
MATEMATIČKI FAKULTET UNIVERZITETA U BEOGRADU

SEMINARSKI RAD
iz predmeta Dizajn programskih jezika

Razvoj programskih jezika

David Aksović i Nikola Damjanović
davidaksovic@outlook.com i nikoladamjan@live.com
profesorka: Milena Vujošević Janičić
3. Novembar 2018.

Sadržaj

1	Uvod	2
2	Razvoj programskih jezika	3
2.1	I generacija računara(1939 - 1958)	4
2.2	II generacija računara(1958-1963)	4
2.3	III generacija računara(1964 - sredina 70-tih)	4
2.4	IV generacija računara-pre interneta(1980-1990)	5
3	Zaključak	7

1 Uvod

Razvoj programskih jezika, u bliskoj je vezi sa razvojem računara tj. sa razvojem hardvera [24]. Programiranje u današnjem smislu nastalo je sa pojavom računara fon Nojmanovog tipa čiji se rad kontroliše programima koji su smešteni u memoriji, zajedno sa podacima nad kojim operišu. Na prvim računarima tog tipa moglo je da se programira samo na mašinski zavisnim programskim jezicima, a od polovine 1950-ih nastali su jezici višeg nivoa koji su drastično olakšali programiranje. Prvi programski jezici zahtevali su od programera da bude upoznat sa najfinijim detaljima računara koji se programira. Problemi ovakvog načina programiranja su višestruki. Naime, ukoliko je želeo da programira na novom računaru, programer je morao da izuči sve detalje njegove arhitekture (na primer, skup instrukcija procesora, broj registara, organizaciju memorije). Programi napisani za jedan računar mogli su da se izvršavaju isključivo na istim takvim računarima i prenosivost programa nije bila moguća. Viši programski jezici namenjeni su ljudima a ne mašinama i sakrivaju detalje konkretnih računara od programera. Specijalizovani programi (tzv. Jezički procesori, programski prevodioci ili kompilatori i interpretatori) na osnovu specifikacije zadate na višem (apstraktnijem) nivou mogu automatski da proizvedu mašinski kod za specifičan računar na kojem se programi izvršavaju. Ovim se omogućava prenosivost programa (pri čemu, da bi se program mogao izvršavati na nekom konkretnom računaru, potrebno je da postoji procesor višeg programskog jezika baš za taj računar). Takođe, znatno se povećava nivo apstrakcije prilikom procesa programiranja što u mnogome olakšava ovaj proces. Razvojni ciklus programa u većini savremenih viših programskih jezika teče na sledeći način. Danas je, nakon faze planiranja, prva faza u razvoju programa njegovo pisanje tj. unošenje programa na višem programskom jeziku (tzv. izvorni program ili izvorni kod – engl. source code), što se radi pomoću nekog editora teksta. Naredna faza je njegovo prevođenje, kada se na osnovu izvornog programa na višem programskom jeziku dobija prevedeni kod na asemblerskom odnosno mašinskom jeziku (tzv. objektni kod – engl. object code), što se radi pomoću nekog programskog prevodioca. U fazi povezivanja više objektnih programa povezuje se sa objektnim kôdom iz standardne biblioteke u jedinstvenu celinu (tzv. izvršivi program – engl. executable program). Povezivanje vrši specijalizovan program poveziivač, tj. linker (engl. linker) ili uređivač veza. Nakon povezivanja, kreiran je program u izvršivom obliku i on može da se izvršava. Nabrojane faze se obično ponavljaju, vrši se dopuna programa, ispravljanje grešaka, itd.

2.1 I generacija računara(1939 - 1958)

Koriste elektronske cevi kao osnovne logičke elemente koje su nepouzdana, troše mnogo struje, mnogo se zagrevaju, velike su i mnogobrojne, pa su računari ogromni.

- Ulazna tehnologija-bušene kartice,
- Unutrašnja memorija-rotirajući magnetni doboši,
- Jezik za programiranje - mašinski ili simbolički (asemblerski).

Asembler [2] je jezik nižeg nivoa koji procesorkse arhitekture predstavlja čitljivom obliku. Program koji se pravi kroz assembler se takođe i naziva asemblerski kod. Taj kod se pretvara u mašinski kod koji procesor izvršava. Kod koji procesor vraća se iz mašinskog pretvara u asemblerski i on je uglavnom teško čitljiv. Asembler se danas koristi samo za pravljenje drajvera ili embeded sistema.

2.2 II generacija računara(1958-1963)

Počinju da se prave tranzistori od silicijuma koji su sitniji, jeftiniji, zauzimaju mnogo manje prostora, pouzdaniji su, troše manje struje, manje se zagrevaju i brzi su od elektronskih cevi. Kao i cevi, morali su da se zavaruju i uklapaju zajedno tako da formiraju elektronsko kolo.

- Unutrašnja memorija-magnetno jezgro (mali magneti oblika torusa)
- Spoljašnja memorija- magnetni diskovi i trake
- Za programiranje se koriste viši programski jezici

Korišćenje računara je u velikom porastu, javljaju se novi korisnici i programski paketi. Programi postaju čitljiviji i prenosiviji.

FORTRAN [3] se pojavio 1958. Njegov glavni autor je Džon Bakus, koji je implementirao i prvi interpretator i prvi kompilator za ovaj jezik. Ime FORTRAN dolazi od *IBM Mathematical FORMula TRANslating System*, što ukazuje na to da je osnovna motivacija bila da se u okviru naučno-tehničkih primena omogući unošenje matematičkih formula, dok je sistem taj koji bi unete matematičke formule prevodio u niz instrukcija koje računar može da izvršava. Do danas Fortran je dodao podršku za strukturano, objektno-orjentisano i konkurentno programiranje. Fortran je bio baza za dizajn velikom broju programskih jezika. Jedan od poznatijih čiji se dizajn bazira na Fortranu II je BASIC.

Lisp [4] (**LIS**t Processing) koji objavljuje Dzon Mekarti godinu dana nakon FORTRAN-a. Pravi je primer funkcionalnog programiranja. Na početku je bio omiljen za razvoj veštačke inteligencije. Kao jedan od prvih programskih jezika, Lisp je dodao ideje o strukturi stabla, automatsko skladištenje podataka, funkcije višeg reda, rekurziju itd.

COBOL [5] (**CO**mmon **B**usiness **O**riented **L**anguage) služi za pisanje biznis i finansijskih aplikacija i kao podrška administrativnim sistemima u kompanijama i vladama. Dizajniran je 1959. godine od strane CODASYL i delimično se bazirao na prethodnom programskom jeziku koji je dizajnirala Grejs Hoper. Proširenja uključuju podršku za strukturalno i objektno-orijentisano programiranje.

ALGOL [6] (**ALGO**rithmic **L**anguage) je proceduralni jezik, izmišljen 1958. godine. Dizajniran je kako bi se izbegli neki problemi sa FORTRAN-om. Imao je dosta uticaja jezike koji su dosli kasnije kao što su C, Paskal i još mnogo drugih.

2.3 III generacija računara(1964 - sredina 70-tih)

Treća generacija računara bila je zasnovana na integrisanim kolima smeštenim na silicijumskim čipovima. Računari se smanjuju i njihova brzina je veća, samim tim imaju veću primenu u društvu. Njihova poslovna primena je zahtevala nove programske jezik. Pored toga uvode se i standardi za više programske jezike kao što je ANSI FORTRAN. Većina jezičkih paradigmi koje danas koristimo su nastale u ovo doba.

Simula [7] – dizajniran 1962. godine, koji je sintaksički naslednik ALGOL 60, smatra se prvim objektno-orijentisanim jezikom, zbog uvođenja objekata, klasi, nasleđivanja i slično. Danas se koristi za modeliranje procesa, algoritama, protokola kao i u računarskoj grafici.

C [8] je proceduralni programski jezik nastao 1972. godine. Autor jezika je Denis Riči, a nastao u istraživačkom centru *Bell Laboratories* za potrebe operativnog sistema UNIX. Njegova arhitektura i logika se nije mnogo razlikovala od asemblerske. Takođe postoje izrazi koji pozivaju odgovarajuće asemblerske naredbe.

Pascal [9] je imperativni programski jezik, koji je 1970. godine razvio Niklaus Virt, kao jezik pogodan za učenje strukturnog programiranja. Imenovan je po čuvenom francuskom matematičaru i filozofu Blezu Paskalu, tvorcu prve računске mašine koja je imala mogućnost izvođenja operacije sabiranja. Standardizovan je 1983. godine od strane Međunarodnog komiteta za standardizaciju. Paskal je razvijen po obrazcu jezika ALGOL 60.

SQL [10] (Structured Query Language) – je relacioni upitni jezik nastao u IBM-ovoj istraživačkoj laboratoriji 1974. godine. Jezik se u početku zvao SEQUEL (Structured English Query Language) i predstavljao je programski interfejs (API) za System R, prototipski sistem za upravljanje bazom podataka.

2.4 IV generacija računara-pre interneta(1980-1990)

Četvrta generacija zasnovana je na visoko integrisanim kolima kod kojih je na hiljade kola smešteno na jedan silikonski čip. Prvi mikropcesori su nastali i cela procesorska jedinica je stajala na jednom čipu. Ovakva tehnologija smanjila je veličinu računara pa su samim tim bili i pogodniji za kućnu upotrebu. Programski jezici koji su nastali u ovo doba nisu se zasnivali na novim paradigmama, već su sve novitete samo dodavali na paradigme prošle decenije.

C++ [11] je viši programski jezik koji je prvobitno razvijen u Bell Labs za objektno orijentisano programiranje u projektu pod rukovodstvom Bjarnea Stroustrupa tokom 1980-ih kao proširenje programskog jezika C. Kasnije 1998. godine je i standardizovan.

MATLAB [12] (MATrix LABoratory) je okruženje za numeričke proračune i programski jezik četvrte generacije koji je razvila firma MathWorks. MATLAB omogućava lako manipulisanje matricama, prikazivanje funkcija, implementaciju algoritama, stvaranje grafičkog korisničkog interfejsa kao i povezivanje sa programima pisanim u drugim jezicima među kojima su C, C++, C#, Java, Fortran i Python.

Perl [13] je nastao je 1987. godine kao Unix skripting jezik radi lakšeg izvršavanja procesa. Perl uzima attribute od C-a i shell-a, kasnije tokom njegovog razvitka Perl dodaje i podršku za objekte, reference, leksičke varijable i module.

Wolfram Language [14] je jezik koji pripada više paradgmi razvio *Wolfram Reaserch* koji se koristi za matematičko simboličko izračunavanje u programu Mathematica i u Wolfram Programming Cloud-u. Pripada paradigmama simboličkog izračunavanja, funkcionalno programiranje i logičko programiranje. Funkcije koje su napravljene do sad su za generisanje i pokretanje Turingovih mašina, kreiranje grafike i zvuka, analiziranje 3D modela, manipulaciju matrica i rešavanje diferencijalnih jednačina.

Ada [15] je projektovan od strane Ministarstva Odbrane SAD, 1980. godine. Nastao je kako bi se sve odbrambene aplikacije pisale u njemu. Bitno svojstvo Ade je multitasking, koji omogućava istovremeno izvršavanje većeg broja programa.

Dolaskom internet tehnologije došle su nove potrebe kao što su preglednije veb stranice, brži internet čitači i slično. Takođe umrežavanje velikog broja kompjutera je dovelo do toga da se informacije brzo i lako prenose, samim tim novi programski jezici su bili vrhunac mašte velikih timova širom sveta.

Haskell [16] je funkcionalni, strogo tipiziran jezik opšte namene sa nestriktnom semantikom, dizajniran 1990. godine. Haskell karakteriše lenjo izračunavanje, uparivanje šablona, tehnika zadavanja listi i polimorfizam tipova. Funkcije u Haskell-u nemaju sporedne efekte.

Python [17] je programski jezik visokog nivoa, opšte namene. Prvi put se pojavio 1991. godine. Python je dizajniran tako da je njegov kod što čitljiviji, to je postignuto tako što su uvedeni značajne

beline. Python poseduje dinamički tipiziran sistem i automatsko upravljanje memorijom i pripada više programskih paradigmi.

Visual Basic [18] je zasnovan na događajima i sastavni deo programskog okruženja Microsoft. Bazira se na Basic-u, na njemu se aplikacije brzo razvijaju i manipulišu baze podataka.

Ruby [19] je opšte namenski jezik, pripada više paradigmi. Prvi put se pojavio 1995. godine. Na njegov razvoj najviše su uticali Perl, Ada i Lisp. Paradgme kojima pripada su funkcionalno programiranje i objektno-orjentisano. Dinamički je tipiziran i poseduje automatsko manipulisanje memorijom.

Java [20] je opšte namenski jezik koji konkurentan, baziran na klasama i objektno-orjentisan. Prvi put se pojavio 1995. godine. Novitet koji donosi je da kada se program kompilira onda može da se pokrene bilo gde. Java aplikacije se kompiliraju do bajt koda i onda može da se pokrene na bilo kojoj Java Virtuelnom mašini. Uticaj na njegovu sintaksu su najviše imali C i C++. Od 2016. godine Java je jedan od najpopularnijih jezika na svetu.

JavaScript [21] je interpretirani programski jezik, takođe je karakterizovan kao dinamički, slabo tipiziran i pripada više paradigmi. HTML, CSS i JS zajedno su struktura svake internet stranice danas. JS podržava objektno-orjentisano, funkcionalno i imperativno programiranje. Na JS je najviše uticao Self i Scheme.

PHP [22] jezik za skriptovanje sa serverske strane specijalno dizajniran za *Web development* i takođe koršćen i kao opšte namenski jezik. Napravljen 1994. godine. PHP je pre značilo Personal Home Page, danas je to promenjeno u PHP: Hypertext Preprocessor. Njegov interpreter je besplatan software koji može da se skine sa interneta.

3 Zaključak

Računarstvo se bavi izučavanjem računara, programa i programskim jezicima nad njima ali i opštije, izučavanjem teorije i prakse procesa računanja i primene računara u raznim oblastima nauke, tehnike i svakodnevnog života. Računari u današnjem smislu nastali su polovinom XX veka, ali koreni računarstva su mnogo stariji od prvih računara. Vekovima su ljudi stvarali mehaničke i elektromehaničke naprave koje su mogle da rešavaju neke numeričke zadatke. Do danas, stvoreno je više hiljada programskih jezika, i novi se eksponencijalno stvaraju svake godine. Današnji programi pisani na višim programskim jezicima su lako prenosivi na većinu drugih računara i računar sada kao takav, programabilan je više nego ikad, toliko da gde god da se okrenemo vidimo koliko svakodnevno služi čovečanstvu, i koliko svet u velikoj meri zavisi od računara.

Literatura

- [1] History of programming languages. <http://rigaux.org/language-study/diagram-light.png>.
- [2] Wikipedia. https://en.wikipedia.org/wiki/Assembly_language.
- [3] Wikipedia. <https://en.wikipedia.org/wiki/Fortran>.
- [4] Wikipedia. [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)).
- [5] Wikipedia. <https://en.wikipedia.org/wiki/COBOL>.
- [6] Wikipedia. <https://en.wikipedia.org/wiki/ALGOL>.
- [7] Wikipedia. <https://en.wikipedia.org/wiki/Simula>.
- [8] Wikipedia. [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)).
- [9] Wikipedia. [https://en.wikipedia.org/wiki/Pascal_\(programming_language\)](https://en.wikipedia.org/wiki/Pascal_(programming_language)).
- [10] Wikipedia. <https://en.wikipedia.org/wiki/SQL>.
- [11] Wikipedia. <https://en.wikipedia.org/wiki/C%2B%2B>.
- [12] Wikipedia. <https://en.wikipedia.org/wiki/MATLAB>.
- [13] Wikipedia. <https://en.wikipedia.org/wiki/Perl>.
- [14] Wikipedia. https://en.wikipedia.org/wiki/Wolfram_Language.
- [15] Wikipedia. [https://en.wikipedia.org/wiki/Ada_\(programming_language\)](https://en.wikipedia.org/wiki/Ada_(programming_language)).
- [16] Wikipedia. [https://en.wikipedia.org/wiki/Haskell_\(programming_language\)](https://en.wikipedia.org/wiki/Haskell_(programming_language)).
- [17] Wikipedia. [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
- [18] Wikipedia. https://en.wikipedia.org/wiki/Visual_Basic.
- [19] Wikipedia. [https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language)).
- [20] Wikipedia. [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).
- [21] Wikipedia. <https://en.wikipedia.org/wiki/JavaScript>.
- [22] Wikipedia. <https://en.wikipedia.org/wiki/PHP>.
- [23] dr Gordana Pavlović-Lažetić. *Programiranje 1*. 2014/2015.
- [24] Predrag Janičić i Filip Marić. *Programiranje 1*. Matematički fakultet Univerziteta u Beogradu, Studentski trg 16,11000 Beograd, 2018.