
Klasa kao realizacija apstrakcije

- ❖ Klasa može predstavljati *logički* koncept (apstrakciju) iz *domena problema* (uključujući i generalizaciju ili specijalizaciju)

Nekoliko primera:

- ❖ *Telefonski poziv (Phone call), Pozivalac (Caller), Pozvani (Callee)*
- ❖ *Grafička kontrola (GUI Control), Dugme (Button), Tekstualno polje (Textbox), ...*
- ❖ *Porudžbina (Order), Isporučka (Delivery), Račun (Invoice), Plaćanje (Payment)*
- ❖ *Zahtev za izdavanje ličnog dokumenta (Application for ID Document), Odobrenje (Approval), Izdavanje (Issuance)*
- ❖ i mnogo, mnogo drugih

Klasa kao realizacija apstrakcije

- ❖ Klasa može realizovati i apstrakciju *osmišljenu unutar samog rešenja* datog problema
- ❖ Takav koncept može da pomogne u rešavanju problema, pojednostavi sistem i njegovo korišćenje, uvede red ili uopštenje u domen problema
- ❖ Stoga takav koncept postaje “vidljiv” za korisnike sistema, jer softver “otkriva” taj koncept ili neposredno, nudeći ga korisnicima ekslicitno kao pojam, ili posredno, kroz ponašanje softvera
- ❖ Na neki način, tako osmišljen koncept može da postane i sastavni deo domena problema, jer je posredno “otkriven”, iako nije inicijalno postojao (ili nije bio eksplicitno prepoznat) u domenu problema
- ❖ Primeri:
 - *Šablon (Template)* slajda, tekstualnog dokumenta i slično
 - *Prečica (Shortcut)* do neke složenije operacije ili kompozicija operacija koje se često ponavljaju
- ❖ Sa druge strane, tako osmišljen koncept može biti isključivo deo rešenja, odnosno implementacije, bez ikakve vidljivosti spolja, jer se je nastao, na primer:
 - kao deo implementacije neke složene strukture, protokola ili mehanizma sistema
 - kao deo implementacije neke druge apstrakcije
 - kao posledica sprovođenja nekog *projektnog obrasca (design patter)*
 - kao posledica sprovođenja neke tehnike implementacije, optimizacije i slično