



Razvoj programskog jezika Java

Seminarski rad u okviru kursa

Dizajn programskih jezika

Matematički fakultet

Marija Nikoloska 407/2017

marinikoloska@gmail.com

5. Decembar 2019

Sažetak

Ovaj rad će se posvetiti istorijskom razvoju programskog jezika Java i jezicima koji su značajno uticali na njegov razvoj, kao i jezici na koje je jezik Java imao veliki uticaj. Pored osnovnih informacija o svakom od ovih jezika i načina na koji su uticali na razvoj jezika Java, takođe će biti prikazano i razvojno stablo koje uključuje ove jezike. Kao zaključak, biće opisano šta je bio cilj kreatora programskog jezika Java u vidu zahteva koji je tada kreirani jezik trebao da ispunjava, kao i karakteristike koje su se u međuvremenu izmenile.

Sadržaj

1	Uvod: razvoj OOP	2
2	Osnovno o Javi	2
2.1	Logo jezika Java	2
2.2	Osnovni koncepti	2
3	Razvojno stablo	4
3.1	C++	4
3.2	Groovy	5
3.3	Scala	5
3.4	Clojure	6
3.5	Kotlin	6
4	Zaključak	7
	Literatura	8

1 Uvod: razvoj OOP¹

Objektno orijentisano programiranje se javilo kao odgovor na softversku krizu koja je zadesila programere 1960-ih godina. Naime, pojavom računara treće generacije i naglim porastom snage računara stvorila se potreba za složenijim softverom [6]. Postojeće metode programiranja su bile neadekvatne: dotadašnje tehnike individualnog programiranja se nisu mogle efikasno skalirati na velike programe. Softver je postao bagovit i skup, ne samo za razvoj, nego i za održavanje. Pokrenuti razvojni projekti višestruko su premašivali planirane troškove i rokove jer je bilo potrebno angažovati veliki broj ljudi za pisanje jednog projekta.

Još neki od problema koji su se javili:

- softver je niskog kvaliteta
- softver je nedovoljno efikasan
- softver troši previše memorije
- kod je izuzetno teško održavati
- teško je dodavati funkcionalnosti

Objektno-orijentisano programiranje nastaje kao novi stil programiranja koji u velikoj meri rešava navedene probleme. [7] OOP uvodi svojevrsnu revoluciju u načinu izrade softvera u čijem primarnom fokusu se nalazi objekat².

Programski jezik Java je jedan od predstavnika objektno-orijentisane paradigme koji se koristi za izradnju i razvoj velikog broja aplikacionih softvera. Java programiranje je jedno od najzastupljenijih danas, posebno imajući u vidu sve veću primenu u izradi aplikacija za mobilne telefone i tablet uređaje kroz Android operativni sistem. Java platforme imaju široku upotrebu u programiranju i primenjenim softverskim rešenjima, koje se kreću od najjednostavnijih digitalnih uređaja, mp3 plejera, mobilnih telefona, pa sve do kompleksnih web servera i korporativnih aplikacija.

2 Osnovno o Javi

2.1 Logo jezika Java

Programski jezik Java nastao je **1991.** godine kao projekat firme **Sun Microsystems**. Jezik je prvobitno nazvan **Hrast** (engl. *Oak*), i bio je namenjen prvenstveno za potrebe interaktivne televizije. Međutim infrastruktura digitalne televizije u tom periodu još nije bila dovoljno uznapredovala da bi se to ostvarilo. Ime projekta Hrast je kasnije promenjeno u Java, po brendu Java kafe (engl. *Java Coffee*), koja je dobila ime po istoimenom ostrvu Java, i zbog toga logo jezika java prikazuje šolju vruće kafe. [10] Jezik Java je **1995.** godine pušten na tržište kao univerzalni programski jezik. Danas je Java vlasništvo kompanije **Oracle**. Logo jezika Java prikazan je na slici 1.

2.2 Osnovni koncepti

Mnogi koncepti Jave su zasnovani na osnovnim konceptima objektno-orijentisane paradigme. Izbačen je tada široko korišćen koncept modula

¹Objektno-orijentisana paradigma, Objektno-orijentisano programiranje

²Integralna celina atributa (podataka) i metoda (procedura za rad sa podacima)



Slika 1: Logo programskog jezika Java

i uvedeni su paketi koji se oslanjaju na fajl sistem. Uveden je i formalni koncept klasa iz objektno-orijentisane paradigme. Osim toga, jezik Java ima sintaksu slučnu jezicima C i C++, ali je dizajniran da bude nezavisan od platforme i sa pojednostavljenim upravljanjem memorijom.

Autori Jave objavili su dokument koji objašnjava dizajn, ciljeve i postignuća Jave. Između ostalog, objavljena je i kratka lista osnovnih principa na kojima se Java zasniva i zahteva koje je jezik morao da ispunjava³:

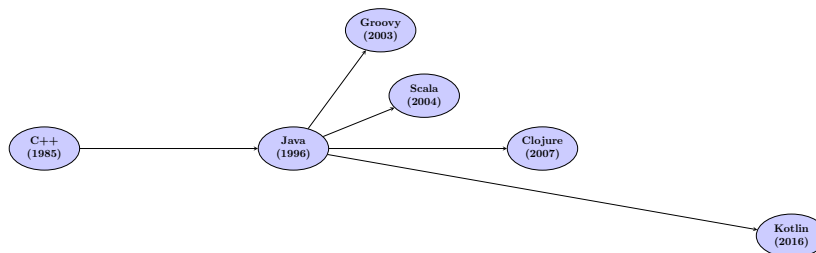
- **Mora biti jednostavan i familijaran**
Sintaksa jave se zasniva na C sintaksi i objektno orijentisanim karakteristikama jezika C++. Za razliku od programskog jezika C i C++, u Javi nema pokazivača, struktura, unija, itd., a samim tim ni eksplicitnog baratanja memorijom od strane programera. To naravno ima i svoje loše strane - često se koristi više memorije nego što je zaista potrebno, što može da utiče na performanse Java programa.
- **Mora biti objektno-orijentisan**
Java, zajedno sa C++ je objektno-orijentisan jezik od samog početka. Razlika između C++ i Jave iz perspektive objektno-orijentisane paradigme je u tome što Java ne podržava višestruko nasleđivanje, ali može da ga simulira korišćenjem interfejsa.
- **Mora biti temeljan i siguran**
Java omogućava kreiranje veoma pouzdanog softvera: insistira na intenzivnoj proveru u veoma ranim fazama razvoja softvera, tokom same kompilacije, kao i tokom izvršavanja programa. Kao posledica toga što je koncept pokazivača izostavljen, model upravljanja memorijom je jednostavan i potpuno eliminiše mogućnost gubljenja i korupcije podataka.

³Lista ovih zahteva, kao i njihovi kratki opisi su preuzeti iz prezentacije dr Aleksandra Kartelja i prof. dr Vladimira Filipovića napravljene za potrebe kursa Objektno-orijentisano programiranje [1].

- **Mora biti arhitektonski neutralan i prenosiv**
Slogan programskog jezika Java je “piši jednom, izvršavaj bilo gde” (engl. *write once, run everywhere*). Java kompajler prilikom kompilacije najpre generiše takozvani *byte* kod (.class fajl) koji je vrlo sličan mašinskom kodu, ali je za razliku od mašinskog koda arhitektonski neutralan, tj. ne zavisi od procesora, pa je samim tim i prenosiv na različite vrste procesora i operativnih sistema. Ovaj .class fajl se može pokrenuti na bilo kojoj platformi uz pomoć JVM (engl. *Java Virtual Machine*) predviđene za tu platformu.
- **Mora biti performantan i interpretiran**
Java kod se kompajlira do bajt-koda, a potom se interpretira. Interpreter radi punom brzinom jer su sigurnosne provere obavljene u ranijim fazama kompilacije. Takođe postoji automatski sakupljač otpadaka (engl. *garbage collector*) koji neutralizuje posledice nemogućnosti eksplicitnog manipulisanja memorijom. Ovaj proces radi nezavisno od pokrenutih programa i samostalno odlučuje koje objekte će ukloniti iz memorije. Osim uklanjanja suvišnih objekata, on vrši i defragmentaciju memorije kada je to potrebno.
- **Mora biti višenitno i dinamičan**
Java podržava višenitno izvršavanje i dinamički učitava klase u toku izvršavanja, što znači da se klase povezuju (engl. *link*) samo kada je to potrebno.

3 Razvojno stablo

Na slici 3 se može videti razvojno stablo jezika Java. Kao što iz razvojnog stabla može uočiti, na nastanak jezika Java je najviše uticao programski jezik C++, dok je on sam uticao na razvoj velikog broja jezika, pretežno zbog svoje portabilnosti i dobre platforme.



Slika 2: Razvojno stablo jezika Java

3.1 C++

C++ je univerzalni imperativni programski jezik kompajlerskog tipa koji kombinuje jezik C sa objektno orijentisanom paradigmom. Inicijalno dizajniran sa naglaskom na sistemskom programiranju, on zadržava osobine jezika C kao što su **direktno adresiranje memorije**, što ga čini pogodnim za programe u kojima su performanse izvršavanja kritično bitne (kao što su embeded sistemi i multimedijalne aplikacije).

C++ je prvobitno razvijen **1979.** godine u Belovim laboratorijama za objektno-orijentisano programiranje u projektu pod rukovodstvom **Bjar-**

nea Stroustrup (engl. *Bjarne Stroustrup*). Zaključivši da je **Simula**⁴ imala veoma dobre karakteristike koje se mogu koristiti za razvoj velikog softvera, ali je bio prespor za korišćenje u praksi, dok je **BCPL** (engl. *Basic Combined Programming Language*) bio dovoljno brz, ali previše niskog nivoa za potrebe razvoja velikih programa, javila se motivacija da se napravi jezik sličan C-u⁵ sa objektno-orijentisanim funkcionalnostima jezika Simula. C++ je stoga nastao kao **proširenje programskog jezika C**, pa mu je originalno ime bilo "**C sa klasama**" (engl. *C with classes*)^[2].

Jezik "C sa klasama" je 1983. godine u duhu programiranja preimenovan u C++, jer operator za inkrementaciju (++) simbolizuje napredak (u odnosu na C), i prvi put je zvanično pušten na tržište **1985.** godine.

C++ je danas neosporno jedan od najmoćnijih, ali i najkompleksnijih jezika. Zbog kompaktnosti programa, brzine izvršavanja i prenosivosti, C++ zauzima dominantno mesto u svetu profesionalnog programiranja.

3.2 Groovy

Apache Groovy je java-sintaksno kompatibilan, objektno-orijentisan programski jezik. Može da se koristi kao programski jezik opšte namene, ali i kao skript alternativa programskog jezika Java. Sintaksa Groovy jezika liči na sintaksu programskog jezika Java - štaviše, veliki deo koda koji je pisan u Java programskom jeziku je sintaksno ispravan Groovy kod ^[4]. Na slici 3 se nalaze delovi koda napisani u jeziku Java i jeziku Groovy.

Od samog početka je cilj jezika Groovy bio da za Java developere obezbedi **dinamički tipiziran skript jezik**⁶ koji se se prevodi do JVM bajt-koda i između ostalog je kompatibilan sa već postojećim Java bibliotekama. Većina važećih Java kodova su takođe važeći kodovi u jeziku Groovy. Ipak, kod napisan u jeziku Groovy su uglavnom kompaktniji jer on ne zahteva sve elemente koji su potrebni Java programu, pa ne važi uvek obrnuto. Ovo omogućava Java programerima da postepeno uče Groovy počevši od poznate Java sintakse.

Groovy funkcionalnosti koje nisu dostupne u Javi uključuju statičko i dinamičko tipiziranje (uz korišćenje ključne reči **def**), preopterećenje operatora, ugrađenu podršku za liste i asocijativne nizove (mape), ugrađenu podršku za regularne izrace, polimorfnu iteraciju, interpolaciju niza, itd. Za razliku od jave, Groovy izvorni kod može biti izvršen kao nekompajliran skript koji se "ispod haube" (engl. *"under the hood"*) u potpunosti parsira, kompajlira a zatim generiše pre samog izvršavanja (slično jeziku Python i Ruby).

3.3 Scala

Programski jezik Scala kombinuje objektno orijentisano i funkcionalno programiranje u jedan koncizan jezik visokog nivoa ^[8]. Scala je **strogo statički tipiziran**, što omogućava da se lakše izbegnu greške u kompleksnim programima. Scala izvorni kod je predviđen da bude kompajliran do Java bajt-koda, tako da rezultujući program može da se izvršava na Java virtuelnoj mašini. Java biblioteke mogu da se koriste direktno u Scala kodu i obrnuto, što omogućava interportabilnost jezika.

⁴Simula se smatra prvim predstavnikom objektno-orijentisane paradigme

⁵Jezik C je izabran kao baza jer je bio generalno orijentisan, brz i u širokoj upotrebi

⁶Ove reči je kreator Groovy jezika upotrebio u svom blogu *Groovy - the birth of a new dynamic language for the Java platform* ^[9]

```

300 public class HelloWorld {
301     String name;
302
303     public void setName(String name) {
304         this.name = name;
305     }
306
307     public String getName() {
308         return name;
309     }
310
311     public String greet() {
312         return "Hello " + name;
313     }
314
315     public static void main(String
316         args[]) {
317         HelloWorld helloWorld = new
318             HelloWorld();
319         helloWorld.setName("Groovy");
320         System.out.println(helloWorld.
321             greet());
322     }
323 }

```

Listing 1: Java

```

300 class HelloWorld {
301     String name
302     def greet() {
303         "Hello $name"
304     }
305
306     def helloWorld = new HelloWorld(
307         name: "Groovy")
308     println helloWorld.greet()
309 }

```

Listing 2: Groovy

Slika 3: Kod u jeziku Java i Groovy

Java je na razvoj jezika Scala uticala na vrlo interesantan način: sam dizajn jezika Scala je inspirisan nedostacima i manama jezika Java⁷

```

300 object HelloWorld extends App {
301     println("Hello World!") }

```

Listing 3: Scala

3.4 Clojure

Clojure je **dinamički tipiziran** programski jezik opšte namene koji kombinuje pristupačnost i interaktivni razvoj skript jezika sa efikasnom i robusnom infrastrukturom za višenitno programiranje. On je **kompajliran** jezik, ali je i dalje potpuno dinamički tipiziran - svaka funkcija koju Clojure podržava je podržana u toku izvršavanja[3]. Clojure predstavlja **dijalekt Lispa** koji obezbeđuje jednostavan pristup java frejmvorku (engl. *framework*) i izvršava se na Java virtuelnoj mašini. Rič Hiki (engl. *Rich Hickey*) je razvio Clojure **2007.** godine jer je želeo pretežno funkcionalan, moderan Lisp, dizajniran da podržava konkurentno programiranje, i koji bi imao simbiozu sa već utvrđenom Java platformom.

3.5 Kotlin

Kotlin je **statički tipiziran** programski jezik za razvoj multiplatformskih aplikacija razvijen je od strane kompanije **Džetbrejns** (engl. *Jetbrains*), sa sedištem u Sankt Petersburgu u Rusiji. Dobio je ime po ostrvu Kotlin u blizini Sankt Peterburga. Kotlin je koncizan, interoperabilan, bezbedan, sa odličnim razvojnim alatima i okruženjem. Radi na Java

⁷https://en.wikipedia.org/wiki/Criticism_of_Java

virtuelnoj mašini (JVM), može se kompajlirati u JavaScript kod ili koristiti LLVM kompajler. Iako sintaksa ovog jezika nije kompatibilna sa Java jezikom, Kotlin je dizajniran za potpunu interoperabilnost sa Java kodom [5]. Sam jezik je nastao u cilju prevazilaženja nedostataka koje je imao programski jezik Java. Dizajniran je da odgovori na zahteve industrije za novim programskim jezikom, "boljim" od Jave, ali opet potpuno interoperabilnim sa Java kodom. Na ovaj način se ostavlja mogućnost kompanijama da postepeno migriraju kod sa Jave na Kotlin.

Kotlin sa sobom donosi dosta novih funkcionalnosti i unapređenja, a kako kažu njegovi kreatori, pokupio je najbolje osobine nekih već žrelih jezika. Kotlin je odličan dodatak za već postojeći Android ekosistem. S obzirom na to da se kompajlira u Java bajt-kod i radi na Java virtuelni mašini, ovaj jezik je u potpunosti interoperabilan sa Java programskim jezikom. Ovo omogućava kombinovanje Kotlin koda sa već postojećim Java kodom. Pozivanje Kotlin koda iz postojećeg Java koda jednostavno funkcioniše bez greške. Takođe, pozivanje Java koda iz Kotlin koda takođe funkcioniše bez bilo kakve intervencije programera.

4 Zaključak

U ovom tekstu ukratko su predstavljene osnove programskog jezika Java. Jezik C++ je prikazan kao jezik koji je značajno uticao na njegov nastanak i razvoj, a uz to su prikazani i jezici na koje je Java značajno uticao: može se uočiti da je Java platforma uvela brojne pogodnosti koje su tvorci novijih jezika želeli da iskoriste, pa je zato veliki broj jezika i predviđen da se kompajlira do Java bajt-koda i kasnije izvršava na JVM. Pored osnovnih informacija o ovim jezicima, prikazano je i elementarno razvojno stablo jezika Java na slici 3.

Literatura

- [1] Aleksandar Kartelj, Vladimir Filipović. Presentacija za predmet objektno-orijentisano programiranje: Karakteristike i dizajn jezika java. <http://poincare.matf.bg.ac.rs/~kartelj/nastava/OOP2018/OOP01-Karakteristike%20i%20dizajn%20jezika%20Java.pdf>.
- [2] C++. Zvanična stranica programskog jezika C++: Istorija programskog jezika C++. <http://www.cplusplus.com/info/history/>.
- [3] Clojure. Zvanična stranica programskog jezika Clojure. <https://clojure.org/>.
- [4] Groovy. Zvanična stranica programskog jezika Groovy. <http://groovy-lang.org/>.
- [5] Kotlin. Zvanična stranica programskog jezika C++: Istorija programskog jezika Kotlin. <https://kotlinlang.org/>.
- [6] Robert Manger. *Softversko inženjerstvo*. <http://ttl.masfak.ni.ac.rs/SUK/Softversko%20inzenjestvo.pdf#page=7>.
- [7] Nemanja Mićović. Presentacija za predmet Objektno-orijentisano programiranje: Uvod u OOP. <https://matfoop.github.io/OOP/vezbe/prezentacije/03.oop.pdf>.
- [8] Scala. Zvanična stranica programskog jezika Scala. <https://www.scala-lang.org/>.
- [9] James Strachan. Groovy - the birth of a new dynamic language for the Java platform. <https://web.archive.org/web/20030901064404/http://radio.weblogs.com/0112098/2003/08/29.html>.
- [10] Wikipedia. Java (programming language). [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).