



Forum: Programiranje

Topic: C, C++, C#, Python

Subject: Re: C, C++, C#, Python

Poslao: slobodan

Postano : 31.05.2006 9:55:32

Ona dva primjera za C i C++ koja si stavio su bezveze, jer su prejednostavni i u njima se uopæe ne vide moguænosti pojedinog jezika. Potrebno je dosta razumijevanja pojedinog jezika da bi razumio temeljne razlike izmeðu cout i printf.

Objektno programiranje je naèin razmi¹ljanja koji ka¾e da se svaki problem mo¾e utrpati u objekt koji ima neke svoje karakteristike te funkcije koje rade na tim karakteristikama. U dijelu problema je to toèno, no objektno programiranje nije rije¹enje za sve.

O tome koja sve "razmi¹ljanja" (odnosno programing paradigms) postoje, mo¾e¹ vidjeti [ovdje](#). Ne postoji univerzalna metoda, tako da ovisno o problemu treba primjeniti odgovarajuæu metodu ili èak i programski jezik, jer su razlièiti programski jezici prilagoðeni razlièitim metodama. Tako je C proceduralni jezik, C++ objektni (IMHO bastard proceduralnog i objektnog koncepta, pravi èisti objektni jezik je smalltalk), C# je MSova reakcija na javu a oba su objektna, a python je funkcionalni jezik.

Naravno, u svakom od tih programskih jezika mo¾e¹ osim njegovog glavnog "razmi¹ljanja" koristiti i druge, ali u pravilu znatno te¾e. AFAIK ako ba¹ %eli¹ jezik koji brije po èim vi¹e razlièitih "razmi¹ljanja", onda posegni za lisp grupacijom jezika (Scheme, Common Lisp...), IMHO naravno...

©to se C++ knjiga tièe, na netu ih ima hrpa, ali... su sve na engleskom. Inaèe, znanje je uvijek skupo, a takva je i situacija sa struènim knjigama, no s druge strane pogledaj po gradskim knji%nicama, vjerojatno neka ima tu knjigu.

Garbage collection je sustav upravljanja memorijom. U nekim programskim jezicima mora¹ sam paziti na to gdje se i na koji naèin nalaze podaci kojima u danom trenutku èaèka tvoj program (neki od takvih su Pascal, C, C++), a u drugim jezicima se sam jezik brine o tome (lisp jezici su to prvi uveli prije cca 20+ godina, a u dana¹nje vrijeme to imaju i mlaði jezici kao ¹to su java i C#). Prednost garbage collection sustava je da programer vi¹e ne mora brinuti oko raznih memory leak problema jer sam jezik brine o dostupnosti podataka i memorije, a mana je ¹to su to "univerzalne" implementacije upravljanja memorijom u ponekim okolnostima sporije no ¹to bi to bilo da se takav sustav implementira za specifiène potrebe.

Ni C ni C++ nisu bolji, nego je svaki dobar za ne¹to. I C i C++ mogu èaèkati direktno po hardveraju, ali C++ zbog objektnih elemenata uvodi dosta overheada pri radu sa funkcijama i sliènim internim stvarima, pa je u neku ruku sporiji. Ali nema neke specijalne prepreke da se ne mo¾e i u njemu pisati kernel. Osim naravno volje samih programera...

©to se GUI-a tièe, zgodniji je C++ zato jer je GUI po svom konceptu bli¾i ideji objekata kojima se manipulira. No gtk i Gnome su pokazatelj da to nije neka specijalna prepreka (oba su pisana u C, ali

se u kodu vidi da su programeri ipak implementirali svoju varijantu objektne metodologije...).