

Dizajn programskih jezika

— Odnos programskih jezika i programskih paradigmi —

Milena Vujošević Jančić

Matematički fakultet, Univerzitet u Beogradu

Sadržaj

1	Uvod	1
1.1	Programski jezici	1
1.2	Paradigme i programski jezici	4
1.3	Povezanost paradigmi i jezika	5
2	Razvoj jezika i paradigmi	6
2.1	Razvoj jezika	6
2.2	Vrste programskih paradigmi	8
2.3	Pitanja	9
3	Osnovne paradigme	10
3.1	Imperativna (proceduralna) paradigma	10
3.2	Objektno-orijentisana paradigma	10
3.3	Funkcionalna paradigma	11
3.4	Logička paradigma	11
3.5	Pitanja	12
4	Dodatne paradigme	13
4.1	Komponentna paradigma	13
4.2	Konkurentna paradigma	14
4.3	Paradigma programiranja ograničenja	15
4.4	Skript paradigma	16
4.5	Paradigma upitnih jezika	16
4.6	Reaktivna paradigma	17
4.7	Vizuelna paradigma	18
4.8	Pitanja	18
5	Literatura	18

1 Uvod

1.1 Programski jezici

Jezici i programski jezici

- Jezik je skup pravila za komunikaciju između subjekata.
- Pomoću jezika se predstavljaju i prenose informacije.
- Prirodni jezik se koristi za komunikaciju između ljudi u govornoj ili pisanoj formi.
- Programski jezik se koristi ...?

Jezici i programski jezici

- Programski jezik služi, prvenstveno, za komunikaciju između čoveka i računara, ali može da se koristi i za komunikaciju između mašina, kao i za komunikaciju između ljudi
- Programski jezici se mogu deliti na razne načine
- Jedna podela je na mašinski zavisne i mašinski nezavisne, i na dalje će uglavnom biti reči o mašinski nezavisnim (višim) programskim jezicima
- Postoje razne definicije programskih jezika.
- Kako biste Vi definisali programski jezik?

Programski jezici — definicije

- Programski jezik je jezik konstruisan formalno da bi se omogućilo zadavanje instrukcija mašinama, posebno računarima. (wikipedia)
- Programski jezik je jezik za pisanje programa koje računar zna i može izvršiti.
- Programski jezik je veštački jezik koji služi za opis računarskih programa.
- Programski jezik je veštački jezik za opis konstrukcija (pisanje instrukcija) koje mogu biti prevedene u mašinski jezik i izvršene od strane računara. (American Heritage Dictionary)

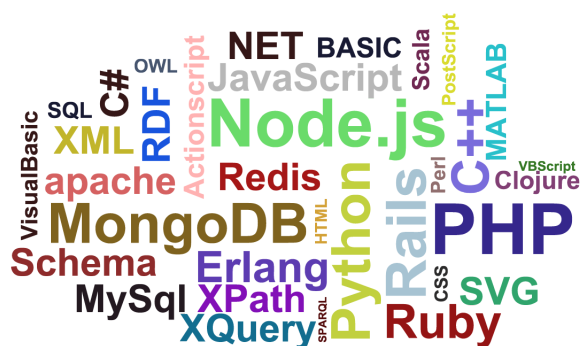
Programski jezici — definicije

- Programski jezik je skup sintaktičkih i semantičkih pravila koji se koriste za opis (definiciju) računarskih programa.
- Programski jezik je notacioni sistem čitljiv za računare i ljude, a služi za opis poslova koje treba da obavi računar.

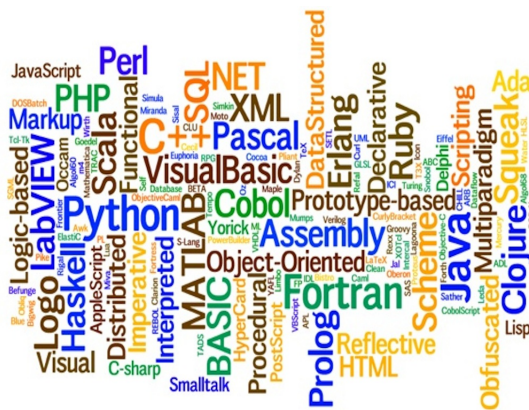
Koliko programskih jezika postoji?



Koliko programskih jezika postoji?



Koliko programskih jezika postoji?



Koliko programskih jezika postoji?

Programska paradigma

- Programski obrazac, programski stil, programski šablon, način programiranja
- Fundamentalni stil programiranja
- Klasifikacija međusobno sličnih programskih jezika

1.3 Povezanost paradigmi i jezika

Uloga programskih paradigmi

- Broj programskih paradigmi nije tako veliki kao broj programskih jezika
- Izučavanjem programskih paradigmi upoznaju se globalna svojstva jezika koji pripadaju toj paradigmi
- Informacija da neki jezik pripada nekoj paradigmi nam govori o osnovnim svojstvima i mogućnostima jezika
- Poznavanje određene paradigme nam značajno olakšava da savladamo svaki programski jezik koji toj paradigmi pripada

Povezanost paradigmi i jezika

- Programske paradigme su usko povezane sa programskim jezicima.
- Svakoj programskoj paradigmi pripada više programskih jezika, na primer proceduralnoj paradigmi pripadaju programski jezici Pascal i C, objektno-orijentisanoj paradigmi pripadaju Simula, JAVA...
- Potrebno je izučiti svojstva najistaknutijih predstavnika pojedinih programskih paradigmi
- Koliko *jezika* [paradigmi](#) znaš, toliko vrediš! Preciznije: Koliko [predstavnika](#) različitih [paradigmi](#) znaš, toliko vrediš!
- Sledeći nivo apstrakcije čine koncepti koji su zajednički za različite paradigme

Povezanost paradigmi i jezika

- Jedan programski jezik može podržati više paradigmi, na primer C++ podržava klasičan proceduralni stil, ali i objektno-orijentisani i genrički stil programiranja
- Za rešavanje nekog konkretnog problema, posebno je bitan izbor programskog jezika
- [izborJezika.png](#)

2 Razvoj jezika i paradigmi

Bitni momenti u razvoju računara

- Jedan od prvih elektronskih računara 1939. ABC za rešavanje sistema linearnih jednačina
- ENIAC — prvi elektronski računar opšte namene (1946)
- Konceptualna promena krajem 1940. u vidu fon Nojmanove arhitekture
- Vezuje se za fon Nojmana i računar EDVAC 1951, iako je o nekim elementima ove arhitekture i ranije bilo reči

Podele programskih jezika

- Mašinski zavisni
- Prednosti i mane mašinski zavisnih jezika
- Mašinski nezavisni
- Prednosti i mane mašinski nezavisnih jezika

2.1 Razvoj jezika

Kratka istorija

- FORTRAN — FORMula TRANslating system, 1957, John Backus i IBM
- LISP — LIST Processing, malo posle FORTRANa, 1958, John McCarthy i Paul Graham
- COBOL — COMmon Business-Oriented language, 1959, Grace Hopper

Kratka istorija

- 60-te ALGOL (58,60,68), Simula, Basic
- 70-te C, Pascal, Smalltalk, Prolog
- 80-te C++, Erlang
- 90-te Haskell, Python, Visual Basic, Ruby, JAVA, PHP, OCaml, Lua, JavaScript...
- C#, Scala, F#, Elixir...

Razvoj programskih jezika

- Postoji veliki broj programskih jezika, neki su široko rasprostranjeni, neki se više ne koriste
- Java, C, C++, C#, Pascal, Visual Basic, Lisp, Scheme, ML, Ruby, Modula-2, JavaScript, Fortran, Cobol, Haskell, Oberon, Prolog, PHP, Perl, Python, Ada, Scala
- Nastanak i razvoj programskih jezika dosta dobro se može prikazati pomoću razvojnog stabla.
- Razvojno stablo mogućava da se sagleda vreme nastanka pojedinih programskih jezika, kao i međusobni uticaji.

Razvoj programskih jezika

- Nema jedinstvenog razvojnog stabla (od autora zavisi na koje jezike će staviti akcenat i kako će ih međusobno povezati).
- [prog_lang.pdf](#)
- [prog_lang_poster.pdf](#)
- [diagram-full.pdf](#)
- [diagram-light.pdf](#)
- <http://startit.rs/kad-bi-programski-jezici-bili-deca-u-vrticu/>
- U kom razdoblju je nastao najveći broj programskih jezika?
- Koji su najuticajniji programski jezici?
- Kada su nastali najuticajniji programski jezici?
- Zašto postoji veliki broj programskih jezika?

Razvoj programskih paradigmi

- Nove programske paradigme nastajale su uz težnju da se olakša proces programiranja.
- Istovremeno, nastanak novih paradigmi povezan je sa efikasnim kreiranjem sve kompleksnijeg softvera.
- Svaka novonastala paradigma, bila je promovisana preko nekog programskog jezika.
- Razvoj programskih paradigmi (kao i programskih jezika) skopčan je i sa razvojem hardvera.

Razvoj programskih paradigmi

- Različita shvatanja programskih paradigmi
- Ne postoji jedinstveno mišljenje naučnika o programskim paradigmi (vrstama programskih paradigmi, njihovom značaju, najistaknutijim programskim jezicima pojedinih paradigmi itd.)
- Moguće su različite podele na programske paradigme.

2.2 Vrste programskih paradigmi

Osnovne programske paradigme

- Najopštija podela je na proceduralnu i deklarativnu paradigmu
- Proceduralna paradigma — osnovni zadatak programera da opiše način (proceduru) kojim se dolazi do rešenja problema.
- Deklarativna paradigma — osnovni zadatak programera je da precizno opiše problem, dok se mehanizam programskog jezika bavi pronalaženjem rešenja problema.

Vrste programskih paradigmi

- Osnovne programske paradigme
 - Imperativna paradigma
 - Objektno-orijentisana paradigma
 - Funkcionalna paradigma
 - Logička paradigma
- Ostale paradigme se često tretiraju kao podparadigme ili kombinacije osnovnih.

Napomena o imperativnoj i proceduralnoj paradigmi

- Postoji više shvatanja proceduralne paradigme:
 1. — Proceduralna paradigma je podparadigma imperativne paradigme koju karakteriše, pored naredbi, i njihovo grupisanje u podprograme (funkcije).
 - U ovom slučaju, u literaturi se često imperativna i proceduralna paradigma koriste kao sinonimi.
 - Imperativna paradigma se karakteriše postojanjem naredbi, dok se deklarativna paradigma karakteriše nepostojanjem naredbi.
 2. — Proceduralna paradigma je svaka paradigma kod koje se u procesu programiranja opisuje algoritam (procedura) rešavanja problema.
 - U ovom slučaju je imperativna paradigma podparadigma proceduralne paradigme dok je deklarativna paradigma (pitanje ŠTA) suprotna od proceduralne paradigme (pitanje KAKO).

Programski jezici i paradigme

- Programski jezik je sredstvo koje koristi **čovjek** da izrazi **proces** pomoću kojeg **računar** rešava nekakav **problem**. U zavisnosti od toga na kojoj od ovih reči je akcenat, programskim jezikom je podržana dominantna programska paradigama:
 - **čovjek** — logička paradigma
 - **proces** — funkcionalna paradigma
 - **računar** — proceduralna paradigma
 - **problem** — objektno-orijentisana paradigma

Programski jezici i paradigme

- Prethodna definicija programskog jezika je prilagođena osnovnim programskim paradigmama.
- Ova definicija se može dopuniti tako da se preko nje mogu obuhvatiti i druge paradigme.
- Na primer, modifikacija može biti: Programski jezik je sredstvo koje koristi **čovjek** da izrazi **proces** pomoću kojeg **računar**, koristeći **paralelnu obradu**, rešava nekakav **problem**. Ako je akcenat na **paralelnoj obradi**, dolazi se do konkurentne (paralelne) paradigme.

Dodatne programske paradigme

- Komponentna paradigma
- Konkurentna paradigma
- Skript paradigma
- Generička paradigma
- Paradigma programiranja ograničenja
- Paradigma upitnih jezika
- Reaktivna paradigma
- Vizuelna paradigma

2.3 Pitanja

Pitanja

- Značenje reči paradigma i programska paradigma.
- Uloga programskih paradigmi.
- Definicija programskog jezika.
- Povezanost paradigmi i jezika.
- Razvoj programskih jezika.

3 Osnovne paradigme

3.1 Imperativna (proceduralna) paradigma

Imperativna paradigma

- Imperativna paradigma nastala je pod uticajem Fon Nojmanove arhitekture računara
- Može se reći da se zasniva na tehnološkom konceptu digitalnog računara
- Proces izračunavanja se odvija slično kao neke svakodnevne rutine (znanovan je na algoritamskom načinu rada), kao što je spremanje hrane korišćenjem recepata, popravljanje kola i sl.
- Može da se okarakteriše rečenicom: “prvo uradi ovo, zatim uradi ono”
- Procedurom se saopštava računaru KAKO se problem rešava, tj navodi se precizan niz koraka (algoritam) potreban za rešavanje problema

Imperativna paradigma

- Osnovni pojam imperativnih jezika je naredba
- Naredbe se grupišu u procedure i izvršavaju se sekvencijalno ukoliko se eksplicitno u programu ne promeni redosled izvršavanja naredbi
- Upravljačke strukture su naredbe grananja, naredbe iteracije, i naredbe skoka (goto)
- Oznake promenljivih su oznake memorijskih lokacija pa se u naredbama često mešaju oznake lokacija i vrednosti - to izaziva bočne efekte.
- C, Pascal, Basic, Fortran, PL, Algol...

3.2 Objektno-orijentisana paradigma

Objektno-orijentisana paradigma

- Ovo je jedna od najpopularnijih programskih paradigmi
- Sazrela je početkom osamdesetih godina prošlog veka, kao težnja da se jednom napisani softver koristi više puta
- Simulacija (modeliranje) spoljašnjeg sveta pomoću objekata
- Objekti interaguju međusobno razmenom poruka
- Mogla bi da se okarakteriše rečenicom: “Uputi poruku objektima da bi simulirao tok nekog fenomena”

Objektno-orijentisana paradigma

- Podaci i procedure (funkcije) se ućauravaju (enkapsuliraju) u objekte
- Koristi se skrivanje podataka da bi se zaštitila unutrašnja svojstva objekata
- Objekti su grupisani po klasama (klasa predstavlja šablon (koncept) na osnovu kojeg se kreiraju konkretni objekti, tj. instance)
- Klase su najčešće hijerarhijski organizovane i povezane mehanizmom nasleđivanja.
- Simula 67, SmallTalk, C++, Eiffel, Java, C#

3.3 Funkcionalna paradigma

Funkcionalna paradigma

- Rezultat težnje da se drugačije organizuje proces programiranja
- Izračunavanja su evaluacije matematičkih funkcija
- Zasnovana je na pojmu matematičke funkcije i ima formalnu strogo definisanu matematičku osnovu u lambda računu
- Mogla bi se okarakterisati narednom rečenicom “Izračunati vrednost izraza i koristiti je”

Funkcionalna paradigma

- Eliminirani su bočni efekti što utiče na lakše razumevanje i predviđanje ponašanja programa — Izlazna vrednost funkcije zavisi samo od ulaznih vrednosti argumenata funkcije
- Najistaknutiji predstavnik funkcionalne paradigme je programski jezik Lisp.
- Nastala pedesetih i početkom šezdesetih godina prošlog veka, stagnacija u razvoju sedamdesetih godina prošlog veka, oživljavanje funkcionane paradigme programskim jezikom Haskell
- Lisp, Scheme, Haskell, ML, Scala, OCaml

3.4 Logička paradigma

Logička paradigma

- Nastaje kao težnja da se u kreiranju programa koristi isti način razmišljanja kao i pri rešavanju problema u svakodnevnom životu
- Deklarativna paradigma
- Opisuju se odnosi između činjenica i pravila u domenu problema; koriste se aksiome, pravila izvođenja i upiti

- Logička paradigma se dosta razlikuje od svih ostalih po načinu pristupa rešavanju problema.
- Nije jednako pogodna za sve oblasti izračunavnja, osnovni domen je rešavanje problema veštačke inteligencije

Logička paradigma

- Izvršavanje programa zasniva se na sistematskom pretraživanju skupa činjenica uz korišćenje određenih pravila zaključivanja.
- Zasnovana na matematičkoj logici, tj. na predikatskom računu 1. reda.
- Zasnovana na automatskom dokazivanju teorema (metod rezolucije)
- Mogla bi da se okarakteriše rečenicom: “Odgovori na pitanje kroz traženje rešenja”
- Najpoznatiji programski jezik logičke paradigme je PROLOG
- Prolog, ASP, Datalog, CLP, ILOG, Solver, ParLog, LIFE

3.5 Pitanja

Pitanja

- Šta karakteriše proceduralnu paradigmu?
- Šta karakteriše deklarativnu paradigmu?
- Koje su osnovne četiri programske paradigme?
- Nabroj bar četiri dodatne programske paradigme.

Pitanja

- Koje su osnovne karakteristike imperativne paradigme?
- Nabroj tri jezika koji pripadaju imperativnoj paradigmi.
- Koje su osnovne karakteristike ... paradigme?
- Nabroj tri jezika koji pripadaju ... paradigmi.

Pitanja

- Šta je programska paradigma?
- Koje su osnovne programske paradigme?
- Šta je programski jezik?
- Koji je odnos programskih jezika i programskih paradigmi?
- Zašto su nastajale i nastaju nove programske paradigme?

4 Dodatne paradigme

Dodatne programske paradigme

- Komponentna paradigma
- Konkurentna paradigma
- Paradigma programiranja ograničenja
- Skript paradigma
- Paradigma upitnih jezika
- Reaktivna paradigma
- Vizuelna paradigma

4.1 Komponentna paradigma

Komponentna paradigma

- Ideja je da se softver sklupa od većih gotovih komponenti, kao što se to radi kod sklapanja elektronskih i tehničkih uređaja
- Softverska komponenta je kolekcija delova (metoda i objekata) koji obezbeđuju neku funkcionalnost.
- Kao i tehničke komponente, i softverske komponente mogu biti proste ili kompleksne, mogu delati samostalno ili u konjunktiji sa drugim jedinicama.

Komponentna paradigma

- Komponentna paradigma je nova paradigma ili potparadigma objektno-orijentisane paradigme?
- Nezavisno od toga, pitanje je: da li treba posebno izučavati komponentno programiranje? Stil programiranja koji je u ekspanziji i treba mu pokloniti posebnu pažnju.
- Ideja je da se uprosti proces programiranja i da se jednom kreirane komponente mnogo puta koriste.
- Komponenta je jedinica funkcionalnosti sa “ugovorenim” interfejsom.
- Interfejs definiše način na koji se komunicira sa komponentom, i on je u potpunosti odvojen od implementacije.

Komponentna paradigma

- Komponente se međusobno povezuju da bi se kreirao kompleksan softver
- Način povezivanja komponenti treba da bude jednostavan, po mogućnosti prevlačenjem i spuštanjem na željenu lokaciju.
- Kreiranje programa se vrši biranjem komponenti i postavljanjem na pravo mesto, a ne pisanjem “linije za linijom”.
- U okviru komponentnog programiranja, važno je razvojno okruženje koje se koristi, dok sama implementacija komponenti i kod koji se komponentnim programiranjem generiše može da bude u različitim programskim jezicima, npr JAVA, C++, C# ...

4.2 Konkurentna paradigma

Konkurentna paradigma

- Konkurentnu paradigmu karakteriše više procesa koji se izvršavaju u istom vremenskom periodu, a koji imaju isti cilj
- Postoje različite forme konkurentnosti:
 - **Konkurentnost u užem smislu** — jedan procesor, jedna memorija
 - **Paralelno programiranje** — više procesora, jedna memorija
 - **Distribuirano programiranje** — više procesora, više memorija

Konkurentna paradigma

- **Konkurentnost u užem smislu** karakteriše preklapajuće izvršavanje više procesa koji koriste isti procesor i koji komuniciraju preko zajedničke memorije.
- Ovi procesi modeliraju procese spoljašnjeg sveta koji mogu da se dese konkurentno, na primer kod operativnih sistema

Konkurentna paradigma

- Ukoliko postoji više procesora sa pristupom jedinstvenoj memoriji, onda je u pitanju **paralelno programiranje**
- Procesni međusobno komuniciraju preko zajedničke memorije
- Cilj paralelnog izračunavanja je ubrzanje toka izračunavanja

Konkurentna paradigma

- Ukoliko postoji više procesora od kojih svaki ima svoju memoriju, onda je u pitanju **distribuirano programiranje**
- Procesi međusobno šalju poruke da bi razmenili informacije.
- Distribuirano izračunavanje čine grupe umreženih računara koje imaju isti cilj za posao koji izvršavaju.
- Može se shvatiti kao vrsta paralelnog izračunavanja ali sa drugačijom međusobnom komunikacijom koja nameće nove izazove.

Konkurentna paradigma

- Pisanje konkurentnih (konkurentnih u užem smislu, paralelnih, distribuiranih) programa je značajno teže od pisanja sekvencijalnih programa.
- Konkurentno programiranje nameće nove probleme, po pitanju sinhronizacije procesa i pristupa zajedničkim podacima.
- Za osnovne koncepte konkurentnog programiranja potrebno je obezbediti odgovarajuću podršku u programskom jeziku.
- Ada, Modula, ML, Java ...

4.3 Paradigma programiranja ograničenja

Paradigma programiranja ograničenja

- U okviru paradigme programiranja ograničenja zadaju se relacije između promenljivih u formi nekakvih ograničenja
- Ograničenja mogu biti raznih vrsta (logička, linearna...)
- Ova ograničenja ne zadaju sekvencu koraka koji treba da se izvrše već osobine rešenja koje treba da se pronađe
- Deklarativna paradigma
- Jezici za programiranje ograničenja često su nadogradnja jezika logičke paradigme, na primer PROLOGa
- Postoje biblioteke za podršku ovoj vrsti programiranja u okviru imperativnih programskih jezika, npr za jezike C, JAVA, C++, Python
- BProlog, OZ, Claire, Curry

4.4 Skript paradigma

Skript jezici

- Skript jezik je programski jezik koji služi za pisanje skriptova.
- Skript je spisak (lista) komandi koje mogu biti izvršene u zadatom okruženju bez interakcije sa korisnikom.
- U prvobitnom obliku pojavljuju se kao komandni jezici operativnih sistema (npr Bash)
- Skript jezici imaju veliku primenu na Internetu.
- Skript jezici mogu imati specifičan domen primene, ali mogu biti i jezici opšte namene (npr Python)

Skript jezici

- Skript jezici se ne kompiliraju već interpreteraju
- Često se koriste za povezivanje komponenti unutar neke aplikacije
- Omogućavaju kratak kod
- Najčešće nisu strogo tipizirani
- Kôd i podaci često mogu zameniti uloge

Skript jezici

- Nije uvek lako napraviti razliku između skript-jezika i drugih programskih jezika
- Skript paradigma je često specifična kombinacija drugih paradigmi, kao što su: objektno-orijentisana, proceduralna, funkcionalna (pa je to razlog što se skript paradigma ne prepoznaje uvek kao posebna paradigma).
- Skript jezici su u ekspanziji
- Unix Shell (sh), JavaScript, PHP, Perl, Python, XSLT, VBScript, Lua, Ruby...

4.5 Paradigma upitnih jezika

Paradigma upitnih jezika

- Upitni jezici mogu biti vezani za baze podataka ili za pronalaženje informacija (information retrieval)
- Deklarativna paradigma

Upitni jezici baza podataka

- Upitni jezici baza podataka — oni na osnovu strukturiranih činjenica zadatih u okviru strukturiranih baza podataka daju konkretne odgovore koji zadovoljavaju nekakve tražene uslove.
- Najpoznatiji predstavnik upitnih jezika za relacione baze podataka je SQL.
- XQuery je jezik za pretraživanje XML strukturiranih podataka

Upitni jezici baza podataka

- Digresija: Jezici za obeležavanje teksta i programske paradigme
 - Poslednjih decenija veliki procvat doživljavaju jezici za obeležavanje teksta, kao što su: SGML, HTML, XML
 - Jezici za obeležavanje teksta **nisu** programski jezici pa samim tim i ne mogu da generišu neku programsku paradigmu.
 - Međutim, paralelno sa razvojem jezika za obeležavanje (posebno XML), razvijeni su specijalizovani programski jezici za razne obrade koje se odnose na jezike za obeležavanje.
 - U takve jezike spadaju: XSLT, XQuery, XLS, ... Ovi jezici se mogu pridružiti raznim paradigmama.

Upitni jezici za pronalaženje informacija

- Upitni jezici za pronalaženje informacija su upitni jezici koji pronalaze dokumenta koji sadrže informacije relevantne za oblast istraživanja.
- CQL jezik za iskazivanje upita za pronalaženje informacija.

4.6 Reaktivna paradigma

Reaktivna paradigma

- Reaktivno programiranje je usmereno na tok podatka u smislu prenošenja izmena prilikom promene podataka
- Na primer, u proceduralnom programskom jeziku, $a = b + c$ je komanda koja se izvršava dodelom vrednosti promenljivoj a na osnovu trenutnih vrednosti promenljivih b i c i kasnija promena vrednosti b ili c ne utiče na promenu vrednosti promenljive a
- Kod reaktivnog programiranja, $a = b + c$ ima značenje da svaka promena vrednosti b i c utiče na izmenu vrednosti promenljive a

Reaktivna paradigma

- Programiranje u okviru tabela, npr VisiCalc, Excel, LibreOffice Calc
- Jezici za opis hardvera pripadaju ovoj paradigmi, jer se izmena jednog kola u dizajnu propagira na celo kolo — Verilog, VHDL ...

4.7 Vizuelna paradigma

Vizuelna paradigma

- Vršiti modelovanje spoljašnjeg sveta (usko povezana sa objektno-orijentisanom paradigmom).
- Koriste se grafički elementi (dijagrami) za opis akcija, svojstva i povezanosti sa raznim resursima
- Vizuelni jezici su dominantni u fazi dizajniranja programa
- Postoje razne vrste dijagrama: dijagram klasa, dijagram korišćenja, dijagram stanja, dijagram aktivnosti, dijagram interakcija...

Vizuelna paradigma

- Postoje softverski alati za prevođenje “vizuelnog opisa” u neki programski jezik (samim tim i mašinski jezik).
- Pogodnija za pravljenje “skica” programa, a ne za detaljan opis
- Glavni predstavnik ove paradigme je UML.

4.8 Pitanja

Pitanja

- Koje su osnovne karakteristike komponentne paradigme?
- Nabroj tri jezika koji pripadaju (podržavaju) komponentnu paradigmu.
- Koje su osnovne karakteristike ... paradigme?
- Nabroj tri jezika koji pripadaju ... paradigmi.

5 Literatura

Literatura

- Peter Van Roy, Seif Haridi — Concepts, Techniques, and Models of Computer Programming, MIT Press, 2003.
- Deo materijala je preuzet od prof Dušana Tošića, iz istoimenog kursa