Objektno orijentisani koncepti, PHP
Vrsta: Seminarski | Broj strana: 24

UVOD
2. Objektno-orijentisano programiranje
Rešavanje problema paradigmom objektno-orijentisanog programiranja, je vrlo slično ljudskom načinu razmišljanja i rešavanju problema. Sastoji se od identifikovanja objekata i postavljanje objekata koji će se koristiti u odgovarajuću sekvencu za rešenje određenog problema. Radi se o dizajnu objekata čija će ponašanja kao jedinica i u njihovoj međusobnoj interakciji, rešiti određeni problem. Interakcija između objekata se sastoji u razmeni poruka, gde određena poruka usmerena prema određenom objektu, pokreće enkapsulirane operacije u tom objektu, čime se rešava deo obično šireg i složenijeg problema.
Uopšteno gledano, objektno-orijentisano rešavanje problema se sastoji iz četiri koraka:
identifikovanje problema
identifikovanje objekata koji su potrebni za njegovo rešenje
identifikovanje poruka koje će objekti međusobno slati i primati
kreiranje sekvence poruka objektima, koje će rešavati problem ili probleme.

2.1. Osnovni principi - Objektno orijentisano programiranje
Strukturnim programiranjem težimo da što realističnije preslikamo (modeliramo, simuliramo) na računaru ponašanje nekog sistema. Objektno-orijentisanim programiranjem (oznaka OOP) preslikavamo isti taj sistem tako što ga predstavljamo kao skup medjusobno povezanih objekata. Naime, ceo sistem se strukturira na manje celine kojima je lakše upravljati, a koje medjusobno komuniciraju. Sistem koji je ovako organizovan lakši je za razumevanje, njime je lakše upravljanje i rad.
Smatra se da je složenost problema koje je neko u stanju da reši derektno srazmerna predmetu (šta?) i kvalitetu (kako?) apstrakcije kojom raspolaže.
Svi programski jezici obezbeđuju podršku određenim apstraktnim kategorijama.
Asemblerski jezik poseduje nizak nivo apstrakcije
.....NAMERNO UKLONJEN DEO TEKSTA.....
g modela mašine) i domena problema (apstraktnog modela realnog sveta).
Napor koji je neophodno uložiti u ovo preslikavanje, a koji u jednom delu leži u samoj strukturi korišćenog programskog jezika, često nema adekvatnu valorizaciju budući da rezultira programima niskog stepena semantike.
Alternativa prethodno opisanom pristupu je modeliranje problema koji se rešava.
Prvi programski jezici koji su razvijeni na problemskoj orijentaciji uvode specifične apstrakcije realnosti na kojima zasnivaju postupak preslikavanja domena problema u domen rešenja.
Tipični primeri su sledeći pristupi:
Sve probleme je moguće apstrahirati listom i operacijama nad njom (LISP – LISt Processing).
Svi problemi su algoritamske prirode (APL-Algoritmic Programming Language).
Svi problemi se daju iskazati kao lanci odlučivanja (PROLOG – PROgramming LOGic).
Svi problemi se mogu iskazati kao skup apstraktnih ograničenja i manipulacija sa njima (Constraint Based Programming).
Svi navedeni pristupi predstavljali su dobra rešenja za određenu usku klasu problema za koju su objektivno i dizajnirani, no svaki pokušaj generalizacije iskoraka van inicijalnog domena gotovo bez izuzetka rezultira neuspehom.