praze

## **Souhrn**

Téma této diplomové práce je programovací jazyk Swift. Cílem práce je vymezení funkcionality programovacího jazyka Swift a poskytnout její kompletní přehled. Dále na praktických příkladech pak ukázat syntaxi a využití jazyka Swift. Práce je rozdělena do několika částí. Teoretická část zahrnuje přehled řešené problematiky. Praktická část analyzuje zdrojový kód a syntaxi, implementuje příklady pokrývající vlastnosti jazyka a porovnává syntaxi a výkon ostatních programovacích jazyků. Závěrečná část shrnuje a zhodnocuje veškeré získané výsledky a poznatky.

## **Klíčová slova**

Swift; Apple; Mac; Programovací jazyky; Informační technologie; Open Source

# 1 Úvod

Programovací jazyk Swift byl představen firmou Apple v roce 2014 na vývojářské konferenci WWDC. Hlavní motto, jež na konferenci zaznělo a mělo by vystihnout jazyk Swift, je: „Objective-C bez zatížení jazyka C.“ Objective-C vznikl v roce 1986 jako objektová nadstavba jazyka C. V současné době, kdy převládá trend aplikací pro mobilní platformy, se Objective-C v porovnání s jazyky jako je Java, Python nebo C# jeví jako zastaralý. Firma Apple proto vyvinula nový programovací jazyk, čímž dala najevo snahu udržet si krok v oblasti inovací a rozšířit si vývojářskou obec.

Swift má ambice být moderním, bezpečným a jednoduchým jazykem, se kterým lze vytvořit jak jednoduchou aplikaci „Hello World“, tak celý operační systém. Licence jazyka Swift byla uvolněna 3. prosince 2015 jako Open Source, načež se otevřela možnost všem vývojářům přispět k vývoji jazyka Swift či použít Swift pro jinou platformu než Apple. Také byly uvolněny kompilátory pro Linux, tudíž není nutné vlastnit Mac pro využití tohoto programovacího jazyka.

Tématem je pokrýt a prozkoumat všechny významné vlastnosti programovacího jazyka Swift. Dále zhodnotit potenciál a ambice, kde se dá jazyk Swift využít.

## 2 Cíle práce a metodika

Primárním cílem práce je vymezení funkcionality programovacího jazyka Swift, aplikovat šíři jeho implementačního využití a kriticky zhodnotit gramatiku a syntax daného jazyka. Další cíle jsou představit základní vlastnosti programovacího jazyka Swift, podrobně prozkoumat architekturu jazyka přes zdrojový kód, vyvinout funkční části kódu v jazyce Swift vymežující nové a kritické vlastnosti jazyka, implementovat obecné návrhové vzory v jazyce Swift, porovnat procesní výkon a rychlost s ostatními jazyky, formulovat závěry a doporučení.

Práce je rozdělena do dvou částí. První část uvede přehled řešené problematiky. Nastíněna je klasifikace programovacích jazyků, základní charakteristika programovacího jazyka Swift, jeho základní knihovna a stručný vývoj jazyka Swift, porovnání s ostatními rozšířenými programovacími jazyky, syntaxe jazyka Swift a přehledné informace o návrhových vzorech.

Druhá, praktická část se věnuje analýze zdrojového kódu jazyka Swift. Podrobně je prozkoumána syntax, kompilace a výkon, nadto dojde k implementaci hlavních vlastností jazyka Swift a návrhových vzorů. Nakonec následuje porovnání s jazyky Objective-C a jazykem Python.

### **3 Výsledek práce a závěr**

Swift je relativně jednoduchý a flexibilní jazyk a zároveň je komplexní a funkční. Swift je vhodný k výuce programování i vytváření komplexních aplikací s UI, podpůrných knihoven či jednoduchých programů pro příkazovou řádku. Vývoj a podpora jazyka je konzistentní. Jazyk Swift má vytvořené stabilní základy a syntaxe je logická a snadno pochopitelná.

Funkce ve Swiftu prošly významnou inovací. Jazyk byl navržen tak, aby umožňoval objektově-orientované programování i nové funkční programování. Swift plně nahrazuje Objective-C. S jazykem Swift lze bez jakýchkoli komplikací implementovat objektově-orientované návrhové vzory. Swift má i přístup k ukazatelům, tudíž i k operacím na nízké úrovni systému, ovšem pro programování (např. ovladačů) tento jazyk není vhodný. Výkonnostní test nepotvrdil, že by jazyk Swift byl rychlejší než Objective-C či Python.

Swift znamená pro Apple dlouhodobou investici a podnikl veškeré kroky k tomu, aby jazyk uspěl. Navázal na Objective-C a zachoval jeho funkcionalitu, kterou rozšířil o nové vlastnosti z jiných jazyků. Od první verze došlo k přeprogramování celého API, aby bylo možné naprogramovat plnohodnotné produkty stejně jako s Objective-C. Xcode obsahuje nástroje, které umožňují přechod na novější verze Swiftu a kompilátor umí přeložit i starší verze Swiftu pomocí ABI. Swift je také na platformě Linux, ovšem nemá žádné API či vývojové prostředí jako je Xcode na Macu. Čtyřletý vývoj Swiftu je však pro masivní adaptaci ještě krátká doba. Je však jen otázka času, kdy se jazyk Swift rozšíří i mimo Apple platformu.

## 4 Seznam literatury

- [1] APPLE, *The Swift Programming Language*, [online]. [cit. 16-08-2016].  
<[https://swift.org/documentation/TheSwiftProgrammingLanguage\(Swift3\).epub](https://swift.org/documentation/TheSwiftProgrammingLanguage(Swift3).epub)>.
- [2] APPLE, *About Swift*. [online]. [cit. 20-08-2016].  
<<https://swift.org/about/#swiftorg-and-open-source>>.
- [3] Erich GAMMA, Richard HELM, Ralph JOHNSON, John VLISSIDES, *Návrh programů pomocí vzorů – Stavební kameny objektově orientovaných programů*. Praha : Grada, 2003. 386s. ISBN: 80-247-0302-5.
- [4] Scott WLASCHIN, *Functional Programming Design Patterns*. NDC London 2014. [online]. [cit. 2017-11-6]. <<https://fsharpforfunandprofit.com/fppatterns/>>.
- [5] Jiva DeVOE, *Objective-C Developer Reference*. Indianapolis, Wiley Publishing, Inc, 2011, 382s., ISBN: 978-0-470-47922-3.
- [6] Michael H. GOLDWASSER, David LETSCHER, *Object-Oriented Programming in Python*. New Jersey, 2008 Pearson Education, Inc. 666s. ISBN: 978-0-13-615031-2.
- [7] Apple, *Using Swift with Cocoa and Objective-C*. 2014-06-02. [online]. [cit. 2-10-2017]. <<https://itunes.apple.com/cz/book/using-swift-with-cocoa-and-objective-c-swift-4/id888894773?l=cs&mt=11>>.