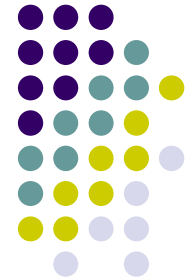




Razvoj softvera i dizajn

- Dizajn više razine
 - specifične tehnike određenog programskog jezika (C++)
 - koncepti koji podupiru određeni dizajn
- Veza između dizajna neovisnog o jeziku i programiranja koje se nužno bavi nekim tehničkim detaljima
- Uloga klasa u organizaciji softvera u fazi dizajna



Razvoj softvera:

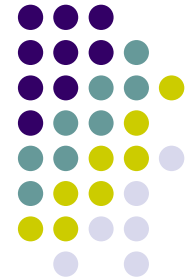
- analiza problema
 - ukupni razvoj programa (dizajn i **programiranje**)
 - dokumentiranje
 - testiranje
 - održavanje
 - koordiniranje svih aktivnosti
-
- **Programiranje** – samo dio procesa izrade softvera
 - Ne postoji univerzalna tehnika razvoja SW pogodna za sve probleme
 - Objektno orijentirani dizajn – apstrakcija podataka i objektno orijentirani pristup

Karakteristike razvoja softvera



- Jasno odrediti što je potrebno napraviti
- Razvoj softvera je dugotajan proces
- Sustavi koji se razvijaju su obično na granici kompleksnosti raspoloživih alata i ljudskih resursa
- Potrebno uložiti inteligenciju, iskustvo, osjećaj za dizajn i programiranje
- Eksperimentiranje je nužno za svaki netrivialni razvoj softvera
- Dizajn i programiranje su iterativne aktivnosti
- Različite faze u razvoju softvera ne mogu se strogo odijeliti
- Programiranjem i dizajnom potrebno je upravljati

Kompleksnost u razvoju softvera



- “Podijeli pa vladaj” (npr. jezgra i sučelje u Mathematici)
- Podjela problema na dijelove i definiranje sučelja
- **Separacija** na dijelove – jednostavno
- **Komunikacija** među komponentama - složeno



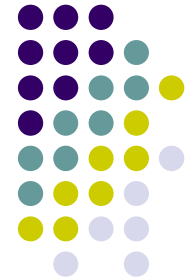
Dvije strane softvera

- **Vanjska** strana (vidljiva korisniku)
 - korisnika ne zanima detaljni dizajn sustava već njegova funkcionalnost
 - u određenim slučajevima (npr. programske biblioteke) korisno je da korisnik razumije djelovanje sustava u svrhu boljeg korištenja
- **Unutarnja** strana (vidljiva dizajnerima i programerima) mora omogućavati
 - testiranje
 - prijenos na druge platforme
 - održavanje
 - reorganizaciju
 - razumijevanje načina rada

Proces razvoja softvera



- Iterativan rastući proces
 - analiza – definiranje problema kojeg treba riješiti
 - dizajn – izrada cjelokupne strukture sustava
 - **implementacija** – pisanje i testiranje koda
- Dijelovi procesa koji se stalno provode
 - eksperimentiranje
 - testiranje
 - analiza dizajna i implementacije
 - dokumentacija
 - održavanje



Razvojni ciklus

- Razmatranje problema
- Izrada globalnog dizajna
- Pronalaženje standardnih komponenti
 - prilagodba komponenti konkretnom dizajnu
- Razvoj novih komponenti
- Povezivanje komponenti



Zadaće dizajna

- Sustav mora biti dako oblikovan da ostaje što je moguće jednostavniji i nakon niza promjena.
- Dizajn mora uzeti u obzir
 - fleksibilnost
 - proširivost
 - prenosivost
- U objektnom programiranju ovo znači dobro definirati metode (članske funkcije) i svojstva objekata



Koraci dizajna

- Rijetko se oblikuje pojedinačna (izolirana) klasa. Obično se istovremeno razvija skup povezanih klasa – **biblioteka klasa** (*class library*) odnosno **komponenta**
- Elementi komponente obično imaju zajednički
 - dizajn
 - dokumentaciju
 - vlasnika