Programske paradigme

— Odnos programskih jezika i programskih paradigmi —

Milena Vujošević Janičić

www.matf.bg.ac.rs/~milena

Matematički fakultet, Univerzitet u Beogradu

Pregled

Pregled

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 mašinaki 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?

Programski jezici

- Postoji veliki broj programskih jezika (broji se u hiljadama)
- Enciklopedija britanika pominje preko 2000
- Drugi izvori pominju preko 2500 dokumentovanih programskih jezika (Bill Kinnersley)
- „Encyclopedia of Computer Languages", autor Diarmuid Pigott, sa Murdoch Univerziteta iz Australije navodi preko 8,000 jezika.

Programski jezici

- Naravno, nisu svi programski jezici jednako važni i zastupljeni
- Indikator popularnosti programskih jezika: TIOBE Index
  `https://www.tiobe.com/tiobe-index/`
- Most Popular Programming Languages 1965 - 2019:
  `https://www.youtube.com/watch?v=Og847HVwRSI`

Programski jezici

- Neke liste programskih jezika
  - http://en.wikipedia.org/wiki/List_of_programming_languages
  - http://en.wikipedia.org/wiki/List_of_BASIC_dialects
  - http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm
- Nemoguće je proučiti sve programske jezike

Paradigma — značenje

- Reč paradigma je grčkog porekla i znači
  - primer za ugled,
  - uzor
  - uzorak
  - obrazac
  - šablon
- Obično se koristi da označi vrstu objekata koji imaju zajedničke karakteristike

Programska paradigma

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

Uloga programskih paradigmi

- Broj programskih paradigmi nije tako veliki kao broj programskih jezika
- Izučavanjem programskih paradigmi upoznaju se globalna svojsta 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 znaš, toliko vrediš!

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 znaš, toliko vrediš!

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 paradigmi znaš, toliko vrediš!

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 paradigmi znaš, toliko vrediš!
  Preciznije: Koliko predstavnika različitih paradigmi znaš, toliko vrediš!

Povezanost paradigmi i jezika

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

Pregled

Bitni momenti u razvoju računara

- Jedan od prvih elektrnoskih 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

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 paradgmi 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 paradigma (vrstama programskih paradigmi, njihovom značaju, najistaknutijim programskim jezicima pojedinih paradigmi itd.)
- Moguće su različite podele na programske paradigme.

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
  - Imepretivna 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 svaka paradigma kod koje se u procesu programiranja opisuje algoritam (procedura) rešavanja problema.
     - U ovom slučaju je imperativna paradigma podparadigma proceduralne paradigme
  2. 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.

Programski jezici i pradigme

- Programski jezik je sredstvo koje koristi čovek 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:
  - čovek — logička paradigma
  - proces — funkcionalna paradigma
  - računar — imperativna paradigma
  - problem — objektno-orijentisana paradigma

Programski jezici i pradigme

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

Pregled

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 (zanovan je na algoritamskom načinu rada), kao što je spremanje hrane korišćenjm 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...

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
- Simualcija (modeliranje) spoljašnjeg sveta pomoću objekata
- Objekti interaguju međusobno razmenom poruka
- Mogla bi da se okarakteriše rečenicom:
  "Uputi poruku objektima da bi simulurao tok nekog fenomena"

Objektno-orijentisana paradigma

- Podaci i procedure (funkcije) se učauravaju (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#

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

- Eliminisani 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 bio je programski jezik Lisp, sada je Haskell.
- Nastala pedesetih i početkom šezdesetih godina prošlog veka, stagnacja u razvoju sedamdesetih godina prošlog veka, oživljavanje funkcionane paradigme programskim jezikom Haskell
- Lisp, Scheme, Haskell, ML, Scala, OCaml

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

Pregled

Dodatne programske paradigme

- Skript paradigma
- Paradigma programiranja ograničenja
- Komponentna paradigma
- Paradigma upitnih jezika
- Generička paradigma
- Vizuelna paradigma
- Konkurentna paradigma
- Reaktivna 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.
- Skript jezici mogu imati specifičan domen primene
  - U prvobitnom obliku pojavljuju se kao komandni jezici operativnih sistema (npr Bash)
  - Velika primena u programiranju web aplikacija (JavaScript, TypeScript...).
  - Često se koriste i za povezivanje komponenti unutar neke aplikacije
- Skript jezici mogu biti i jezici opšte namene (npr Python)

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 imaju razne specifične osobine
  - Obično se ne kompiliraju već interpreteraju
  - Obično dinamički određuju tipove
- Skript jezici su u ekspanziji
- Unix Shell (sh), JavaScript, PHP, Perl, Python, XSLT, VBScript, Lua, Ruby...

Paradigma programiranja ograničenja

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

Komponentna paradigma

- Ideja je da se softver sklapa od većih gotovih komponenti, kao što se to radi kod sklapanja elektronskih i tehničkih uređaja
- Komponenta je jedinica funkcionalnosti sa „ugovorenim" interfejsom.
- Interfejs definiše način na koji se komunicira sa komponentom, i on je u potpunosti odvojen od implementacije.
- Približavanje deklarativnom stilu programiranja
- Cilj je da se uprosti proces programiranja i da se jednom kreirane komponente mnogo puta koriste.

Komponentna paradigma

- Komponentna paradigma je nova paradigma ili potparadigma objektno-orijentisane paradigme?
- 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 konjunkciji sa drugim jedinicama.

Komponentna paradigma

- Komponente se međusobno povezuju da bi se kreirao kompleksan softver
- Način povezivanja komponenti treba da bude jednostavan,
- Najjednostavniji pristup je „prevlačenjem" i spuštanjem na željenu lokaciju, tj kada se kreiranje programa vrši biranjem komponenti i postavljanjem na pravo mesto, a ne pisanjem „linije za linijom".
- Primer: kreiranje interfejsa aplikacija (postavljanje prozora, tekstualnih polja, dugmića)
- U okviru komponentnog programiranja, važno je razvojno okruženje koje se koristi, dok sama implementacija komponenti i kôd koji se komponentnim programiranjem generiše može da bude u različitim programskim jezicima, npr JAVA, C++, C# ...

Paradigma upitnih jezika

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

Upitni jezici baza podataka

- Upitni jezici baza podataka:
  Na osnovu struktuiranih činjenica zadatih u okviru struktuiranih baza podataka daju konkretne odgovore koji zadovoljavaju nekakve tražene uslove.

Upitni jezici baza podataka

Najpoznatiji predstavnik upitnih jezika je SQL koji se koristi za relacione baze podataka (select, from, where)
Iz tabele Studenti izaberi sva imena studenata koji se prezivaju Petrovic

```
SELECT Ime
FROM Studenti
WHERE Prezime='Petrovic';
```

https://www.w3schools.com/sql/

Upitni jezici baza podataka

SPARQL je jezik koji se koristi za podatke u RDF (*Resource Description Framework*) formatu (select, from, where)
Izdvoj sve studente osnovnih studija, kurseve koji oni slusaju i njihova imena

```
select ?x ?y ?n
where {
?x a :UndergradStudent .
?x :takesCourse ?y .
?x :name ?n
}
```

https://www.w3.org/TR/sparql11-query/

Upitni jezici baza podataka

XQuery je jezik za pretraživanje XML struktuiranih podataka (for, let, where, order by, return)
Za svaku knjigu iz prodavnice za koju se informacije čuvaju u okviru dokumenta books.xml, i za koju važi da je skuplja od 30, vrati njen naslov u sortiranom redosledu po naslovima

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

https://www.w3schools.com/xml/xquery_intro.asp

Upitni jezici za pronalaženje informacija

- Upitni jezici za pronalaženje informacija su upitni jezici koji pronalaze dokumenta koji sadrže informacije relevantne za oblast istraživanja.
- CQL (engl. *Contextual Query Language*) jezik za iskazivanje upita za pronalaženje informacija.
  - Formalni jezik za predstavljanje upita za izvlačenje informacija u sistemima kao što su web indeksi, bibliografski katalozi, informacije o muzejskim zbirkama.
  - Dizajn jezika podrazumeva da su upiti takvi da se mogu lako čitati i pisati, i da je i jezik intuitivan pri čemu zadržava ekspresivnost kompleksnijih upitnih jezika

```
title any fish
title any fish sortBy date/sort.ascending
title any fish or creator any sanderson
```

`https://www.loc.gov/standards/sru/cql/`

Generička paradigma

- Generičko programiranje je stil programiranja u kojem se algoritmi pišu sa apstrahovanim tipovima
- Tipovi se obezbeđuju kao parametri i time se izbegava dupliranje koda
- Na primer, algoritmi traženja minimuma, maksimuma, sortiranja, pretrage...
- Ova ideja je nastala u jeziku ML 1973 godine
- *Generics* u jezicima Pajton, Ada, C#, F#, Java, Nim, Rust, Swift, TypeScript, Visual Basic...
- Šabloni (engl. *templates*) u jezicima C++ i D
- Implicitni polimorfizam u jezicima ML, Scala, Julia, Haskell

Vizuelna paradigma

- Vrši 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

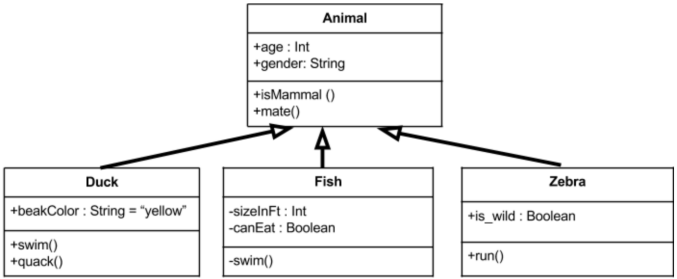
Vizuelna paradigma

- Glavni predstavnik ove paradigme je UML (engl. *Unified Modeling Language*) `https://www.uml.org/`.
- Pogodnija za pravljenje „skica" programa, a ne za detaljan opis
- Postoje softverski alati za prevođenje „vizuelnog opisa" u neki programski jezik (samim tim i u izvršivi kôd).
- Postoji 14 vrsta različitih dijagrama, 7 koji odgovaraju strukturnim informacijama o programu i 7 koji odgovaraju ponašanju programa

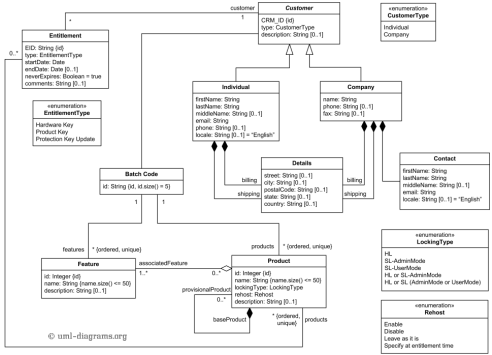
Vizuelna paradigma

- Primeri strukturnih informacija:
  - dijagram klasa,
  - dijagram objekata,
  - dijagram komponenti.
- Primeri informacija o ponašanju:
  - dijagram komunikacije,
  - dijagram aktivnosti,
  - dijagram slučajeva upotrebe.

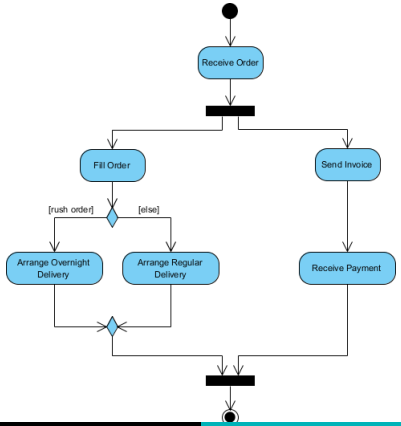
Dijagram klasa



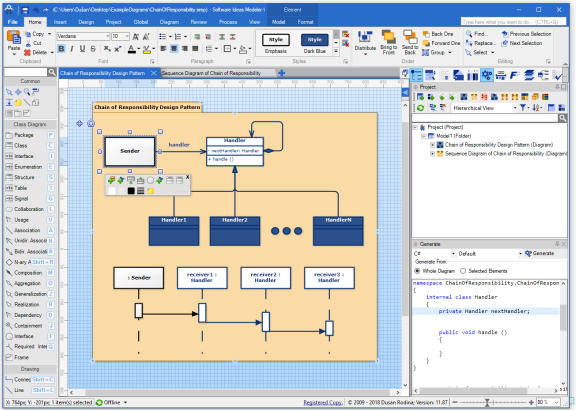
Dijagram klasa



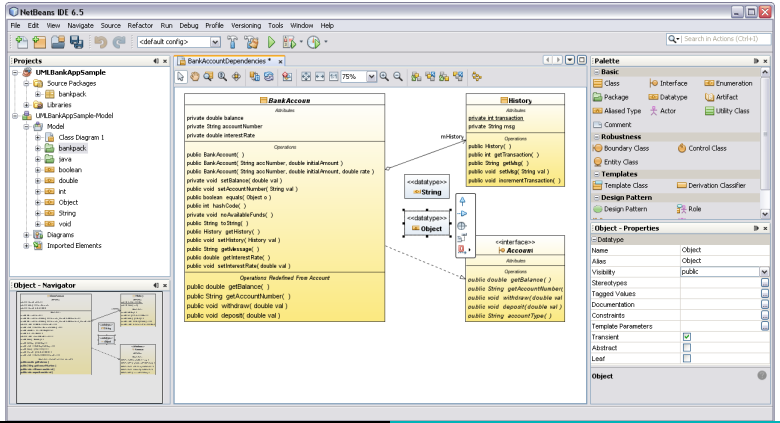
Dijagram aktivnosti



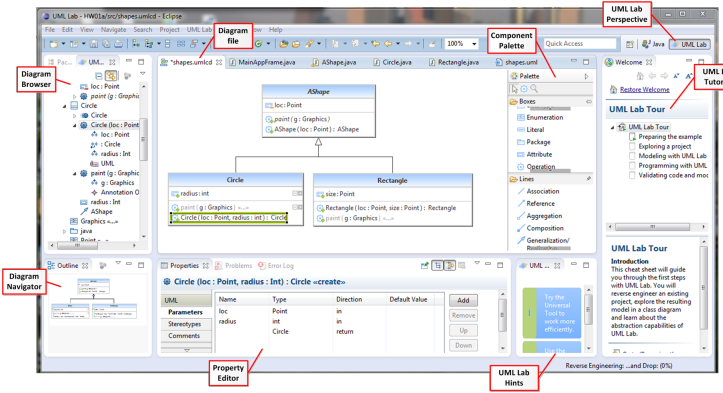
Primer razvojnog okruženja



Primer razvojnog okruženja



Primer razvojnog okruženja



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 modeluju 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
- Procesi međusobno mogu da komuniciraju preko zajedničke memorije

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.
- Pristuno je u gotovo svim savremenim programskim jezicima: Ada, Modula, ML, Java, C, C++, Haskell ...

Reaktivna paradigma

- Rekativno programiranje je usmereno na tok podatka u smislu prenošenja izmena prilikom asinhrone promene podataka
- Na primer, u imperativnom 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 (kao što je to u okviru tabela, npr VisiCalc, Excel, LibreOffice Calc)

Reaktivna paradigma

- Reaktivnoj paradigmi izuzetno raste popularnost: poznavanje je osnovno za pisanje
  - veb servisa,
  - mobilnih aplikacije,
  - sistema sa real-time komponentama

Reaktivna paradigma

Osnovne karakteristike reaktivnih aplikacija:

- Reaguje na događaje — vođena događajima (engl. *event-driven*)
- Reaguje na rast upotrebe — skalabilna (engl. *scalable*)
- Reaguje na padove — mogućnost brzog oporavka od pada (engl. *resilient*),
  - Pad može biti softverski izuzetak, hardverski pad, pad veze
  - Da bi se to ostvarilo, mora da bude sastavni deo arhitekture sistema (međusobno slabo zavisne komponente)
- Reaguje na zahteve korisnika — brza reakcija na korisničke zahteve (engl. *responsive*), čak i kada je opterećenje sistema veliko i postoje padovi.
  - Grade se nad *event-driven*, *scalable* i *resilient* arhitekturama ali mora da se vodi računa o algoritmima, dizajnu sistema i puno drugih detalja

Reaktivna paradigma

Reaktivne apstrakcije:

- Apstrakcija događaja — *futures*
- Apstrakcija tokova događaja — *observables*
- Arhitektura slanja poruka — čvorovi ove arhitekture su *actors*
- Praćenje padova — koncept *supervisors*
- Skaliranje korišćenjem distribuiranih sistema — *distributed actors*

Reaktivna paradigma se zasniva na veoma naprednim konceptima
`https://www.youtube.com/watch?v=7D9QfMj_KwI`

Reaktivna paradigma

Reaktivno programiranje nije vezano za programski jezik, već je to stil programiranja koji se može ostvariti u najrazličitijim programskim jezicima:

- Reactive Design Patterns
- Functional Reactive Programming - Manning
- Reactive Programming With Java 9: Build Asynchronous applications with Rx.Java 2.0,
- C++ Reactive Programming: Design concurrent and asynchronous applications using the RxCpp library and Modern C++17
- RxSwift: Reactive Programming with Swift
- Scala Reactive Programming: Build Scalable, Functional Reactive Microservices with Akka, Play, and Lagom
- Hands-On Reactive Programming in Spring 5: Build Cloud-ready, Reactive Systems with Spring 5 and Project Reactor
- Reactive Programming in Kotlin: Design and Build Non-blocking, Asynchronous Kotlin Applications with RXKotlin, Reactor-Kotlin, Android, and Spring

Pregled

Jezici za obeležavanje teksta/podataka i programske paradigme

- Postoje jezici za obeležavanje/struktuiranje teksta/podataka, kao što su: HTML, CSS, XML, SGML, JSON...
- Jezici za obeležavanje teksta/podataka nisu programski jezici jer se ne uklapaju u definiciju programskih jezika
- Naime, po definiciji, programskim jezikom se zadaje nakakvo izračunavanje odnosno definišu se programi koje računar može da izvrši
- Ovim jezicima definiše se struktura teksta/podataka, ne pišu se programi i nema nikakvog izvršavanja
- U skladu sa time, ovi jezici nisu programski jezici pa samim tim ne mogu da formiraju programsku paradigmu

Jezici za obeležavanje teksta/podataka

- Na primer, HTML definiše strukturu veb stranice i određuje naslove, paragrafe i slično. Brauzer na osnovu tih informacija (i eventulano dodatnih stilskih informacija kroz CSS) određuje kako će podaci biti prikazani.
- Dakle, HTML se koristi za određivanje strukturnih osobina veb stranice, a ne njene funkcionalnosti. Programski jezici imaju funkcionalnu namenu.
- HTML, kao jezik za obeležavanje, ne može da uradi ništa u smislu u kojem programski jezici to rade jer ne sadrži logiku programiranja: ne sadrži kontrolu toka, evaluaciju izraza, funkcije, promenljive, pojam događaja i slično... Ne može da modifikuje podatke niti da ima ulaz/izlaz...

Jezici za obeležavanje teksta/podataka

- 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. Na primer, to su najčešće domenski specifični jezici i skript jezici.

Pregled

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?

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.

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