

Glava 5: Objektna dekompozicija

- ❖ Karakteristike lošeg i dobrog softvera
- ❖ Raspodela odgovornosti
- ❖ Algoritamska dekompozicija
- ❖ Relacije i zavisnosti između klasa
- ❖ Enkapsulacija
- ❖ Hijerarhijska dekompozicija



Karakteristike lošeg i dobrog softvera

Karakteristike lošeg softvera:

- ❖ *Rigidnost (rigidity)* — teško ga je promeniti, jer svaka promena jednog utiče na suviše drugih delova softvera:
 - delovi softvera su međusobno jako zavisni, pa svaka promena u jednom zahteva promene u drugim delovima
 - zato je efekte i troškove promene teško predvideti, programeri ne mogu da ih procene, pa rukovodioci teško odobravaju promene
- ❖ *Krhkost (lomljivost, fragility)* — tendencija da promena u softveru izaziva neočekivane probleme i otkaze u mnogim delovima tog softvera:
 - zbog jake isprepletanosti i zamršenosti, svaka promena u jednom delu izaziva problem u drugom - *efekat domina (domino effect)*
 - problemi su često u delovima koji nemaju konceptualne veze sa onim na koje se promena odnosi
 - degradira se kredibilitet softvera i poverenje u njegov kvalitet
- ❖ *Neprenosivost (immobility)* — delove jedne aplikacije teško je ponovo upotrebiti u drugoj, jer se ne mogu rasplesti od ostatka koji nije potreban, opet zbog prevelikih zavisnosti:
 - napor potreban za takvo raspetljavanje je prevelik, pa se lakše odlučuje za pravljenje istih stvari iznova