



# Testiranje softvera

ETF BEOGRAD, 2019/2020.

VEŽBE #9, ASISTENT: DR DRAŽEN DRAŠKOVIĆ

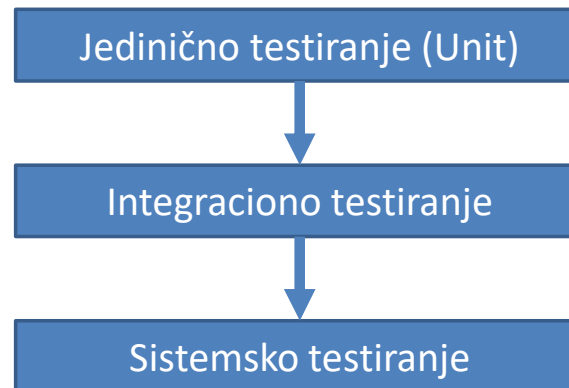


# Integraciono testiranje

*INTEGRATION TESTING*

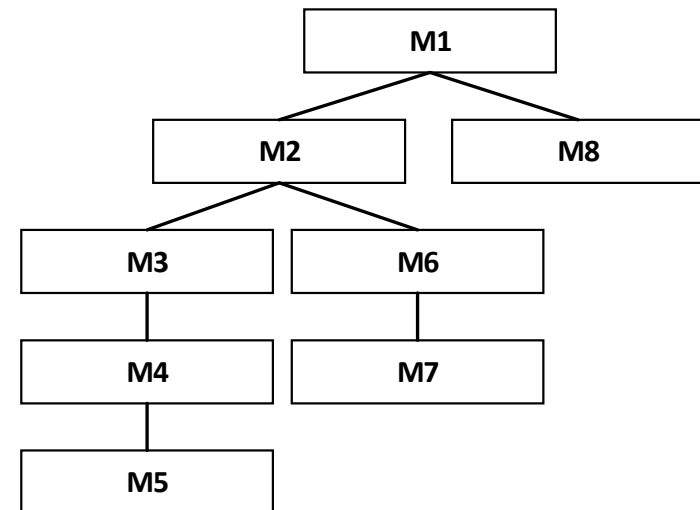
# Integraciono testiranje

- ▶ Testiranje gde se individualne celine programskog koda kombinuju i testiraju kao grupa
- ▶ Svrha ovog nivoa testiranja: otkriti potencijalne probleme i greške u interakciji između integrisanih celina



# Vrste integracionog testiranja

- ▶ Vrste integracionog testiranja, zasnovane na dekompoziciji (eng. *Decomposition-Based Integration*):
  - ▶ Integracija od vrha ka dnu (eng. *Top-down testing*)
  - ▶ Integracija od dna ka vrhu (eng. *Bottom-up testing*)
  - ▶ Mešovita integracija (eng. *Sandwich testing*)
  - ▶ Integracija po principu „velikog praska“ (eng. *Big Bang*)

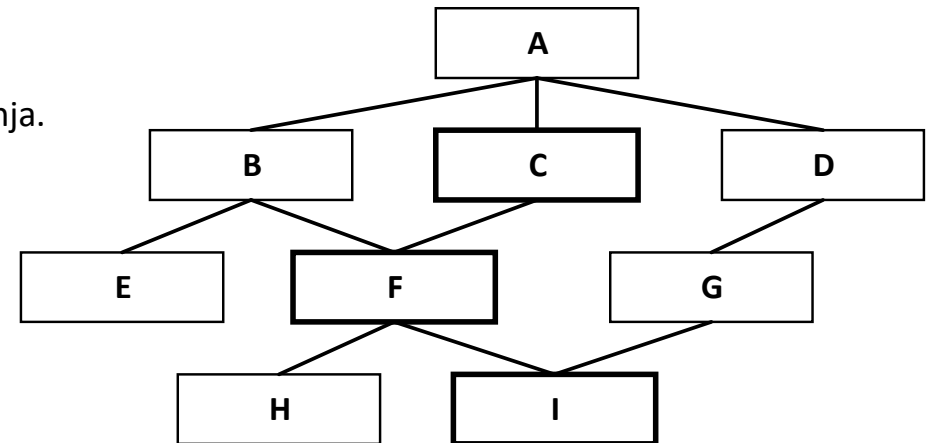


# Integracija od vrha ka dnu

- ▶ Identifikujemo kontrolnu hijerarhiju, počevši od glavnog kontrolnog modula (glavnog programa)
- ▶ Aktivnosti:
  - ▶ 1) Glavni modul se koristi kao pokretač testova, a umesto svih njegovih sledbenika u grafu programa koriste se njihovi stabovi (eng. *stub*).
  - ▶ 2) Izvršava se odgovarajući skup testova
  - ▶ 3) Prema vrsti testiranja (po širini ili po dubini), jedan od stabova se zamenjuje stvarnim modulom
  - ▶ 4) Po potrebi se sprovodi regresivno testiranje
  - ▶ 5) Ponavljaju se koraci od 2. do 5. sve dok se kompletna programska struktura ne integriše i istestira.

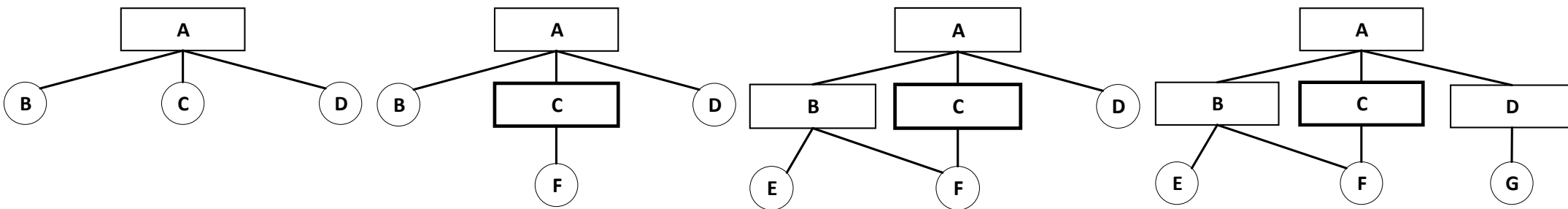
# Zadatak 1 - Integracija od vrha ka dnu

- ▶ Sistem sa slike integraciono se testira pristupom od vrha ka dnu, u nizu koraka pri čemu se u svakom dodaje novi modul već postojećim. Podebljano su označeni kritični moduli. Skicirati šta tačno sadrži deo sistema koji se testira (koje module i stabove) u 4. koraku testiranja, ukoliko se primenjuje strategija:
  - ▶ po širini;
  - ▶ po dubini;
  - ▶ navesti karakteristike kritičnih modula kod integracionog testiranja.



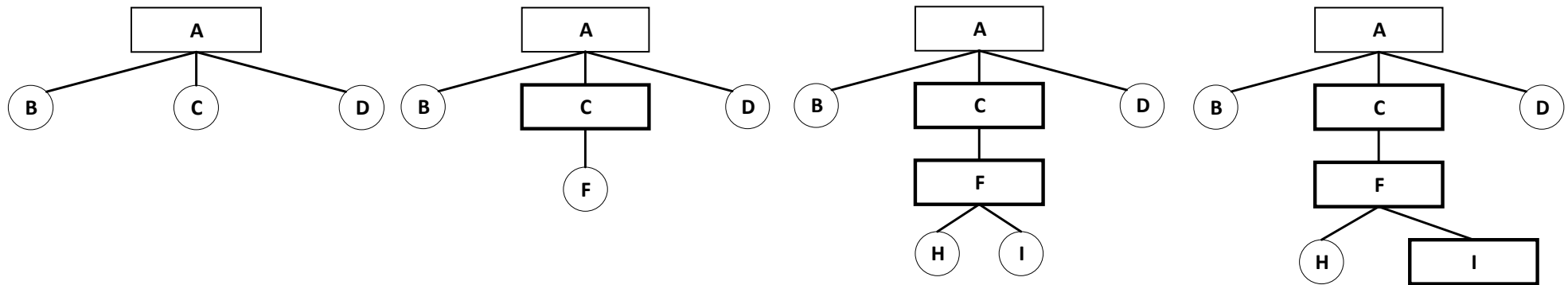
# Zadatak 1 - Integracija od vrha ka dnu - Rešenje (a)

- ▶ Metoda po širini: integracija ide po nivoima programske strukture, prvo se integrišu sve komponente na nivou N, pa se potom prelazi na nivo N+1, N+2 i tako redom do kraja.
- ▶ Pravougaonici = IMPLEMENTIRANI MODULI, Krugovi = STABOVI (eng. *stub*)



# Zadatak 1 - Integracija od vrha ka dnu - Rešenje (b)

- ▶ Metoda po dubini: integracija ide po glavnoj kontrolnoj strukturi koja obuhvata kritične module A-C-F-I.





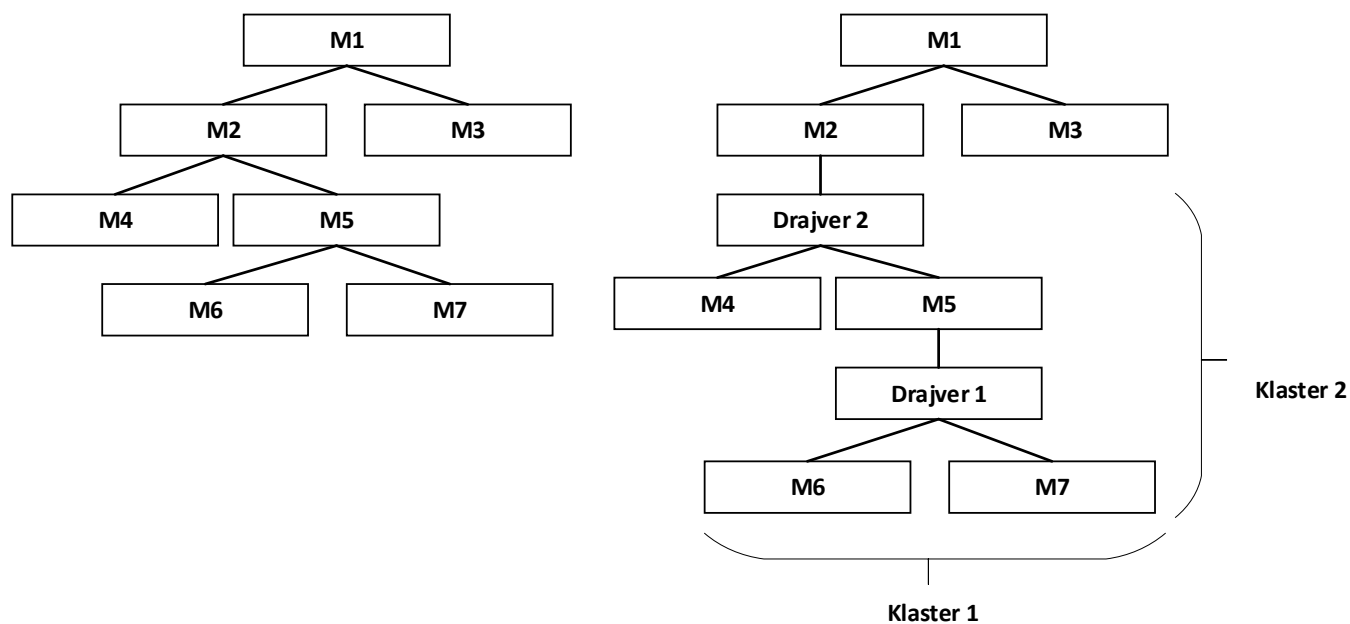
# Zadatak 1 - Integracija od vrha ka dnu - Rešenje (c)

- ▶ Kritični modul (koji treba testirati što je pre moguće) poseduje jednu ili više od sledećih karakteristika:
  - ▶ zadovoljava veći broj funkcionalnih zahteva;
  - ▶ poseduje visok stepen kontrole (nalazi se relativno visoko u programskoj strukturi);
  - ▶ kompleksan je ili podložan greškama (kao indikator može se koristiti ciklomatska kompleksnost);
  - ▶ zahtevane su konačne performanse

# Integracija od dna ka vrhu (1)

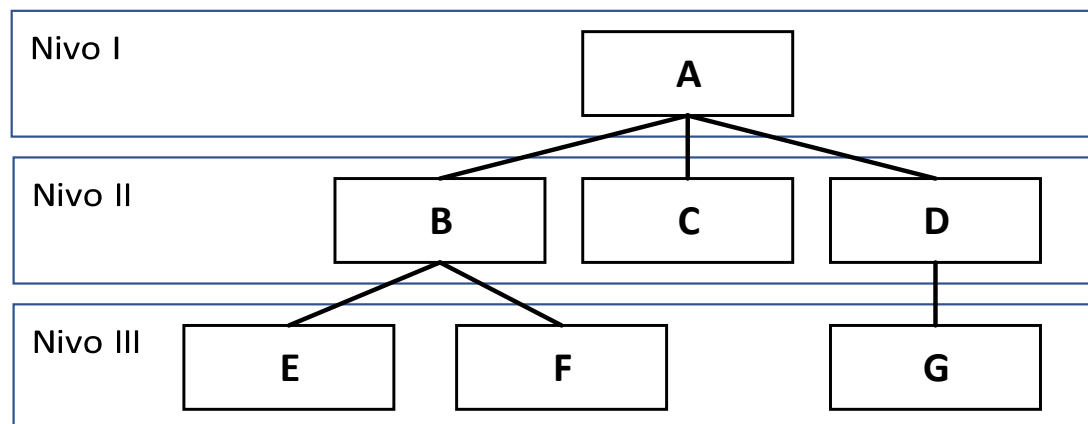
- ▶ Integracija od dna ka vrhu (eng. *Bottom-up testing*) podrazumeva konstrukciju i testiranje počevši od najnižih modula u hijerarhiji
- ▶ Aktivnosti:
  - ▶ 1) Moduli na nižem nivou se kombinuju u klaster (eng. *cluster*) koji vrše neku podfunkciju softverskog sistema. Nazivaju se još i grozdovi.
  - ▶ 2) Pravi se drajver (kontrolni program za testiranje) koji koordiniše testove na ulazu i izlazu.
  - ▶ 3) Testira se klaster.
  - ▶ 4) Drajveri se uklanjaju, a klasteri se kombinuju i pomeraju naviše u programskoj strukturi.

# Integracija od dna ka vrhu (2)

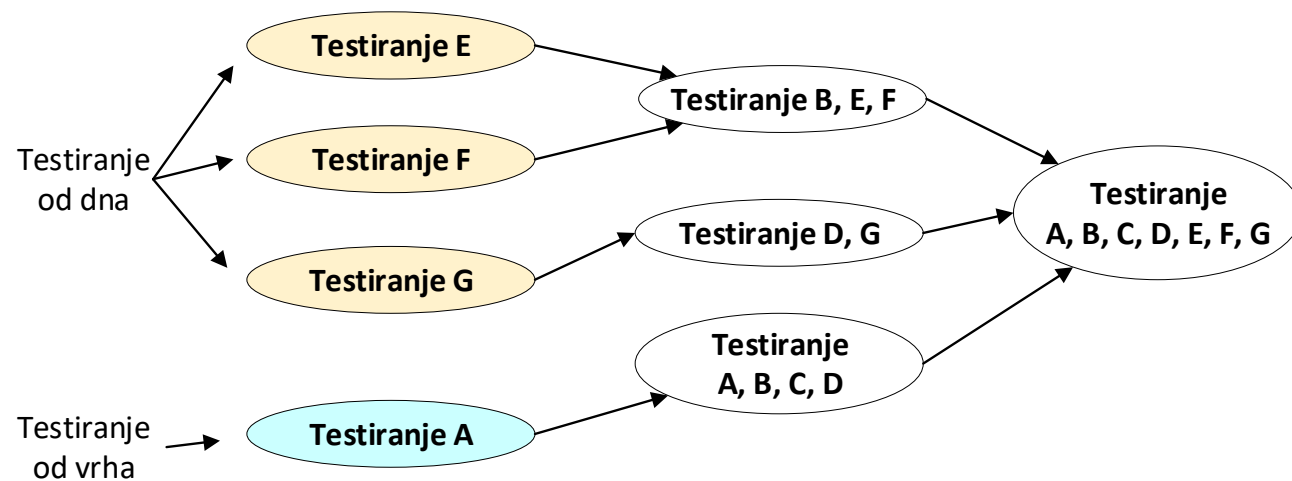
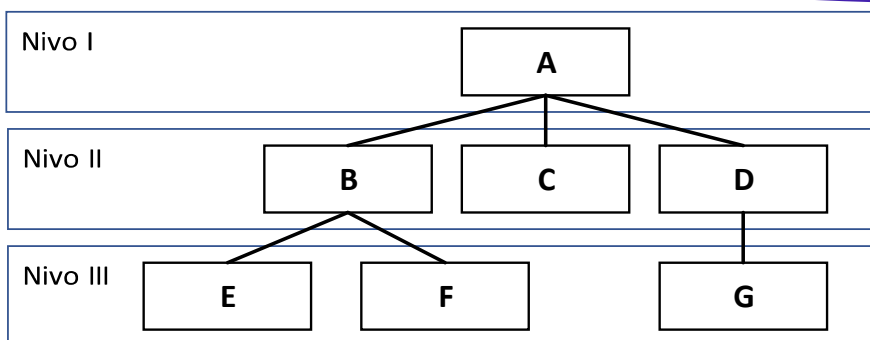


# Mešovita integracija (ili Sendvič)

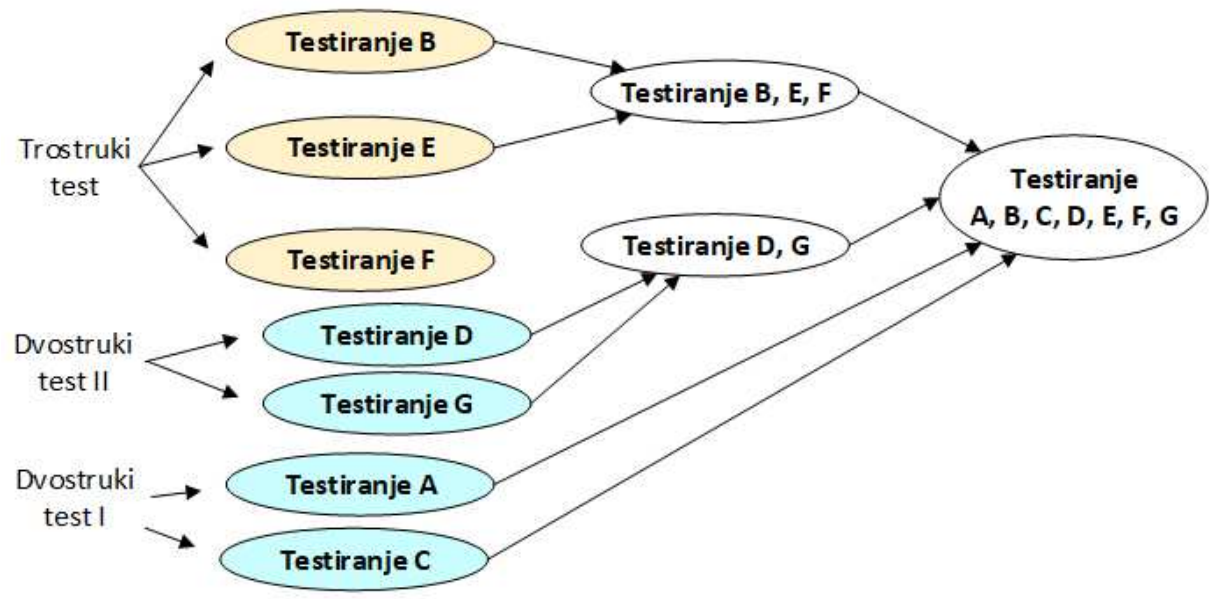
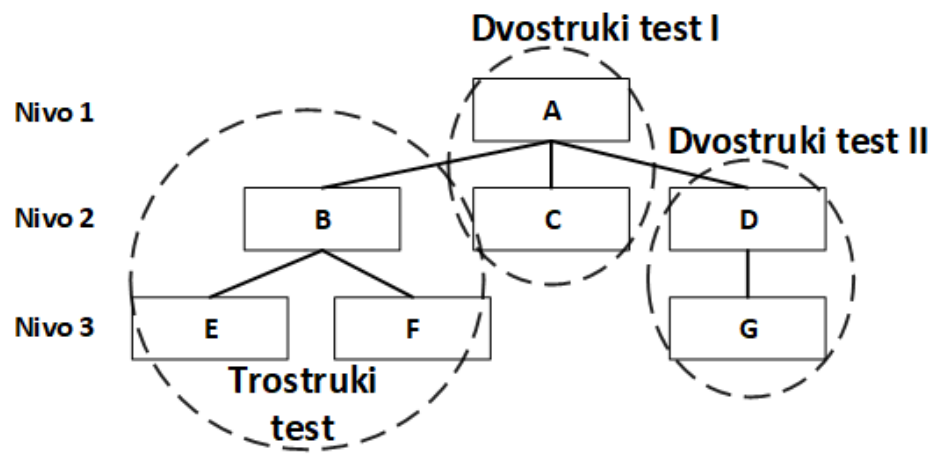
- ▶ Mešovita integracija (eng. *Sandwich testing*) je pristup u kome se kombinuje integraciono testiranje od vrha ka dnu, sa testiranjem od dna ka vrhu
- ▶ Kod ovog testiranja postoje sledeća tri sloja:
  - ▶ Ciljni sloj, koji je izabran u sredini;
  - ▶ Sloj iznad ciljnog sloja;
  - ▶ Sloj ispod ciljnog sloja.
- ▶ Testiranje konvergira ka ciljnom sloju



# Mešovita integracija (ili Sendvič) (2)



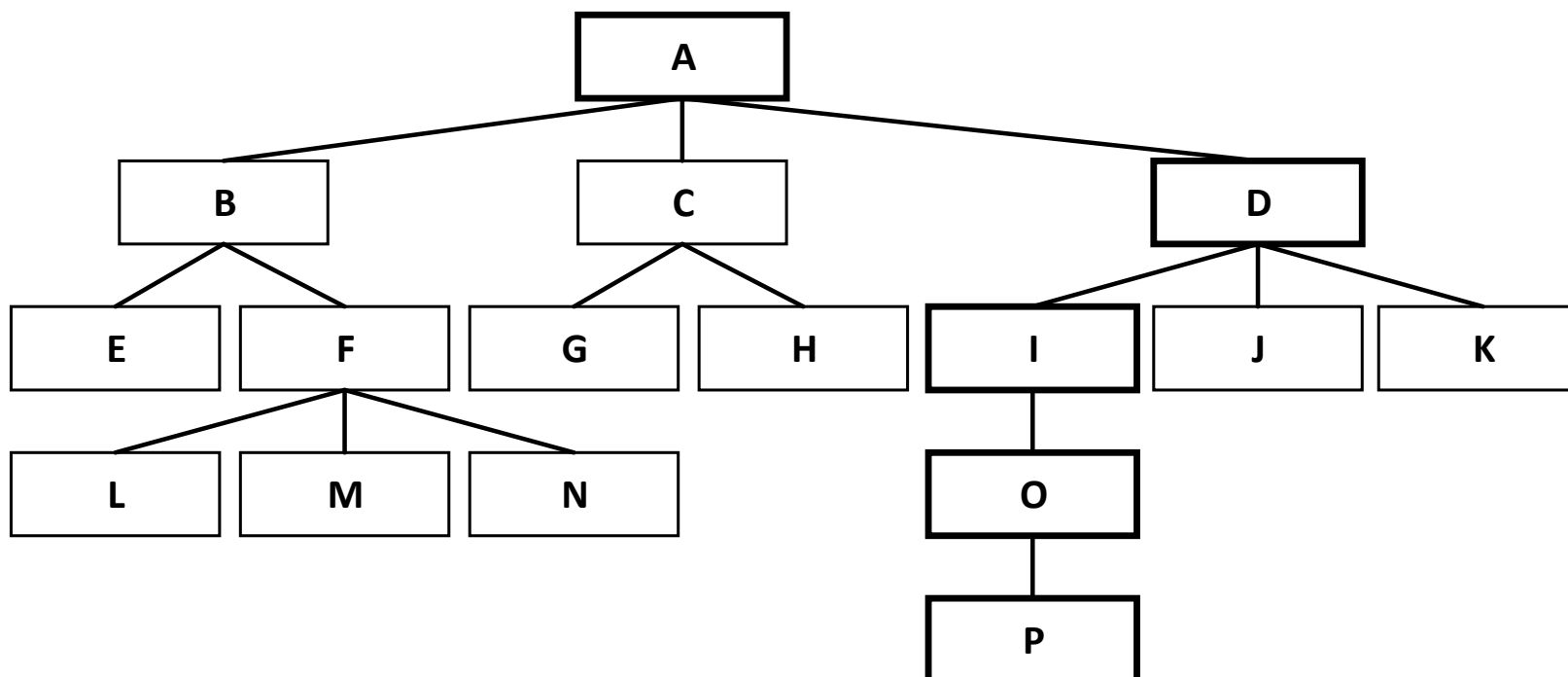
# Modifikovana mešovita integracija



## Zadatak 2 - Integracija programa (svi pristupi)

- ▶ Sistem sa slike, koji je izdeljen na module, integraciono se testira sledećim pristupima:
  - ▶ od vrha ka dnu,
  - ▶ od dna ka vrhu,
  - ▶ mešovitom itegracijom (sendvič testiranjem).
- ▶ U nizu koraka potrebno je objasniti svaki od ova tri pristupa. Podebljano su označeni kritični moduli (A, D, I, O, P). Skicirati šta tačno sadrži deo sistema koji se testira (koje module, stabove, drajvere) u svakom koraku, do potpune integracije sistema. Ukratko objasniti svaki pristup, ali ne morate objašnjavati svaki korak pojedinačno.

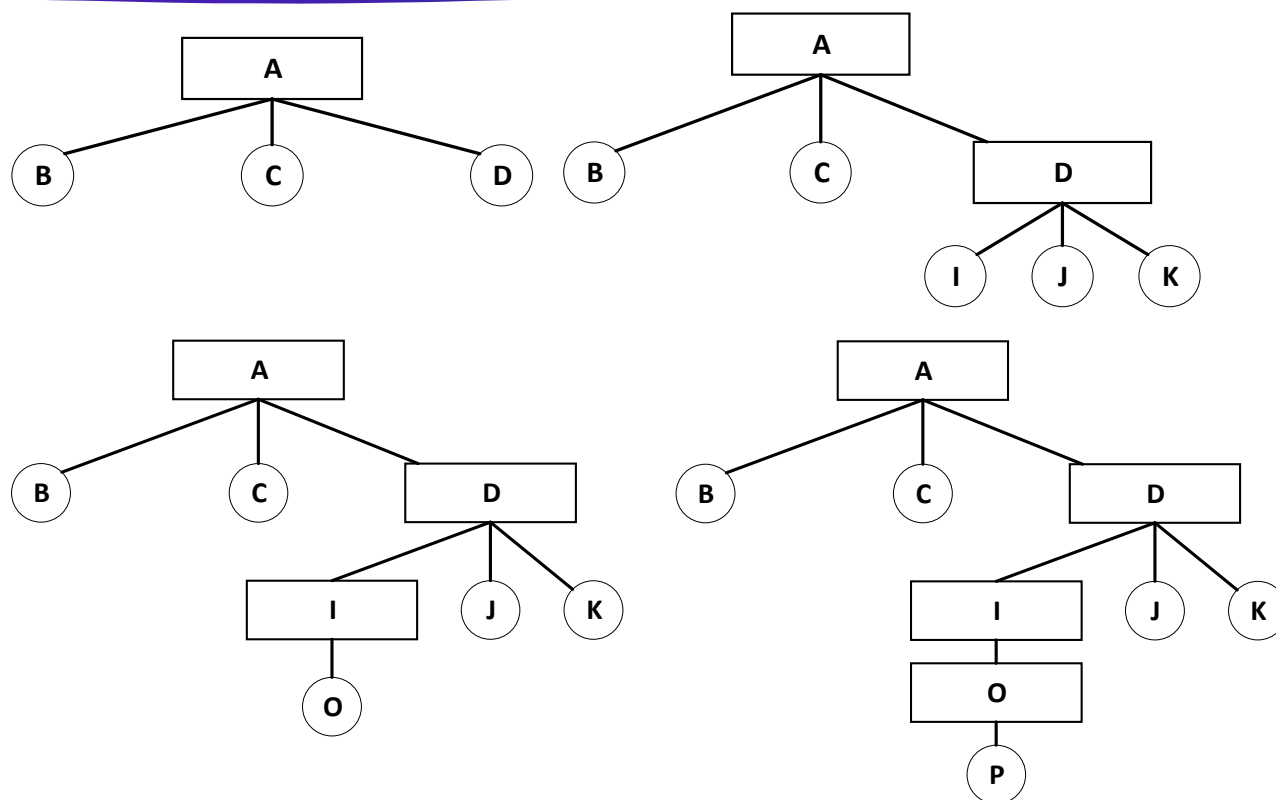
## Zadatak 2 - Integracija programa





## Zadatak 2 - Rešenje (1) - Od vrha ka dnu (a)

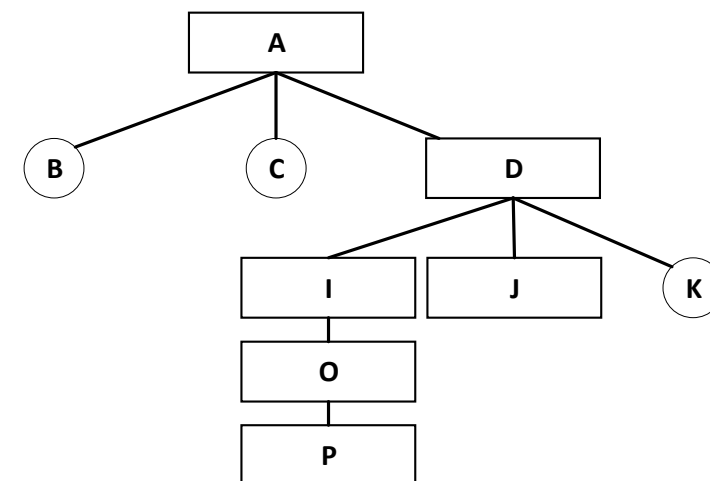
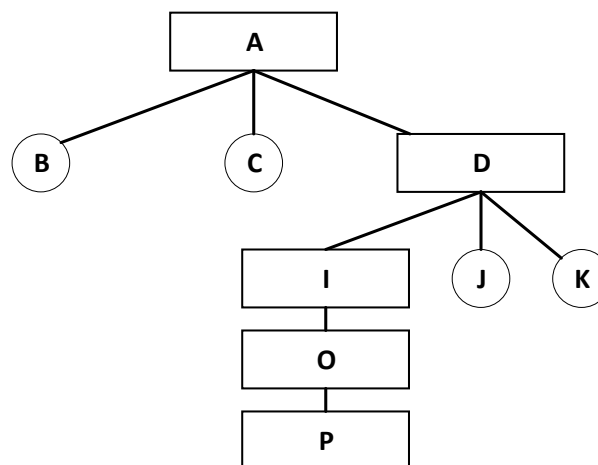
- ▶ Pristup od vrha ka dnu, po dubini
- ▶ 1. korak = koren
- ▶ 2. korak = integracija D
- ▶ 3. korak = integracija I
- ▶ 4. korak = integracija O



## Zadatak 2 - Rešenje (2) - Od vrha ka dnu (a)

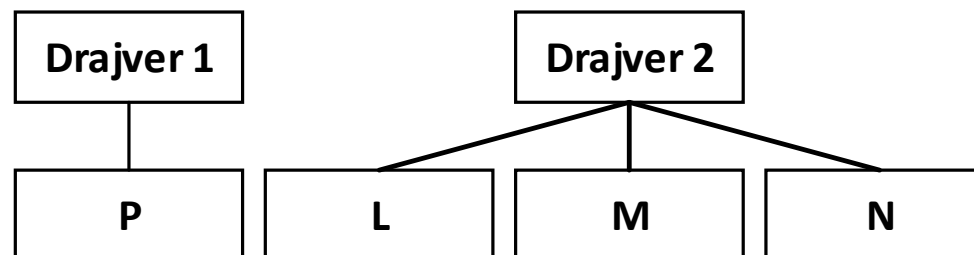
- ▶ 5. korak = integracija P
- ▶ 6. korak = integracija J
- ▶ 7. korak = integracija K
- ▶ 8. korak = integracija B
- ▶ 9. korak = integracija E
- ▶ 10. korak = integracija F
- ▶ 11. korak = integracija L
- ▶ 12. korak = integracija M

...



## Zadatak 2 - Rešenje (3) - Od dna ka vrhu (b)

- ▶ Pristup od dna ka vrhu: svaka komponenta koja je najniža u hijerarhiji, testira se pojedinačno, a zatim se testiraju komponente koje se oslanjaju na te komponente.
- ▶ Korak 1:  
testirati pojedinačno komponentu P iz najnižeg nivoa (prvi klaster) i komponente L, M i N (drugi klaster), korišćenjem drajvera:



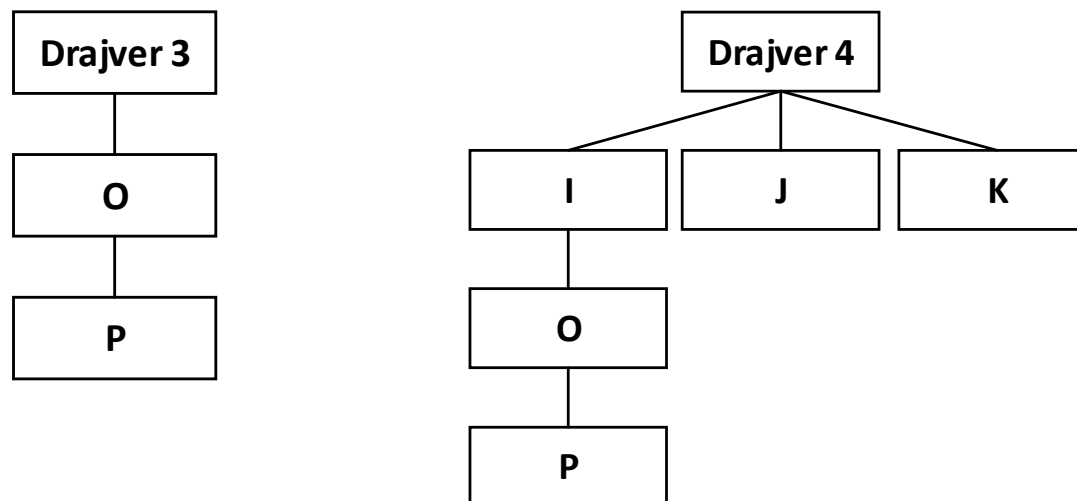
- ▶ Korak 2:  
Po integraciji P, L, M, N, radimo zamenu Drajvera 1 sa O i Drajvera 2 sa F
- ▶ Šta ukoliko primetimo grešku između O i P? Ili recimo F i M?

## Zadatak 2 - Rešenje (4) - Od dna ka vrhu (b)

► Korak 3:

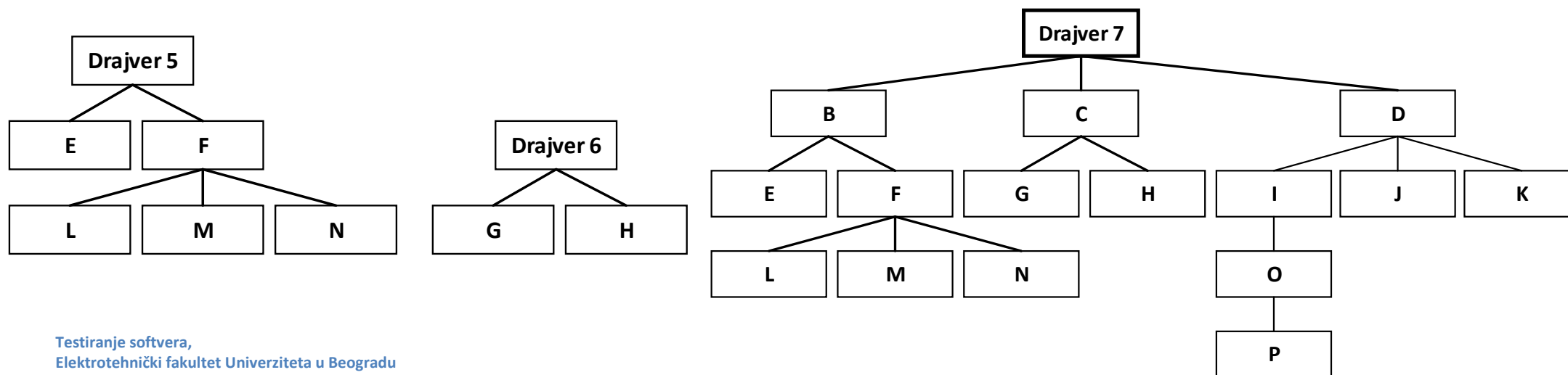
Integrišemo i komponentu I, sa O i P, nakon zamene Drajver 3 sa komponentnom I.

Pre nego što Drajver 4 zamenimo sa komponentom D, moramo da istestiramo klaster 3 (I-O-P), i pojedinačno komponente J i K, a zatim da ih integrišemo sa Drajver 4.

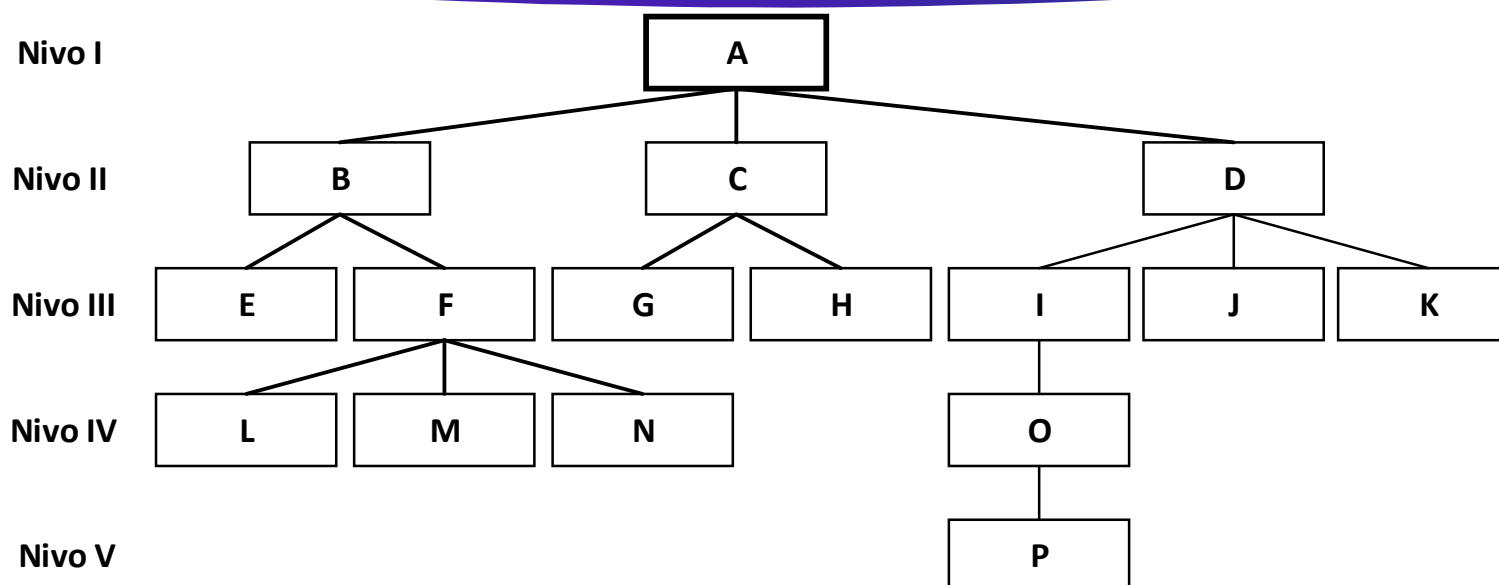


## Zadatak 2 - Rešenje (5) - Od dna ka vrhu (b)

- ▶ **Korak 4:** Paralelno sa ovim možemo testirati individualno i E, G i H, pa napraviti 2 nova klastera (grozda) sa 2 nova drajvera i njih testiramo kao posebne funkcionalnosti
- ▶ Nakon što zamenimo B, C i D umesto Drajvera 5, 6 i 4, povezaćemo sve u jednu celinu, ali tako što testirane klasterne dodajemo jedan po jedan (B klaster integrišemo sa Drajver 7, C sa Drajver 7 i D sa Drajver 7)



## Zadatak 2 - Rešenje (6) - Mešovita integracija (c)



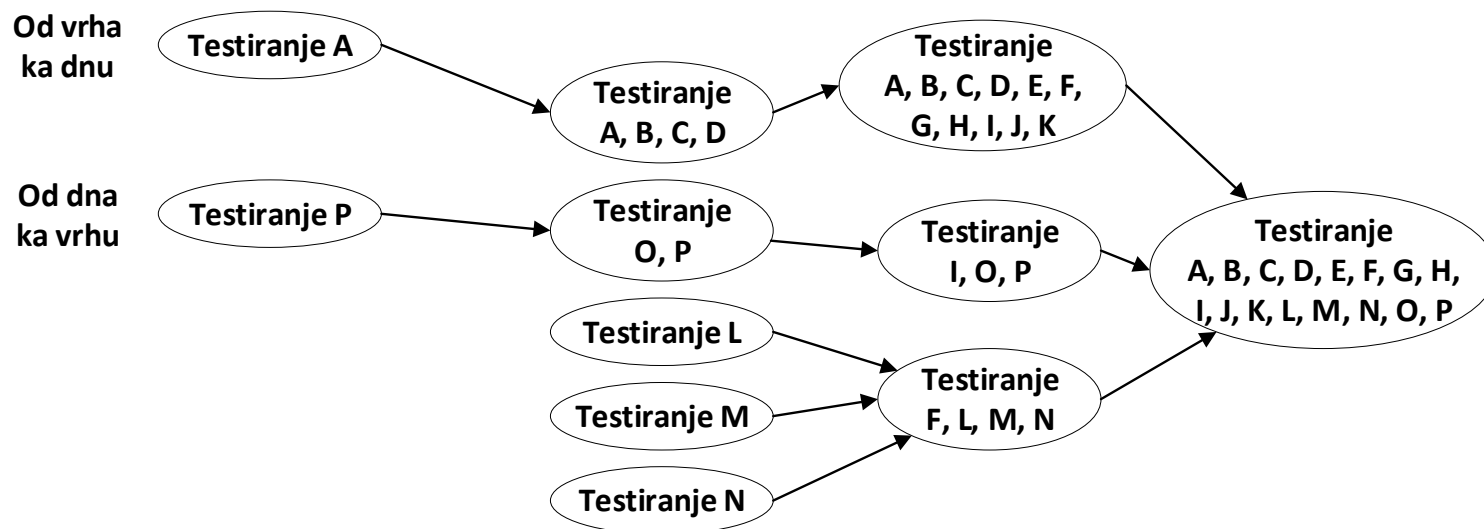
- ▶ Komponenta A ima 3 staba (B, C, D), a komponenta P ima Drajver 1. Svaki stab menjamo sa komponentom i redom integrišemo B, pa C, pa D (nećemo uzimati u obzir kritičnu putanju).
- ▶ Komp. O će zameniti Drajver 1, kada integrišemo O i P komponente unutar prvog klastera.

## Zadatak 2 - Rešenje (7) - Mešovita integracija (c)

- ▶ Drugi korak:
- ▶ Komponenta B koja je priključena ima 2 staba, komponenta C ima 2 staba i komponenta D 3 staba.
- ▶ Menjamo stabove redom komponentama E i F (vezujemo sa komponentom B), G i H (vezujemo sa komponentom C), I, J i K (vezujemo sa komponentom D), i na taj način smo uspešno pristupom od vrha ka dnu, integrisali nivo III.
- ▶ Komponente L, M i N testiramo pojedinačno i sa Drajerom 2, kao jedan klaster, zatim komponentu F menjamo umesto Drajer 2, i testiramo ceo klaster (F, L, M, N).
- ▶ Klaster O-P testiramo sa Drajerom 3, menjamo komponentu I umesto Drajer 3, i testiramo ceo klaster (I, O, P).

## Zadatak 2 - Rešenje (8) - Mešovita integracija (c)

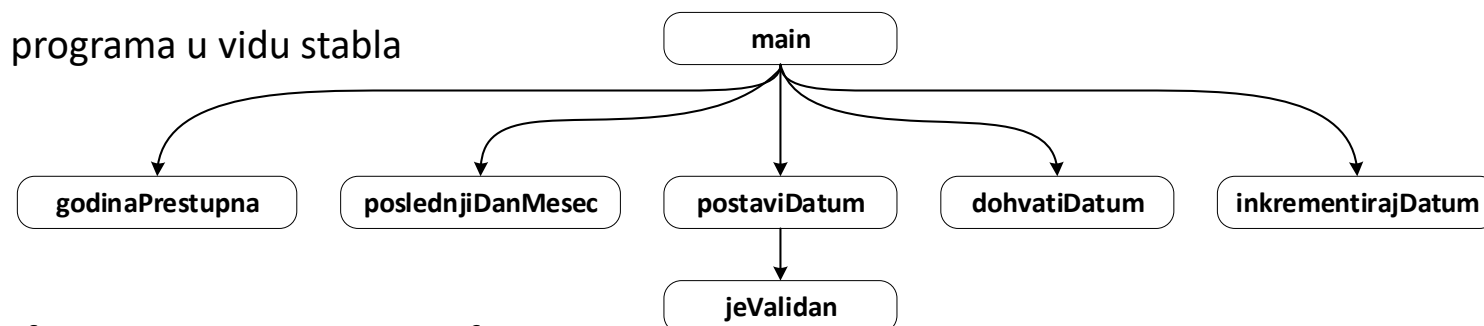
- ▶ Treći korak:
- ▶ U ovom koraku nam se susreću pristupi od vrha ka dnu i od dna ka vrhu, tako da realizujemo poslednji korak mešovite integracije.





## Zadatak 3 - Novi dan - Primeri stabla i usmerenog grafa

- ▶ Funkcionalna dekompozicija programa u vidu stabla



- ▶ Integraciono testiranje po grafu poziva => Usmereni graf poziva

