

Razvoj programskog jezika Ruby

Seminarski rad u okviru kursa
Dizajn programskih jezika
Matematički fakultet

Dejana Mandić
dejanamandic2@gmail.com

17. decembar 2019

Sažetak

U ovom radu se prikazuje razvoj programskog jezika **Ruby**, i pominju se programski jezici koji su doprineli tom razvoju, kao što su: Perl, Smalltalk, Eiffel, Ada, Basic i Lisp. Objasnjen je uticaj navadenih programskih jezika na Ruby i prikazano je razvojno stablo koje slikovito prikazuje ovaj uticaj.

Sadržaj

1	Uvod	2
2	Osnovno o programskom jeziku Ruby	2
3	Razvojno stablo	4
3.1	Perl	4
3.2	Smalltalk	4
3.3	Eiffel	5
3.4	Ada	5
3.5	Lisp	5
4	Zaključak	6
	Literatura	6

1 Uvod

U današnje vreme ubrzanog tehnološkog razvoja veliki značaj se pridaje razvoju novih programskih jezika koji bi doprineli lakšem, prirodnijem i efikasnijem programiranju. Akcenat se stavlja na čitljivost napisanog programa radi jednostavnijeg timskog razvoja programa ili aplikacije, što je uglavnom slučaj u praksi. U poslednje tri decenije razvija se ogroman broj programskih jezika sa idejom da budu što razumljiviji običnom čoveku. Programski jezik Ruby nastaje kao kombinacija imperativne, funkcionalne i objektno-orijentisane paradigme u pokušaju da se napravi što prirodniji jezik za skriptno programiranje. On postaje popularan radi svoje fleksibilnosti, prirodnosti programiranja zbog uticaja objektno-orijentisane paradigme, i jednostavnog povezivanja sa nekim od najkorišćenijih programskih jezika. Logo programskog jezika Ruby nalazi se na slici 1.



Slika 1: Logo programskog jezika Ruby

2 Osnovno o programskom jeziku Ruby

Ruby je viši programski jezik opšte namene koji se interpretira, što znači da se izvorni kod prevodi u kod razumljiv računaru prilikom svakog izvršavanja programa. Prva verzija je objavljena **1995.** godine u Japanu, od strane čoveka po imenu Yukihiro "Matz" Matsumoto. Nastao je kao sporedni projekat nekoliko kolega, koji su kasnije i nastavili njegov razvoj. Nije postojala specijalna potreba za njegovim nastankom u tom trenutku, prema rečima kreatora programskog jezika, nastao je sa svrhom da programeri koji u njemu kucaju uživaju u programiranju.¹ Zbog toga, Ruby ima elegantnu sintaksu koja je prirodna za čitanje i lako pisanje. Primer Hello World programa u programskom jeziku Ruby:

```
1000 puts "Hello World"
```

Listing 1: Ruby - Hello World

Ruby je dinamički tipiziran jezik koji koristi sistem za automatsko upravljanje memorijom tj. prikupljanje smeća (eng. *garbage collector*) i prati princip najmanje zabune (eng. *least astonishment*) [5] [7]

¹ "I hope to see Ruby help every programmer in the world to be productive, and to enjoy programming, and to be happy. That is the primary purpose of Ruby language.", Yukihiro Matsumoto [5]

Karakteristike programskog jezika Ruby:

- ekstremna objektna orijentisanost (sve je objekat, čak i broj se smatra objektom (listing 2) i lako se proširuju i osnovni tipovi)
- pačije kucanje (eng. *duck typing*)²
- laka refleksija i aktivno metaprogramiranje
- sistem za prikupljanje smeća (eng. *garbage collector*)
- sve je true osim false i null/nil
- postoji obrada izuzetaka (try-catch princip)
- moguće je predefinisati operatore
- podržava rad sa nitima na svim platformama

```
1000 5.times { print "This will be printed 5 times." }
```

Listing 2: Ruby - Broj kao objekat

Osnova jezika su anonimne funkcije i one se mogu vezati za bilo koji objekat, često se nazivaju i blokovima. Blok je kod koji se nalazi između do i end (slika 2). Primer ovakvog bloka:

```
1000 search_engines =  
      %w[Google Yahoo MSN].map do |engine|  
1002   "http://www." + engine.downcase + ".com"  
      end
```

Listing 3: Ruby: do-end

Što se tiče objektno-orijentisanih principa: moguće je naslediti samo jedan objekat i ne postoje interfejsi. Ruby uvodi koncept modula koji predstavlja kolekciju određenih metoda. Klasa može da dodaje više modula i da praktično doda te metode sebi, ovo se pokazalo kao rešenje koje dovodi do manje zabune nego višestruko nasleđivanje.

```
1000 class MyArray  
      include Enumerable  
1002 end
```

Listing 4: Ruby - primer sa modulom Enumerable

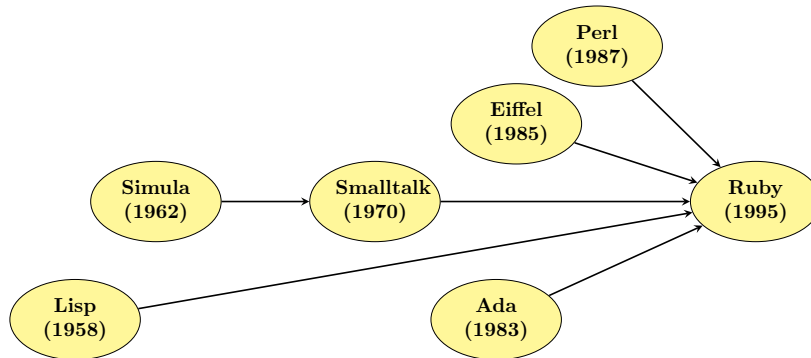
U ovom primeru ukoliko je u klasi implementirana metoda **each** može da se uključi modul Enumerable, koji će klasi dodati gomilu već implementiranih metoda koje koriste each za iteraciju.

²Duck typing - pojam koji označava da je jasno šta se dešava u kodu iz onoga što piše. - "If it walks like a duck and it quacks like a duck, then it must be a duck"

3 Razvojno stablo

Na nastanak jezika Ruby je uticalo nekoliko jezika i svaki od njih je zaslužan za neke bitne koncepte samog jezika. Jezici koji su uticali na programski jezik Ruby su: **Perl** (verzija 4 a ne kasnije), **Smalltalk**, **Eiffel**, **Ada**, **Lisp**.^[7]

Razvojno stablo jezika Ruby može se videti na slici 2.



Slika 2: Razvojno stablo jezika Elixir

3.1 Perl

Prvobitna ideja nastanka Ruby-ja je bila napraviti objektno-orijentisanki skriptni jezik. **Perl** je kao takav već postojao, ali Yukihiro Matsumoto je smatrao da taj jezik ima vrlo usku primenu i ne može se primeniti na veliki broj problema. Larry Wall je počeo da radi na programskom jeziku Perl **1987.** dok je radio kao programer u Unisys-u, i objavio je verziju 1.0 u decembru iste godine. Nastao je kao skriptni jezik opšte namene koji bi olakšao pravljenje izveštaja. Dobijao je razne nadimke, zbog svoje moći i fleksibilnosti ali i ružnog izgleda koda. Jezik se razvijao ogromnom brzinom pa je nekoliko godina kasnije (1991.) objavljena verzija 4.0 i ta verzija je poslužila kao uzor za nastanak programskog jezika Ruby. Ta verzija, kao i sve prethodne su zapravo imale dosta nedostataka i ogroman potencijal za dalju nadogradnju, za razliku od kasnijih verzija 5 i 6 koje se danas koriste, i koje su dovele Perl do svog trenutnog maksimuma popularnosti. ^[6]

3.2 Smalltalk

Smalltalk je jezik koji je nastao **1970.** godine u edukativne svrhe od strane grupe istraživača od kojih su neki: Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Diana Merry, Scott Wallace itd. Pripada objektno-orijentisanoj paradigmi i nastao je pod uticajem programskog jezika **Simula**³, ali je sam Smalltalk kasnije postao jedan od najuticajnijih programskih jezika sa uticajem na jezike kao što su: Flavors, CLOS, Objective-C, Java, Python, **Ruby**. Objekat u

³Simula je prvi objektno orijentisani programski jezik, nastao 60ih godina. Sa njim su se pojavili danas vrlo poznati koncepti, kao što su klasa, objekat, nasleđivanje i automatsko oslobađanje memorije.

Smalltalk-u predstavlja instancu klase i može da radi neku od sledeće 3 stvari:

- drži stanje (reference ka drugim objektima)
- prima poruke od sebe ili drugih objekata
- u svrhe obrade poruke, šalje poruke sebi i ostalim objektima

Smalltalk odlikuje **ekstremna objektna orijentisanost** tj. sve se smatra objektima, čak i primitivni tipovi. [2] Ovaj objektni pristup programiranju, i svim podacima koji su nam potrebni je ono gde se primećuje uticaj ovog programskog jezika na Ruby, ali takođe, Smalltalk je prvi jezik koji uvodi duck typing kao pojam. [3]

3.3 Eiffel

Eiffel je jezik koji pripada objektno-orijentisanoj paradigmi koji je dizajnirao Bertrand Meyer 1985. godine sa ciljem da poveća pouzdanost softvera za komercijalnu upotrebu. Kasnije, 2005. ovo je postao ISO-standardizovan jezik. Karakteristike ovog jezika su: statičko tipiziranje, višestruko nasleđivanje, generičko programiranje, **automatsko upravljanje memorijom** (koristi garbage collector), podržava **konkurentno objektno-orijentisano programiranje** na jednostavnom nivou itd. [4]

Ono što se primećuje u poređenju da programskim jezikom Ruby jeste da je objektno-orijentisani koncept razrađen na drugačiji način tj. nemamo višestruko nasleđivanje i tipizacija je dinamička ali ono što je Ruby nasledio je automatsko upravljanje memorijom i ideja o konkurentnom pristupu objektno-orijentisanom programiranju.

3.4 Ada

Ada je struktuiran, statički tipiziran, imperativni i objektno-orijentisani programski jezik koji je dizajnirao ceo tim ljudi vođen francuskim naučnikom po imenu Jean Ichbiah. Dizajniran je od **1977.** do **1983.** godine i svoje ime je dobio po Adi Lovelace koja se smatra prvim programerom. Glavni cilj ovog programskog jezika je bila primena u ugrađenim sistemima i sistemima koji funkcionišu u realnom vremenu. Osnovne karakteristike ovog programskog jezika su: modularnost (organizacija po paketima), provere u realnom vremenu, zaštićeni objekti (protected), **obrada izuzetaka** itd. Takođe, koristi neke ključne reči kao što su **begin** i **end** koje označavaju početak i kraj nekog bloka koda. [1] Ovaj blok koda je analogon Ruby-jevom do-end bloku i ono što je jasno preuzeto iz ovog programskog jezika je obrada izuzetaka za koju znamo da postoji u Ruby-ju u vidu try-catch bloka.

3.5 Lisp

Lisp je programski jezik zasnovan na matematičkoj teoriji rekurzivnih funkcija (u kojoj se funkcija pojavljuje u sopstvenoj definiciji), a Lisp program je funkcija koja se primenjuje na podatke. Lisp je jedan od najstarijih programskih jezika nastao **1958.** godine. Danas, pod pojmom Lisp podrazumevamo familiju programskih jezika koji su nastali kao unapređenje prvobitne verzije iz 1958. godine. Nastao je na univerzitetu MIT od strane John McCarthy-ja, i njegova ideja je bila da napravi kompletan programski jezik za algoritme. Lisp uvodi strukture

podataka, **dinamičku tipizaciju**, automatsko upravljenje memorijom, uslove, rekurziju, samoodržavajući kompilator itd. [2]

Ono što je jasno da je sigurno Ruby nasledio odavde je dinamička tipizacija i koncept automatskog upravljanja memorijom.

4 Zaključak

U ovom radu ukratko su predstavljene osnove razvoja programskog jezika Ruby, jezici koji su najviše uticali na njegov nastanak i razvoj, kao i razvojno stablo sa tim jezicima. Za dodatne informacije i praćenje daljeg razvoja, najbolje je koristiti zvaničnu stranu programskog jezika Ruby [8].

Literatura

- [1] AdaCore. Zvanična stranica programskog jezika Ada.
<https://www.adacore.com/about-ada>.
- [2] Grupa autora. *Velika opšta enciklopedija 1-5*. Larousse, 2005.
- [3] Howard I. Cannon. *A non-hierarchical approach to object-oriented programming*. MIT, 2003.
- [4] Eiffel. Zvanična stranica programskog jezika Eiffel.
<https://www.eiffel.org/welcome>.
- [5] Yukihiro Matsumoto. *The Ruby Programming Language*. O'Reilly Media, 2008.
- [6] Perl. Zvanična stranica programskog jezika Perl.
<https://www.perl.org>.
- [7] Ruby. Uvod u Ruby. <https://www.ruby-lang.org/en/about/>.
- [8] Ruby. Zvanična stranica programskog jezika Ruby.
<https://www.ruby-lang.org/en/>.