

Razvoj programskog jezika Objective-C

Seminarski rad u okviru kursa
Dizajn programskih jezika
Matematički fakultet

Nevena Manevska
manevsska@gmail.com

18. decembar 2019.

Sažetak

U ovom radu prikazane su osnove razvoja programskog jezika **Objective-C** kao i njegove glavne karakteristike. Takođe ukratko su opisani jezici koji su najviše uticali na njegov razvoj, kao i jezici koji su se kasnije razvijali pod njegovim uticajem.

Sadržaj

1	Uvod	2
2	Osnovno o Objective-C-u	2
3	Istorija	3
4	Sintaksa	4
5	Razvojno stablo	5
5.1	C	5
5.2	Smalltalk	6
5.3	Java	7
5.4	Groovy	7
5.5	Swift	7
6	Dodatak	8
7	Zaključak	8
	Literatura	9

1 Uvod

Ranije se verovalo da programeri koji koriste *Objective-C* pripadaju malo, elitnoj grupi. Oni su pisali programe za *Mac*, pravili aplikacije za *iOS* koji je tada imao veoma malu tržišnu vrednost, a još manji potencijal za profit. Danas, zahvaljujući popularnošću *iPhone-a*, *iPod Touch-a*, *iPad-a*, porasla je i popularnost *Objective-C-a*. Svi, počevši od ljudi kojima je to hobi, preko naučnika, do komercijalnih developera i konsultanata prave aplikacije za *iOS* koristeći *Objective-C*. [8]

Objective-C je opšte-namenski, objektno-orijentisani programski jezik koji dodaje neke karakteristike *Smalltalk-a* programskom jeziku *C*. Bio je glavni jezik koji je podržavala kompanija *Apple* za svoje operativne sisteme *macOS*, *iOS* i *iPadOS*, kao i njihove aplikacione programske interfejsne (API) *Cocoa* i *Cocoa Touch* sve dok nije predstavljen *Swift* 2014. godine. [11, 10]



Slika 1: Logo programskog jezika Objective-C

2 Osnovno o Objective-C-u

Jezik je originalno nastao ranih 80ih, tačnije 1983. kao mešavina *Smalltalk-a* i *C-a*. U osnovi, *Objective-C* je ANSI standardna verzija programskog jezika *C* obmotana skupom ekstenzija, inspirisanih *Smalltalk-om*, koji jeziku daju *objektno-orijentisane* mogućnosti kao i nekoliko drugih poboljšanja koja ne dobijate iz uobičajene verzije *C-a*. [14, 8]

Kasnije je izabran za glavni jezik koji koristi *NeXT* za svoj operativni sistem *NeXTSTEP* čiji su derivati i *macOS* i *iOS*. Generički *Objective-C* programi koji ne sadrže *Apple* biblioteke ili biblioteke koje mogu biti prenete i reimplementirane za korišćenje na drugim operativnim sistemima takođe mogu biti prevedeni od strane *GNU Compiler Collection* (*GCC*) i *Clang*.

Izvorni kodovi programa pisanih u *Objective-C* programskom jeziku često imaju *.m* ekstenziju, dok *Objective-C* hederi/intefrejsi imaju *.h* ekstenziju kao kod fajlova jezika *C*. *Objective-C++* fajlovi su obeleženi *.mm* ekstenzijom.

3 Istorija

Objective-C su primarno kreirali Bred Koks (engl. *Brad Cox*) i Tom Lov (engl. *Tom Love*) 1983. godine sa ciljem da olakšaju ljudima pisanje jasnijeg i modelarnijeg koda.[8] Koks su zainteresovali problemi ponovne upotrebe u softverskom dizajnu i programiranju. Shvatio je da bi jezik poput *Smalltalk-a* bio neprocenjiv za izgradnju razvojnih okruženja za programere sistema na ITT-u¹. Takođe, primetili su da je zaostajanje kompatibilnosti sa *C-om* veoma važno u ITT-ovom telekom inženjerskom okruženju. [4]

Koks je počeo da sastavlja pred-processor za *C* kome će dodati neke od mogućnosti *Smalltalk-a*. Došao je do objektno-orijentisane ekstenzije jezika *C*, koju je nazivao **OOPC** — *Object-Oriented Pre-Compiler (objektno-orijentisani pred-kompajler)*. [5] Lova je unajmila kompanija *Schlumberger Research* 1982. godine i dobio je priliku da napravi prvu komercijalnu kopiju *Smalltalk-a-80*, koja je uticala na razvoj njihove ideje. Želeli su da pruže podršku objektima na fleksibilan način, pridruže izvestan broj upotrebljivih biblioteka kao i da omoguće da se kod (i svi resursi potrebni kodu) spajaju u jedan *cross-platform* format.

Lov i Koks su na kraju kreirali kompaniju **PPI** - *Productivity Products International* da bi komercijalizovali njihov proizvod, što je spojilo *Objective-C* kompajler sa bibliotekama klase. Koks je 1986. godine u knjizi *Object-Oriented Programming, An Evolutionary Approach (Objektno programiranje, Evolucijski pristup)* objavio glavni opis **Objective-C-a** u svom izvornom obliku. Iako je bio oprezan da istakne da problem ponovne upotrebe postoji i van onoga što **Objective-C** pruža, jezik je često bio upoređivan po karakteristikama sa drugim jezicima.

1988. *NeXT*² je licencirao **Objective-C** kompanije *StepStone*³ i nadogradio *GCC* prevodilac da podrži **Objective-C**. Kako bi zaobišao uslove *GPL-a*⁴[1], *NeXT* je prvobitno nameravao da isporuči *Objective-C frontend* odvojeno, omogućavajući korisniku da ga poveže sa *GCC-om* kako bi proizveo kompajler koji se može izvršiti. Nakon što ga je prihvatio Ričard Stalmen (engl. *Richard M. Stallman*), ovaj plan je odbijen nakon što se Stalmen konsultovao sa advokatima *GNU-a* i *NeXT* je pristao da **Objective-C** postane deo *GCC-a*. Rad na proširivanju *GCC-a* vodio je Stiv Nerof (engl. *Steve Naroff*). Izmene kompajlera bile su dostupne u skladu sa uslovima licenci *GPL*, ali runtime biblioteke nisu, što je učinilo da doprinos otvorenog koda bude neupotrebljiv za širu javnost. Denis Glatin (engl. *Dennis Glatting*) napisao je prvi GNU *Objective-C* runtime tokom 1992. godine, a danas je u upotrebi GNU *Objective-C* runtime koji je razvio Kresten Trup (engl. *Kresten Krab Thorup*) dok je bio student na Danskom univerzitetu 1993. godine.

¹ITT Inc., formalno ITT Corporation, je poznata Američka kompanija iz White Plains, New York. Kompanija proizvodi specijalne delove za vazduhoplovna, transportna, energetska i industrijska tržišta.

²NeXT Inc., je bila Američka kompanija koju je osnovao Stiv Jobs 1985. godine nakon što je dobio otkaz u kompaniji Apple. Njihov inovativni objektno-orijentisani operativni sistem NeXTSTEP bio je veoma uticajan. NeXT je razvio AppKit i FoundationKit biblioteke na kojima su zasnovani NeXTSTEP korisnički interfejs i Interface Builder.

³StepStone je novo ime kompanije PPI, vlasnika *Objective-C-a*

⁴The GNU General Public License (GNU GPL or GPL) je široko upotrebljena besplatna softverska lincenca koja garantuje korisnicima slobodu pri pokretanju, učenju, deljenju i menjanju softvera. Licencu je originalno napisao Richard Stallman, direktor kompanije Free Software Foundation (FSF), za GNU projekat.

Nakon što je 1996. godine kupio *NeXT*, *Apple Computer* je koristio *OpenStep*⁵ u svom tada novom operativnom sistemu, *Mac OS X*. Tu su obuhvaćeni **Objective-C**, *NeXT*-ov alat za razvojne programere zasnovane na **Objective-C-u** (*Project Builder*) i njegov alat za dizajn interfejsa (*Interface Builder*), oba sada spojena u jednu aplikaciju - *Xcode*. Većina *Apple*-ovog trenutnog API-ja za *Cocoa* zasnovan je na objektima *OpenStep* interfejsa i najznačajnije je okruženje **Objective-C-a** koje se koristi za aktivni razvoj. Na WWDC⁶ 2014. *Apple* je predstavio novi jezik, *Swift*, koji je okarakterisan kao **Objective-C** bez **C-a** („*Objective-C without the C*“).

4 Sintaksa

Ukoliko ste do sada imali bilo kakvog iskustva sa nekim od programskih jezika *C*, *C#* ili *Java* učenje **Objective-C** ne bi trebalo da vam stvara probleme. Većina programera smatra da je potrebno jako malo vremena za učenje same sintakse, ali da je dosta zahtevnije učenje o svim alatima i kontrolama dostupnim u *Cocoa* za *Mac OS X*. [8, 3]

Objective-C je vrlo kratak sloj na vrhu *C-a*. To znači da se korišćenjem **Objective-C** kompajlera mogu kompajlirati i *C* datoteke, a u klasama se potpuno slobodno može koristiti *C* kod.

Objektna sintaksa je derivat iz *Smalltalk-a*.

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        // insert code here...
        NSLog(@"Hello, World!");
    }
    return 0;
}
```

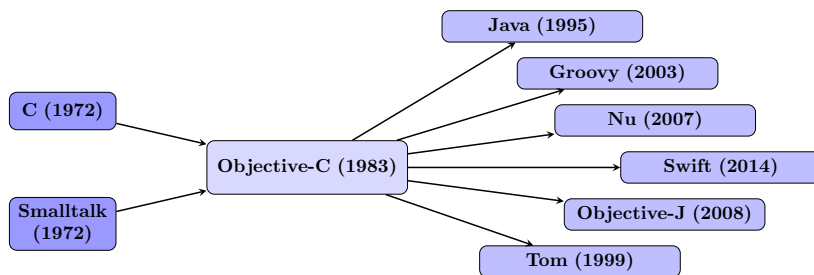
Slika 2: *Hello world!* u **Objective-C**

⁵OpenSTEP - NeXT-ova platformu za prilagođeno programiranje.

⁶The Apple Worldwide Developers Conference (WWDC, poznata i kao Dub Dub) je konferencija koju Apple održava jednom godišnje u San Jose, California. Apple koristi ovaj događaj kako bi prikazao svoje nove softvere i tehnologije korisne programerima

5 Razvojno stablo

Dakle, kao što smo već videli, na nastanak jezika *Objective-C* najviše uticaja imali su programski jezici *C* i *Smalltalk*, međutim, on je kasnije bio značajan za razvoj drugih jezika poput *Groovy*, *Java*, *Nu*, *Objective-J*, *TOM*, *Swift*. Razvojno stablo jezika Objective-C može se videti na slici 3.



Slika 3: Razvojno stablo jezika Objective-C

5.1 C

C je proceduralni programski jezik, nastao 1972. godine. [13, 12] Autor jezika je Denis Riči, a nastao je u istraživačkom centru Bell Laboratories u Nju Džerziju za potrebe operativnog sistema *UNIX*.

Prve verzije operativnog sistema *UNIX* su bile pisane koristeći *assembler*, ali se ovaj projekat brzo približavao svom kraju jer je programski kod *assemblera* izrazito velik i težak za održavanje. Ideja je bila načiniti programski jezik koji bi imao sve ovo, pružio komfor već postojećih viših programskih jezika, ali čija arhitektura i logika ne bi mnogo odudarala od *asemblerske*.

UNIX operativni sistem, *C* kompajler i svi *UNIX*-ovi aplikativni programi su napisani na *C*-u. Proslavivši se svojom brzinom, postao je jezik izbora za mnoge buduće projekte, poput različitih verzija *Windows*-a i *UNIX*-a, ali i u aplikativnom programiranju za mnogobrojne aplikacije tokom decenija.

Programski jezik *C* je jezik opšte namene. Iako je razvijan kao „jezik za sistemsko programiranje”, podjednako dobro se koristi za pisanje značajnih programa u različitim oblastima. *C* je takode veoma uticao na mnoge druge popularne programe, posebno na *C++*, koji je originalno razvijan kao nadgradnja *C*-a.

Organizacione celine *C*-a su *funkcije*. Početna tačka izvršavanja programa je tzv. „glavna funkcija” (engl. *main*), a celokupan program se organizuje u određeni broj drugih funkcija, u zavisnosti od potrebe. Telo funkcije je ograničeno vitičastim zagradama, kako bi joj se označili početak i kraj. Ona može pozivati druge funkcije ili samu sebe (rekurzija). Funkcije se posmatraju kao odvojene, izolovane celine čime se postiže modularnost koda koji je kao takav lak za održavanje i dalje razvijanje. Po završetku funkcije *main*, čitav program se završava i oslobađa iz memorije.

Uz standardni *C* se distribuiraju i *standardne biblioteke*. To su datoteke koje sadrže mnoštvo korisnih alata koji se koriste u svakodnevnom radu. Programer može da uključi jednu ili više ovih biblioteka u svoj program, a može i sam da piše biblioteke za svoje potrebe.

5.2 Smalltalk

Smalltalk je objektno-orijentisani, dinamički, reflektivni programski jezik. Jedan je od mnogih objektno-orijentisanih jezika baziranih na programskom jeziku *Simula*. Razvila ga je grupa istraživača iz kompanije *Xerox PARC* na čelu sa Alan Kejom (engl. *Alan Kay*). [6, 9]

Nastankom verzije *Smalltalk-80* ovaj programski jezik je po prvi put publikovan. Ona se sastojala od dve podverzije:

Smalltalk-80 Version 1 ustupljena je nekolicini kompanija i univerziteta radi testiranja i implementacije jezika na njihovim platformama

Smalltalk-80 Version 2 predstavljena kao finalna verzija dostupna široj javnosti.

Danas su u svetu programiranja popularne dve implemetacije *Smalltalk-a* zasnovane upravo na verziji *Smalltalk-80*:

Squeak - open source implementacija izvedena iz *Smalltalk-80 Version 1* pomoću *Apple Smalltalk-a*

VisualWorks - izvedena iz *Smalltalk-80 Version 2* pomoću *Smalltalk-80 2.5* i *ObjectWorks-a*.

Godine 1998., Smalltalk postaje ANSI-standardizovani jezik i dobija naziv *ANSI Smalltalk*.

Mnoge stvari koje su u današnjem programiranju podrazumevane, nastale su sa razvojem *Smalltalk-a*:

- Prvi je programski jezik sa integrisanim razvojnim okruženjem.
- Jedan je od prvih programskih jezika sa grafičkim okruženjem koje omogućava komunikaciju pomoću miša.
- Napravljen je tako da ne prekida rad programa usled neočekivanog ponašanja. Na primer, ukoliko se objektu pošalje poruka koju on ne razume, debager nam nudi opciju da definišemo metod koji odgovara datoj poruci.
- Smalltalk sadrži *TDD*⁷ alate koji su još uvek jedni od najboljih u svetu programiranja.
- Takođe, u Smalltalk-u je podržana i 'Just in time' kompilacija.

Smalltalk u prevodu sa engleskog jezika znači "ćaskanje" što ilustruje njegovu jednostavnu sintaksu. Ovaj programski jezik ne sadrži kontronle strukture (npr. *if*, *for* i *while* petlja) i ne podržava statičke promenljive. Takođe ne sadrži primitivne tipove (npr. *int* i *boolean*), tačnije ne postoji razlika između klasnih i primitivnih tipova - svi primitivni tipovi podržani ovim programskim jezikom predstavljaju objekte njima odgovarajućih klasa. Pošto su sve vrednosti u *Smalltalk-u* predstavljene objektima, klase su takođe objekti. Svaka klasa je instanca neke metaklase, dok su sve metaklase instance klase *Metaclass*.

⁷Test-driven development (TDD) je proces razvoja softvera koji se oslanja na ponavljanje vrlo kratkog razvojnog ciklusa: zahtevi se pretvaraju u vrlo specifične testne slučajeve, a zatim se softver poboljšava tako da testovi prođu.

5.3 Java

Java (engl. *Java*; izgovor: java, džava) je *objektno-orijentisani programski jezik*, koji je razvila kompanija *Sun Microsystems* početkom 1990-ih godina. Prva verzija je zvanično objavljena 1995. godine. Mnogi koncepti **Java** su zasnovani na jeziku *Oberon*. Izbačen je koncept modula, a uvedeni su paketi kakve danas znamo koji se oslanjaju na fajl sistem. Takođe, uveden je formalan koncept klasa iz objektno-orijentisane paradigme. Osim toga, jezik ima sintaksu sličnu jezicima *C* i *C++*, ali je mnogo stroži pri prevodenju. Dizajniran je tako da bude nezavisan od platforme i sa pojednostavljenim upravljanjem memorijom. [2]

Java je, uz *Kotlin*, zvanično podržan jezik za izradu mobilnih aplikacija za *Android* uređaje.

- U stvaranju **Java** jezika oslanjalo se na pet osnovnih ciljeva:
- Mora biti jednostavan, objektno orijentisan i prepoznatljiv.
- Mora biti temeljan i siguran.
- Mora biti arhitektički neutralan i prenosiv.
- Mora da se izvršava sa visokim performansama.
- Mora biti interpretiran, praćen i dinamičan.

5.4 Groovy

Groovy je *objektno-orijentisan programski jezik opšte namene*. Kreiran je za *Java* platformu i može da se koristi kao alternativa *Java* programskom jeziku. Sintaksa **Groovy** jezika liči na sintaksu *Java* programskog jezika, štaviše, veliki deo koda koji je pisan u *Java* programskom jeziku sintaksno je ispravan **Groovy** kod. Programi pisani u **Groovy** jeziku mogu da se povežu sa *Java* kodom i da se koriste *Java* biblioteke. Kada je potrebno da se **Groovy** kod izvrši, dinamički se prevodi u *Java* bajt kod. [7]

Groovy sadrži i osobine koje izdvajaju programske jezike kakav je *Python*, *Ruby*, *Smalltalk*. Na primer poseduje i dinamičku proveru tipova podataka.

Groovy može da se koristi za pisanje samostalnih programa, za pisanje delova *Java* programa ili kao skript jezik.

5.5 Swift

Swift je *programski jezik opšte namene*. Podržava *imperativni, objektno-orijentisani i funkcionalni* način programiranja. Stvoren od strane preduzeća *Apple* da bi zamenio **Objective-C** pri programiranju *iOS* i *OS X* programa. Napravljen je upotrebom *LLVM* programskog prevodioca koji je dostupan u *Xcode 6 beta*. [15, 16]

Zvanična aplikacija *WWDC* je 2. juna 2014. postala prva javno dostupna aplikacija napisana u **Swift**-u.

6 Dodatak

Da li se programski jezik Objective-C i dalje koristi i koja je njegova poslednja stabilna verzija?

Dok za većinu programskih jezika možemo da kažemo da su korišćeni, Objective-C je i dalje jedan od glavnih jezika koje koristi kompanija Apple. Poslednja stabilna verzija je 2.0.

Za šta se koristi?

Objective-C je programski jezik koji je korišćen za OS X i iOS operativne sisteme kao i njihove aplikacione programske interfejse.

Da li je Objective-C bolji od programskog jezika Swift? Da li je lak za učenje?

Programski jezik Objective-C je star preko 30 godina što znači da iziskuje više nespretnih sintakse, dok sa druge strane Swift zahteva mnogo manje koda i mnogo je jednostavniji za čitanje i razumevanje. Međutim i ako je lakši na oko, Swift je dosta komplikovaniji i teži za učenje od Objective-C-a .

Da li je Objective-C sličan Javi?

Java nasleđuje sintaksu iz C++, dok je Objective-C baziran na programskom jeziku C. Postoje ekstenzije dostupne za Apple jezik, koju su slične sintaksi jezika Smalltalk.

Da li je Objective-C isto što i C++?

Nije, evo par bitnih razlika:

Dizajn: C++ objektno-orijentisan i proceduralan, Objective-C objektno-orijentisan sa karakteristikama poruka jezika Smalltalk

Nasleđivanje: C++ podržava višestruko nasleđivanje, Objective-C ne podržava višestruko nasleđivanje

Paradigma: C++ koristi metode za poziv već implementiranih funkcionalnosti, Objective-C koristi poruke za komunikaciju, slično Smalltalk-u

Klase: C++ Strukture i klase su isto u C++, Objective-C Strukture i klase se tretiraju drugačije

Tip: C++ Statički, Objective-C Dinamički

Licenca: C++ open source project 2.0, Objective-C General Public License

Boolean operatori: C++ koristi true, false and bool, Objective-C koristi yes, no and bool

Templates: C++ STL (Standard Template Library), Objective-C nema ih

7 Zaključak

U ovom tekstu ukratko su predstavljene osnove razvoja programskog jezika *Objective-C*. Prikazani su jezici koji su imali uticaj na njegov nastanak i razvoj, jezici nastali posle njega kao i njegovo elementarno razvojno stablo.

Literatura

- [1] GNUstep: Introduction. <http://www.gnustep.org/information/aboutGNUstep.html>.
- [2] Skupina Java knjiga dostupnih za slobodno preuzimanje. <http://www.computer-books.us/java.php>.
- [3] Apple. Zvanična stranica kompanije Apple. <https://developer.apple.com/library>.
- [4] Federico Biancuzzi. *Masterminds of Programming: Conversations with the Creators of Major Programming Languages*. O'Reilly Media, 2009.
- [5] Brad Cox. The object oriented pre-compiler: programming smalltalk 80 methods in c language. January 1983.
- [6] Adele Goldberg. *Smalltalk-80: The Interactive Programming Environment*. Addison-Wesley, 1983.
- [7] Groovy. Zvaničan sajt. <https://web.archive.org/web/20140302111159/http://groovy.codehaus.org/>.
- [8] Kevin Hoffman. *Sams Teach Yourself Mac OS X Lion App Development in 24 Hours*. Sams Publishing, 2012.
- [9] Alan C. Kay. *The Early History of Smalltalk*. 1993.
- [10] Objective-C. Tutorijal za programski jezik Objective-C. https://www.tutorialspoint.com/objective_c/.
- [11] Objective-C. Zvanična stranica kompanije Apple. <https://developer.apple.com/documentation>.
- [12] Tim Parker. *Unix Unleashes*. Sams Publishing, 1998.
- [13] Brian W. Kernighan; Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, 1978.
- [14] Amit Singh. A brief history of mac os x. December 2003.
- [15] Swift. Zvanična sajt. <https://www.swift.com/>.
- [16] Swift. Zvanična stranica kompanije Apple. <https://developer.apple.com/swift/>.