
Zašto OOP?

- ❖ Problemi u razvoju softvera:
 - Zahtevi korisnika su složeni i stalno se povećavaju
 - Softverski sistemi su inherentno složeni
- ❖ Uvek je potrebno povećati produktivnost proizvodnje softvera. Kako? Povećanjem broja programera u timu? Problemi – interakcija između delova softvera!
- ❖ Način povećanja produktivnosti – ponovna upotreba softvera (*software reuse*). Kako obezbediti?
- ❖ Problemi održavanja softvera: ispravljanje grešaka, promena zahteva i dodavanje zahteva. Kako postići?
- ❖ Kako odgovoriti na izazove? Unapređenjem koncepata!

Šta donosi OOP?

OOP je deo objektne paradigme koja obuhvata osnovne objektne koncepte, od kojih su neki:

- ❖ apstraktni tipovi podataka (*abstract data types*): tip koji je definisao programer, za koji se mogu kreirati primeri (instance) i koji je predstavljen strukturom i ponašanjem, s tim da je za korisnike tipa bitno samo ponašanje, ne i interna struktura (implementacija)
- ❖ enkapsulacija (*encapsulation*): deo softvera ima jasno definisan interfejs i implementaciju; interfejs je svima dostupan, implementacija je nedostupna
- ❖ nasleđivanje (*inheritance*): jedan tip može da nasledi drugi, osnovni tip, sa značenjem da su njegove instance jedna vrsta instanci tog osnovnog tipa
- ❖ polimorfizam (*polymorphism*)