

Razvoj programskog jezika Groovy

Seminarski rad u okviru kursa Dizajn programskih jezika
Matematički fakultet

Irena Maričić
maricicirena@yahoo.com

21. decembar 2019.

Sadržaj

1	Uvod	3
2	Razvojno stablo	3
2.1	Java	3
2.1.1	Karakteristike	4
2.2	Python	5
2.3	Ruby	5
2.4	Smalltalk	6
3	Zaključak	6
4	Literatura	7

1 Uvod



Slika 1: Logo programskog jezika Groovy

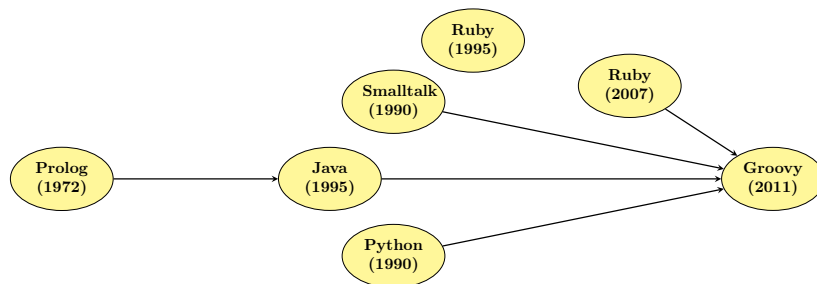
Groovy je objektno-orijentisan programski jezik opšte namene. *Groovy* programski jezik je kreiran za Java platformu i može se koristiti kao alternativa Java programskom jeziku. Sintaksa *Groovy* jezika liči na sintaksu *Java* programskog jezika. Štaviše, veliki deo koda koji je pisan u *Java* programskom jeziku sintaksno je ispravan *Groovy* kod. Programi pisan u *Groovy* jeziku mogu da se povežu sa *Java* kodom i da koriste *Java* biblioteke. Kada je potrebno da se *Groovy* kod izvrši, dinamički se prevodi u *Java* bajt kod. *Groovy* sadrži i osobine koje izdvajaju programske jezike kao što su **Python**, **Ruby** i **Smalltalk**. Na primer poseduje i dinamičku proveru tipova podataka. *Groovy* može da se koristi za pisanje samostalnih programa, za pisanje delova *Java* projekata ali i kao skript jezik. *Groovy* 1.0 je objavljen 2. Januara 2007 godine, a *Groovy* 2.0 u Julu 2012 godine. Počevši od verzije 2.0, *Groovy* se može kompajlirati statički. *Groovy* 2.4 je poslednje veliko izdanje pod sponzorstvom Pivotal Software, koje je završeno ju Martu 2015. godine. Najnovija stabilna verzija je *Groovy* 2.5.8. *Groovy* je svoju dotadašnju upravljačku strukturu promenio u odbor za upravljanje projektima u Apache Software Fondaciji. Većina važećih *Java* datoteka su takodje važeće *Groovy* datoteke. Iako su ova dva jezika jako sićna, *Groovy* može biti kompaktniji, jer mu nisu potrebni svi elementi koji su potrebni u *Java* programima. *Groovy* funkcije koje nisu dostupne u *Javi* kao što su: preopterećenje operatora, nativna sintaksa za liste i asocijativne nizove tj mape, automatsko proveravanje ima li *NULL* pokazivača i mnoge dr. se mogu definisati sa ključnom rečju *def*. *Groovy* pruža podršku za razne jezike za označavanje kao što su *XML* i *HTML*, koja se ostvaruje pomoću sinteze DOM (Document Object Model). Ova karakteristika omogućava definisanje i manipulaciju mnogih vrsta podataka sa ujednačenom i sažetom sintaksom i programskom metodologijom. Za razliku od *Jave*, *Groovy* datoteka izvornog koda se može izvršiti kao (nekompileirani) skript pod sledećim uslovima: ako sadrži kod izvan bilo koje definicije klase, ako je klasa sa glavnim metodom (*main*) ili ako je *Runnable* ili *GroovyTestCase*. *Groovy* skript je u potpunosti raščlanjen, kompajliran i generisan pre izvršavanja, slično kao i *Python* i *Ruby* skripte.

2 Razvojno stablo

Na nastanak jezika *Groovy* najviše je uticao programski jezik *Java*. Takodje veliki uticaj je imao i programski jezici *Ruby*, *Python* i *Smalltalk*. Razvojno stablo *Groovy* jezika može se videti na slici 2.

2.1 Java

Java je objektno-orijentisani programski jezik koji je razvila kompanija Sun Microsystems početkom 1990-ih godina. Mnogi koncepti *Jave* su zasnovani na



Slika 2: Razvojno stablo jezika Groovy

jeziku Oberon. Izbacili su koncept modula i uveli pakete kakve danas znamo, koji se oslanjaju na fajl sistem, takodje su uveli formalno koncept klasa iz objektno-orijentisane paradigme. Osim toga, jezik ima sličnu sintaksu kao jezici C i C++, ali je mnogo stroži pri prevodjenju. Java je dizajnirana tako da bude nezavisna od platforme, vodjena motom: "Pisi jednom prevodi svuda.". Prva verzija je zvanično objavljena 1995. godine. Java je uz Kotlin zvanično podržan jezik za Android uređjaje. U stvaranju Java jezika oslanjalo se na pet osnovnih cijeva:

1. Mora biti jednostavan, objektno-orijentisan i prepoznatljiv
2. Mor biti temeljan i siguran
3. Mora biti arhitektički neutralan i prenosiv
4. Mora da se izvršava sa visokim performansama
5. Mora biti interpretiran, praćen i dinamičan

2.1.1 Karakteristike

- jednostavan
- objektno-orijentisan
- distribuiran
- robustan
- bezbedan
- neutralan
- prenosiv
- interpretiran
- performantan
- visenitan
- dinamičan

Java platforma tj. Java okruženje u kom se program izvršava, dizajnirana je tako što manje zavisi od specifičnih karakteristika konkretnog računarskog sistema. Java se izvršava korišćenjem Java virtuelne mašine - JVM. Java je jezik koji se prevodi i interpretira. Java izvorni kod nakon kompajliranja, kompajlerom javac, se prevodi u izvršni kod bajtkod. Formiran bajtkod je isti za sve arhitekture računara i predstavlja instrukcije za Java virtuelnu mašinu. Bajtkod u Java virtuelnoj mašini, java interpretatorom - java, se dešifruje i interpretira u izvršni kod, pogodan za izvršavanje na računaru, odgovarajućem hardveru.

U zavisnosti od JVM, za isti izvorni kod možemo dobiti različite izvršne kodove, za različite platforme. Java izvorni kod i bajtkod ostaju isti bez obzira na kojoj se platformi izvršava program. Javina prenosivost međjutim, izaziva gubitak performansi. To je zato što se tek prilikom interpretiranja bajtkod prevodi u mašinski za konkretnu platformu. ovo može biti problem ako se isti kod više puta interpretira na istoj platformi.

2.2 Python

Python je programski jezik visokog nivoa opšte namene. Nastao je početkom devedesetih godina prošlog veka. Njegov autor je Gvido van Rosum. Značajan uticaj na koncept ovog jezika imao je autorov rad na kreiranju programskog jezika **ABC**. Podržava, u prvom redu imperativni, objektno-orijentisan i funkcionalni stil programiranja. Sintaksa jezika Python omogućava pisanje veoma preglednih programa. Jezik se brzo i lako uči. Programi pisani u Python jeziku se najčešće interpretiraju. Uz interpretator se obično isporučuje i veoma razvijena standardna biblioteka modula. Podržava više paradigmi programiranja uključujući proceduralno, objektno-orijentisano i funkcionalno programiranje. Podaci u programskom jeziku Pajton su predstavljeni objektima. Tip podatka u programskom jeziku Pajton nije vezan za promenljivu. Svako promenljivoj tokom izvršavanja programa može da bude dodeljena vrednost bilo kog tipa kao i da ta vrednost bude zamenjena drugom različitog tipa. Svaki objekat sadrži tip objekta i njegovu vrednost. Jednom kreirani objekat ne može da menja tip, dok vrednost nekih objekata može da bude promenjena.

2.3 Ruby

Ruby je objektno orijentisani programski jezik. U sebi kombinuje sintaksu inspirisanu jezicima Perl i Ada, sa objektno orijentisanim osobinama nalik jeziku Smalltalk, a deli i neke osobine sa jezicima Python, Lisp, Dylan i CLU. Ruby je jednoprolazni interpretirani jezik. Njegova glavna implementacija je slobodni softver pod licencom otvorenog koda. Yukihiro Matsumoto je sa razvojem ovog jezika počeo u februaru 1993. godine. Prvi put je objavljen 1995. godine, a trenutno je aktuelna stabilna verzija 1.9.3-p0 koja se može preuzeti sa zvanične strane jezika. Po mnogim programerima, Ruby sledi princip „najmanjeg iznenađenja“, čime misle da je taj jezik oslobođen svih zamki i kontradiktornosti poznatih iz drugih jezika. Ime „rubi“ potiče od engleske riječi za „rubin“ (engl. Ruby) što je samo aluzija na „perl“ (engleski „biser“). Matsumoto je rekao da je Ruby dizajniran za produktivnost programera i zabavu, na načelima dobrog dizajna korisničkog interfejsa. Na Google Tech Talk-u, 2008. godine Matsumoto dalje navodi: „Nadam se da će Ruby pomoći svakom programeru u svetu da bi bude produktivniji, i da uživa u programiranju, i da bude srećniji. To je primar-

na svrha jezika Rubi. “ On naglašava da dizajn sistema treba da pre zadovolji potrebe čoveka, a ne računara. Neke od osobina Ruby jezika su:

- jednostavna i čitljiva sintaksa
- netipizovane varijable
- obrada izuzetaka
- automatsko oslobadjanje nepotrebno zauzete memorije
- podrška na više operativnih sistema
- automatska dokumentacija

2.4 Smalltalk

Smalltalk je objektno-orijentisani, dinamički, reflektivni programski jezik. Jedan je od mnogih objektno-orijentisanih jezika baziranih na programskom jeziku Simula. Rezultat je višegodišnjeg proučavanja softvera koji je pogodan za pravljenje visoko-funkcionalne i interaktivne komunikacije sa operativnim sistemima personalnih računara. Jedan njegov deo je stvoren za obrazovne svrhe, ali se uglavnom koristi za konstruktivno učenje. Razvila ga je grupa istraživača iz kompanije Xerox PARC na čelu sa Alan Kay-om. Smalltalk je čist primer objektno-orijentisanog programiranja. Zbog svoje izričitosti po tom pitanju, uočavaju se bitne razlike u odnosu na mejnstrim predstavnike ove paradigme. On ne pravi razliku između vrednosti primitivnih tipova (npr. boolean, int) i klasnih tipova. Svi primitivni tipovi podržani ovim programskim jezikom predstavljaju objekte njima odgovarajućih klasa. Kao takvi u mogućnosti su da ravnopravno, putem poruka, komuniciraju sa instancama drugih klasa. S obzirom na takvo poimanje primitivnih tipova, programer ima mogućnost da, u okviru njima odgovarajućih klasa, vrši izmene i dodaje određene funkcionalnosti ili menja postojeće. 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.

3 Zaključak

U ovom tekstu ukratko su predstavljene osnove razvoja programskog jezika Groovy, kao i jezika koji su uticali na njega: Ruby, Smalltalk, Java i Python. Za više informacija posetite zvanični sajt programskog jezika Groovy.

4 Literatura

- [1] [Wikipedija-Groovy](#)
- [2] [Wikipedija-Java](#)
- [3] [Wikipedija-Python](#)
- [4] [Wikipedija-Ruby](#)
- [5] [Wikipedija-Smalltalk](#)