Dizajn programskih jezika

Milena Vujošević Janičić

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

Dizajn programskih jezika
Beograd, 11. oktobar, 2016.

Pregled

1. Dodatne paradigme
2. Pitanja i literatura

Pregled

1. Dodatne paradigme
   - Komponentna paradigma
   - Konkurentna paradigma
   - Paradigma programiranja ograničenja
   - Skript paradigma
   - Paradigma upitnih jezika
   - Reaktivna paradigma
   - Vizuelna paradigma
2. Pitanja i literatura

Dodatne programske paradigme

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

Komponentna paradigma

- Ideja je da se softver sklapa 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 konjunkciji 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# ...

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
- Procesi 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 ...

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

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...

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 zadovoljavanju 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 programski 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.

Reaktivna paradigma

- Rekativno 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 ...

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
- 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.

Pregled

1 Dodatne paradigme

2 Pitanja i literatura
- Pitanja
- Literatura

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.

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