

Razvoj programskog jezika Kotlin

Seminarski rad u okviru kursa
Dizajn programskih jezika
Matematički fakultet

Sanja Milosavljević
mr16200@alas.matf.bg.ac.rs

10. decembar 2019.

Sažetak

U ovom radu biće prikazane osnove razvoja programskog jezika Kotlin. Opisani su jezici koji su najviše uticali na njegov razvoj. Za svaki jezik date su osnovne informacije kao i način na koji je taj jezik uticao na osobine i razvoj jezika Kotlin. Prikazano je razvojno stablo u koje su uključeni ovi jezici.

Sadržaj

1	Uvod	2
2	O Kotlinu	2
3	Razvojno stablo	3
3.1	Java	4
3.2	Scala	4
3.3	JavaScript	5
3.4	C#	5
3.5	Groovy	6
	Literatura	7

1 Uvod

Kotlin je novi general-purpose programski jezik koji se izvršava na Java virtuelnoj mašini i koji se sa svojim izvanrednim funkcionalnostima vrlo brzo istakao i postao popularan. Sam jezik je nastao u cilju prevazilaženja nedostataka koje je imao programski jezik Java što je olakšalo posao mnogim Java programerima. Da je ovo jedan od programskih jezika budućnosti govori i činjenica da je nedavno uvršten kao zvanični programski jezik za razvoj na Android platformi od strane Google-a[4], a postaje i sve popularniji u Spring zajednici za razvoj backend-a.



Slika 1: Logo Kotlin programskog jezika

2 O Kotlinu

Godine 2011. češka kompanija JetBrains je predstavila projekat Kotlin, novi jezik za izvršavanje na Java Virtuelnoj mašini. Predstavljajući novi jezik, kompanija JetBrains je istakla da su njeni programeri imali potrebe za nekim funkcionalnostima koje postojeći jezici nisu mogli da ponude, izuzev programskog jezika Scala[8]. Medjutim, naišli su na problem nešto sporije kompilacije kod ovog programskog jezika, tako da je jedan od navedenih ciljeva Kotlinu bio da ima brzinu kompilacije kao Java.

Kotlin je prenosivi, koncizan, statički tipiziran jezik opšte namene koji je napravljen da bude u potpunosti kompatibilan sa Javom. Odlično funkcionise na svim postojećim Java bibliotekama, ne razlikujući se od Jave što se tiče performansi, što omogućava pripajanje koda napisanog u Kotlinu u već postojeće Java projekte. Naravno, osnovna motivacija za nastanak jezika je dovela do toga da Kotlin ima jednostavniju sintaksu od Jave, čime je omogućeno uraditi više sa manje koda. Posledica toga je da kod biva čitljiviji i lakši za razumevanje.

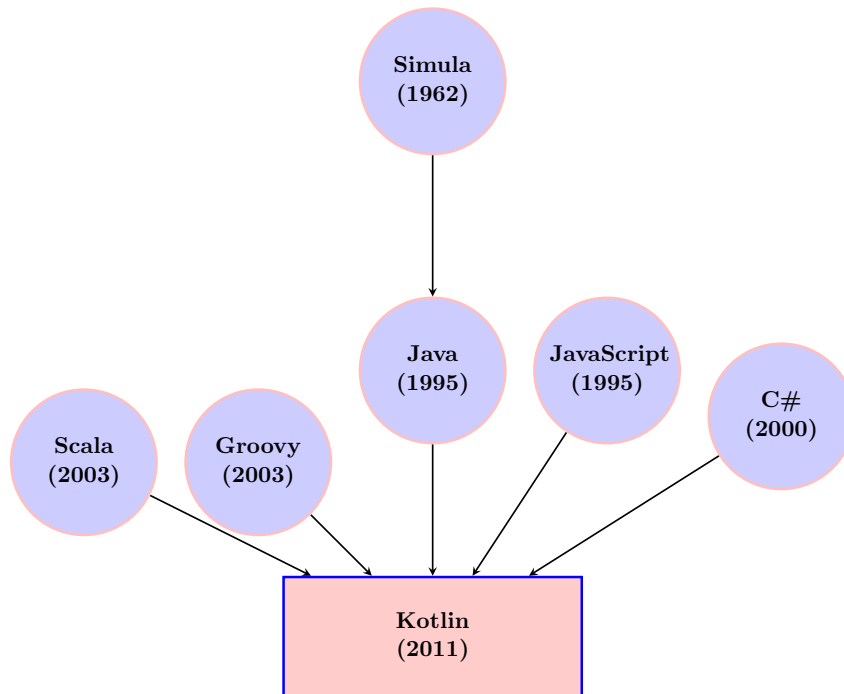
Kotlin se koristi za razvoj raznih serverskih aplikacija, kao i Android aplikacija, a može se koristiti svuda gde se koristi Java[1].

Ovaj programski jezik nosi naziv po ostrvu Kotlin koje se nalazi blizu Sankt Peterburga, baš kao što je Java dobila ime po ostrvu u Indoneziji.

3 Razvojno stablo

Na nastanak programskog jezika Kotlin, najveći uticaj imao je programski jezik **Java**, ali su kreatori bili inspirisani i programskim jezicima **C#**, **JavaScript**, **Scala** i **Groovy**.

Razvojno stablo programskog jezika Kotlin može se videti na slici 2.



Slika 2: Razvojno stablo programskog jezika Kotlin

3.1 Java

Početak jezika Java vezuje se za 1991. godinu kada se kod grupe inženjera u kompaniji Sun javila potreba za malim, efikasnim kodom koji je nezavistan od platforme na kojoj se izvršava[2]. Niklaus Virt, pronalazač programskog jezika Paskal, osmislio je dizajn prensivog jezika koji generiše medjukod za neku hipotetičku mašinu(virtuelnu mašinu). Ovaj kod bi zatim mogao da se koristi na bilo kojoj mašini koja ima odgovarajući interpreter. Iako je koncept prenosivog jezika zadržan, zaposleni u kompaniji Sun nisu svoj jezik zasnovali na Paskalu, već na C++, budući da su koristili UNIX. Dalje, oni su svoj jezik pravili kao objektno orijentisan, a ne kao proceduralni jezik. Nakon par godina projektovanja i unapredjivanja jezika, Sun je objavio prvu verziju jezika Java početkom 1996.godine.

Java je jednostavan jezik, dakle nema potrebe za pokazivačkom aritmetikom, strukturama i preopterećivanjem operatora. Sintaksa jezika Java je unapredjena verzija sintakse C++. Java je objektno-orijentisan i bezbedan jezik, takodje je prenosiv, što olakšava posao programerima koji pišu projekte, čije izvršavanje treba da bude obezbedjeno na svakom operativnom sistemu. Ovo je obezbedjeno time što se java ne prevodi, već se interpretira, što znaci da prilikom pokretanja programa, kompajler stvara objektnu datoteku, čiji je format nezavistan od operativnog sistema na kome se pokreće-kompajlirani kod se može izvršavati na mnogim procesorima, pod pretpostavkom prisustva izvršnog sistema Java. Takodje, Java je statički tipiziran jezik, što znači da moramo deklarirati promenljivu ili objekat pre korišćenja, ali ne nužno inicijalizovati.

Kotlin je od Jave nasledio sintaksu, objektno-orijentisan je i pokreće se na java viruelnoj mašini i podržava uključivanje i korišćenje već postojećih Java biblioteka. Prednosti u odnosu na jezik Java su što Kotlin poseduje lakšu sintaksu što ukazuje na to da se pisanjem programa u Kotlinu pravi manje grešaka, takodje, pokazalo se da se da u nekim slučajevima izvršavanje traje kraće od vremena izvršavanja Java koda kao i da program koji je napisan u Kotlinu zauzima bar onoliko memorije koliko bi zauzimao program napisan u Javi[1].

3.2 Scala

Scala je programski jezik za opšte softverske aplikacije[9]. Scala ima punu podršku za funkcionalno programiranje i veoma jak sistem statičkih tipova, što omogućava da se ostvari konciznost programa napisanih u Scali čime se obezbedjuje kod manjih dimenzija u odnosu na kodove napisane u

drugim jezicima opšte namene.

Scala je zapravo dizajnirana kako bi se prevazišli nedostaci u programskom jeziku Java, a predviđeno je da se izvorni kod Scala kompajlira u Java bajt kod, da bi se rezultujući kod izvršavao na Java virtuelnoj mašini. Isto kao kod jezika Kotlin, Java biblioteke se mogu koristiti direktno u Scala kodu i obrnuto, gde vidimo interoperabilnost ova dva jezika. Scala je takodje objektno-orijentisan jezik, ali za ralikom od Java, Scala ima mnogo karakteristika funkcionalnih programskih jezika, podrazumevane tipove podataka, imutabilnost, tipove višeg reda.

Ime Scala je kovanica od "skalabilnosti" i "jezika", što znači da je dizajnirana da raste sa potrebama svojih korisnika.

Kotlin je kao i Scala statički tipiziran jezik, takodje koristi ključnu reč **val**[3], čime se neka promenljiva deklarise kao konstanta, što znači da jednom kada je inicijalizujemo, ne možemo joj promeniti vrednost, slično kao **final** u Javi. Jedna osobina koju je takodje nasledio od Scala je sintaksa lambda izraza.

3.3 JavaScript

JavaScript se pojavio 1995.godine, pod nazivom LiveScript, ali je ubrzo preimenovan u JavaScript[7]. Napisao ga je Brendan Ajk, a dizajniran je od strane Netskejp kompanije. Motivacija za nastanak ovog jezika bila je nedovoljna dinamičnost veba, odnosno svetske komunikacione mreže. Sada je JavaScript vodeći jezik kada govorimo o pisanju veb sajtova, a svi moderni veb čitači ga podržavaju bez potrebe za instaliranjem dodatka.

JavaScript je dinamičan, slabo tipiziran i interpretiran programski jezik visokog nivoa. Standardizovan je po ECMAScript specifikaciji jezika. JavaScript je jezik višestruke paradigme, koji podržava objektno-orijentisani, imperativni i funkcionalni način programiranja. Sadrži API za rad sa tekstom, nizovima, datumima i regularnim izrazima.

Jedna od osobina koje je Kotlin nasledio od JavaScripta je deklaracija promenljive ključnom reči **var**, kao i definisanje funkcija ključnom reči **fun**.

3.4 C#

C# je jedan od mlađjih programskih jezika[6]. Nastao je 2000. godine kao sastavni deo Majkrosoftovog okruženja .NET Framework 1.0. Timom, koji se bavio razvojem ovog jezika, rukovodio je Anders Hejlsberg, koji je zapravo želeo da ovaj jezik bude kao C, samo objektno orijentisan.

C#, kao i većina modernih viših programskih jezika, podržava više paradigmi-objektno-orijentisanu, imperativnu, deklarativnu, generičku. Ovo je jezik opšte primene i namenjen je izradi aplikacija za .NET Framework platformu.

Ovaj jezik predstavlja naslednika jezika C i C++, a ime je dobio po muzičkoj notaciji, što znači da se nota C izvodi za pola tona više.

Razvojem ovog jezika, on se približio i funkcionalnom stilu. Značajne karakteristike koje poseduje C# su podržavanje lambda izraza, metode proširenja i anonimni tipovi. Ove osobine su omogućile C# programerima da koriste tehnike funkcionalnog programiranja kada god je to moguće. Metoda proširenja je jedna osobina koju je Kotlin nasledio od C#, čime se omogućava korisniku da u već postojećoj klasi samo "prilepi" funkciju na listu javnih metoda, a da je pritom ne postavi formalno unutar klase.

3.5 Groovy

Groovy je objektno orijentisani programski jezik opšte namene[5]. Gruvi programski jezik je kreiran za Java platformu i može se koristiti kao alternativa Java programskom jeziku. Sintaksa Gruvi jezika liči na sintaksu Java programskog jezika i ne samo to, već i veliki deo koda napisan u Java programskom jeziku je sintaksno ispravan u Groovy programskom jeziku. Programi pisani u Gruvi jeziku mogu da se povežu sa Java kodom i da koriste Java biblioteke. Groovy se takodje interpre kao Java, i dinamički se prevodi u Java bajt kod. Poseduje i dinamičku proveru tipova podataka.

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

Jedna od osobina koju je Kotlin nasledio od jezika Groovy je dopuštanje da se metode imenuju sa razmakom u imenu, pa čak i samo sa razmakom.

Zaključak

U ovom tekstu su prikazane osnove razvoja programskog jezika Kotlin, što podrazumeva jezike koji su uticali na njegovo nastajanje kao i njegovo razvojno stablo. Za više informacija o samom razvoju i unapredjivanju jezika, posetite zvaničnu veb stranicu programskog jezika Kotlin.

Literatura

- [1] Jakovljević Aleksandar, Šerbić Marko, Veljković Marko, and Vukadinović Selen. Kotlin kao programski jezik nove generacije. http://poincare.matf.bg.ac.rs/~milena/programskijezici/10_Kotlin_VeljkovicSerbicJakovljevicVukadinovic.pdf.
- [2] Cay S. Horstmann and Gary Cornell. *Core Java Volume I - Fundamentals, Ninth Edition*. Oracle, 2013.
- [3] Scala. Scala Documentation. <https://docs.scala-lang.org/tutorials/tour/basics.html.html>.
- [4] STARTIT. Uvod u Kotlin — Koje su prednosti novog zvaničnog jezika za razvoj Android aplikacija? <https://startit.rs/uvod-u-kotlin-koje-su-prednosti-zvanicnog-jezika-za-razvoj-android-aplikacija/>.
- [5] Wikipedia. Apache Groovy. https://en.wikipedia.org/wiki/Apache_Groovy.
- [6] Wikipedia. C Sharp. https://sr.wikipedia.org/wiki/C_Sharp.
- [7] Wikipedia. JavaScript. <https://en.wikipedia.org/wiki/JavaScript>.
- [8] Wikipedia. Kotlin(programming language). [https://en.wikipedia.org/wiki/Kotlin_\(programming_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)).
- [9] Wikipedia. Scala(programming language). [https://en.wikipedia.org/wiki/Scala_\(programming_language\)](https://en.wikipedia.org/wiki/Scala_(programming_language)).