

Jezik C++

Vrsta: Skripta | Broj strana: 174 | Nivo: Pmf, Niš

Osnovni pojmovi

OOP uvodi drugačiji način razmišljanja u programiranju! U OOP-u, mnogo više vremena troši se na projektovanje, a mnogo manje na samu implementaciju (kodovanje). U OOP-u se razmišlja o delovima sistema (objektima) koji nešto rade, a ne o tome kako se nešto radi (algoritmi). Drugim rečima, OOP prvenstveno koristi objektnu dekompoziciju umesto isključivo algoritamske dekompozicije.

U osnovi objektno-orijentisanog programiranja leži ideja da se u jednoj strukturi podataka objedine i podaci i akcije koje se sprovode nad tim podacima (to su metodi u terminologiji OOP). Ključni pojmovi u objektno-orijentisanim programskim jezicima jesu:

- klase,
- objekti, i
- poruke.

Klasama se opisuje struktura i ponašanje pojedinačnih primeraka klase. Pojedinačni primerci (instance) klase, nazivaju se objekti. Svi objekti koji pripadaju jednoj klasi imaju strukturu koja je definisana klasom kojoj pripadaju, i nad njima se mogu izvršavati samo operacije koje su definisane tom klasom. Klase se mogu posmatrati kao šabloni za kreiranje novih objekata. U tom smislu je pojam klase sličan pojmu tipa. Pripadnost klasi je svojstvo objekata, dok je pripadnost tipu svojstvo promenljivih i izraza. Svaka klasa se može posmatrati kao celina u kojoj su pored polja u kojima su opisani podaci (svojstva) opisana i polja koja predstavljaju funkcije i/ili procedure. Ovakve funkcije, odnosno procedure, definisane unutar klasa, nazivaju se metodi klase.

Svojstva objekata menjaju se pozivom odgovarajućih metoda. Pozivi metoda nazivaju se poruke. Objekat odgovara na poruku tako što izvršava odgovarajući metod. Sintaksa poruka zavisi od konkretnog programskog jezika, ali je generalno slična pozivu procedura i funkcija u proceduralnim programskim jezicima.

Program u proceduralnim programskim jezicima se može posmatrati kao skup potprograma (procedura i funkcija) kojima se obrađuju podaci, dok program u objektno-orijentisanim programskim jezicima predstavlja skup objekata koji deluju jedni na druge slanjem poruka.

Objektno-orijentisani programski jezici se mogu implementirati statički i dinamički. Dinamičku implementaciju poseduju netipizirani jezici (kakav je na primer SMALLTALK). U ovakvim programskim jezicima ne koristi se informacija o tipovima podataka, već samo informacija o pripadnosti objekta određenoj klasi. Informacija o pripadnosti objekta određenoj klasi koristi se tek u fazi izvršavanja programa. Većina objektno-orijentisanih jezika je implementirana statički. Kod ovakvih programskih jezika, informacija o pripadnosti objekta određenoj klasi koristi se u fazi kompilovanja programa, na sličan način kao što se kod proceduralnih programskih jezika koristi informacija o tipu promenljive.

Za razumevanje klasa i metoda veoma su važni sledeći pojmovi:

- apstraksija i enkapsulacija,
- nasleđivanje, i
- polimorfizam.

Tip koji modeluje neki pojam iz problema koji se rešava pomoću (neodvojive veze) podataka i pridruženih operacija naziva se apstraktni tip podataka (abstract data type). Sakrivanje interne reprezentacije tipa, tj. detalja njegove implementacije od korisnika tog tipa naziva se enkapsulacija (encapsulation). Apstrakcija i enkapsulacija se mogu realizovati pomoću klase.

...

**----- OSTATAK TEKSTA NIJE PRIKAZAN. CEO RAD MOŽETE
PREUZETI NA SAJTU. -----**

www.maturskiradovi.net

MOŽETE NAS KONTAKTIRATI NA E-MAIL: maturskiradovi.net@gmail.com