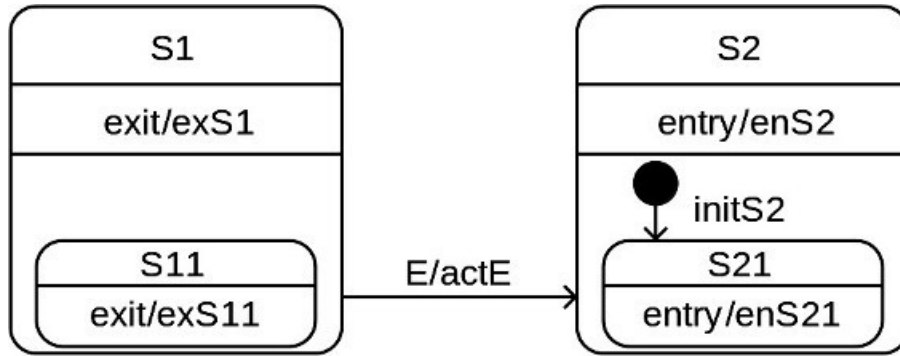


Samoprovjera znanja II. Ciklusa – OPP

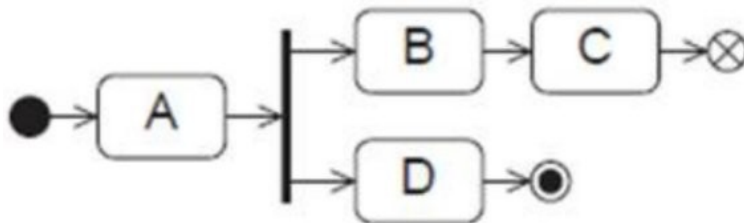
1. UML dijagrami koji se primjenjuju za opis modela toka upravljanja (engl. *control flow*) ili toka podataka su:
 - a. Dijagrami aktivnosti (engl. *activity diagram*)
 - b. Sekvencijski dijagrami (engl. *sequence diagram*)
 - c. Komunikacijski dijagrami (engl. *communication diagram*)
 - d. Dijagrami stanja (engl. *statechart diagram*)
2. Artefakti dijagrama komponenata mogu biti:
 - a. statičke i dinamičke biblioteke
 - b. izvorni kodovi
 - c. poslužitelji
 - d. sklopovske komponente
 - e. izvršne datoteke
3. Za objekt u UML dijagramu objekata (engl. *object diagram*) vrijede tvrdnje:
 - a. Simbol objekta je pravokutnik s dva pretinca.
 - b. Objekti nemaju definiciju atributa.
 - c. Objekti imaju definiciju metoda
 - d. Objekti imaju vrijednosti atributa.
 - e. Simbol objekta je pravokutnik s tri pretinca.
4. Moguće razine vidljivosti unutar UML dijagrama paketa (engl. *package diagram*) su:
 - a. javno (engl. *public*)
 - b. paket (engl. *package*)
 - c. privatno (engl. *private*)
 - d. zaštićeno (engl. *protected*)
5. Topologiju sustava te odnose njegovih sklopovskih i programskih dijelova prikazuju UML dijagrami:
 - a. objekata (engl. *object diagram*)
 - b. razmještaja (engl. *deployment diagram*)
 - c. komunikacije (engl. *communication diagram*)
 - d. paketa (engl. *package diagram*)
 - e. komponenti (engl. *component diagram*)

6. Za zadani dijagram stanja procesa odredite redoslijed akcija pri pojavi događaja E:



exS11 → actE → enS2 → initS2 → enS21

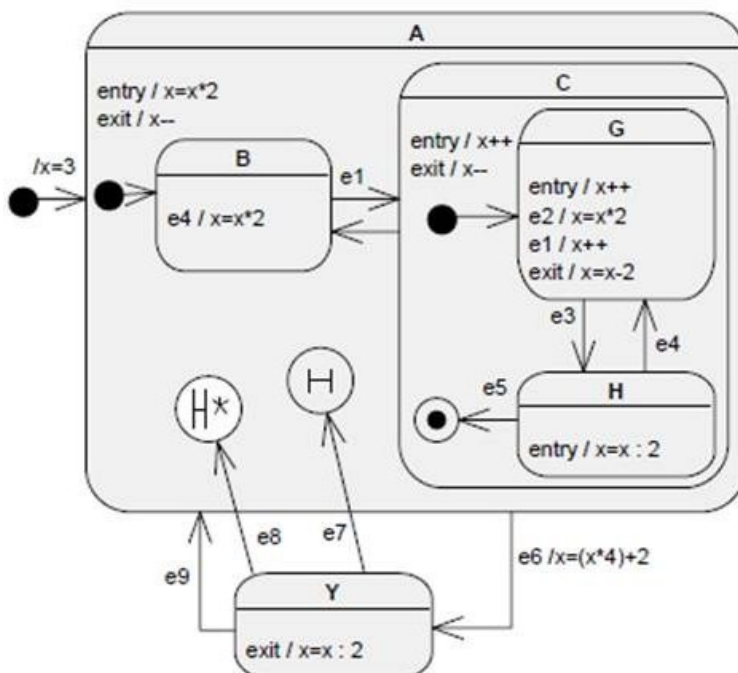
7.



Sve aktivnosti dijagrama stanja sa slike završavaju dohvaćanjem završnog stanja:

- a. D
- b. B
- c. C
- d. A

8.



Stanje u kojem se nalazi automat nakon slijeda događaja **e1 e2 e3 e6 e8** je: **H**.

9. U skupinu UML dijagrama međudjelovanja (engl. interaction diagrams) spadaju:
- a. sekvencijski dijagram (engl. sequence diagram)
 - b. dijagram aktivnosti (engl. activity diagram)
 - c. komunikacijski dijagram (engl. communication diagram)
 - d. dijagram stanja (engl. statechart diagram)
 - e. dijagram obrazaca uporabe (engl. use-case diagram)
10. Prikažite ispravan odnos uporabe biblioteka i radnih okvira pri izgradnji programske potpore:
- program -----> biblioteka
program <----- radni okvir
11. Princip oblikovanja kojemu je cilj osigurati sakrivanje ili odgodu razmatranja detalja te time i smanjenje složenosti naziva se:
- a. Zadrži razinu apstrakcije
 - b. Planiraj zastaru
 - c. Oblikuj za fleksibilnost
 - d. Povećaj ponovnu uporabivost
 - e. Oblikuj po ugovoru
12. Princip oblikovanja koji se temelji na aktivnom predviđanju budućih mogućih promjena i provedbi pripreme za njih naziva se:
- a. Zadrži razinu apstrakcije
 - b. Planiraj zastaru
 - c. Oblikuj za fleksibilnost
 - d. Oblikuj konzervativno
 - e. Oblikuj za ispitivanje
13. Ako za neki sustav vrijedi da promjene na jednom mjestu zahtijevaju i promjene drugdje može se reći da ima veliku:
- a. prenosivost
 - b. koheziju
 - c. fleksibilnost
 - d. međuovisnost
 - e. zastaru
14. Prednosti uporabe arhitektura s n razina su:
- a. Pojednostavljeno oblikovanje temeljem viših razina apstrakcije.
 - b. Ograničeni utjecaj promjena jedne razine na cijeli sustav.
 - c. Postizanje poboljšanja performansi izričito korištenjem definirane n -razinske arhitekture pri oblikovanju.
 - d. Lako je odrediti optimalno preslikavanje odgovornosti na razine.
 - e. Olakšana potpora zahtjevima povećanja i poboljšanja sustava.
 - f. Dobra potpora ponovnom korištenju (engl. reuse), prenosivost i sl.

15. Hoće li doći do greške u sustavu ako administrator poslužitelja tijekom rada kada postoje spojeni klijenti promijeni mrežne parametre poslužitelja (npr. port)?

a. NE

b. DA

c. U OCFS radnom okviru port parametar je konstanta i ne može se mijenjati.

16. Od navedenih koji razred pripada klijentskoj strani?

a. AbstractServer

b. AbstractClient

c. ConnestionToCLient

17. Koje metode korisnik OCSF radnog okvira može (ali ne mora) implementirati za poslužiteljsku stranu?



a. sendToAllClients()

b. getNumberOfClients()

c. clientException()

d. handleMessageFromClient()

e. listen()

18. Dretva u Javi uobičajeno završava svoj rad:

a. završetkom metode `run()`

b. pozivom metode `stop()`

c. Dretva u Javi nema definiran završetak rada.

d. pozivom metode `sleep()`

19. Na kojoj se web tehnologiji baziraju standardni komunikacijski protokoli u SOA-i?

XML

20. Ako program u svom radu ne zadovoljava korisničke zahtjeve tu manifestaciju nazivamo:
- a. zatajenje
 - b. sve navedeno
 - c. pogreška
 - d. nedostatak
 - e. kvar
21. Programskom potporom nazivamo:
- a. program, pripadajuću dokumentaciju, korisničke upute
 - b. program i sklopovsku dokumentaciju
 - c. skup povezanih programa
 - d. skup naredbi i procedura
 - e. razvojnu okolinu
22. Za funkcijsko ispitivanje odredite ispravne tvrdnje:
- a. Pretpostavlja da nema znanja programskog koda
 - b. Ne može upotrijebiti ekvivalentne podjele ulaza
 - c. Oblikovanje ispitnih slučajeva je zasnovano na specifikaciji sustava
 - d. Pretpostavlja da nema znanja o oblikovanju sustava
 - e. Pogodno za slučaj kada ne možemo predvidjeti izlaz za sve kombinacije ulaza
23. Uporabom se programska potpora kvari jer:
- a. je izložena teškim radnim uvjetima
 - b. krajnji korisnici se mijenjaju
 - c. pogreške se uočavaju tek kod intenzivne i dugotrajne uporabe
 - d. rezervni dijelovi zastarijevaju
 - e. sklopovlje na kojem se izvodi podložno je starenju
 - f. postaje nekompatibilna s okolinom
24. Za teorem dedukcije vrijedi:
- a. S je logička posljedica G ako je $(G \models \neg \models S)$ nezadovoljiva
 - b. S je logička posljedica G ako je $(G \models \models \neg \models S)$ zadovoljiva
 - c. S je logička posljedica G ako je $(G \models \neg \models S)$ tautologija
 - d. S je logička posljedica G ako je $(G \models \neg \models S)$ nezadovoljiva
25. Odredite ispravnu tvrdnju ako za sustav vrijedi da pravilima {L} možemo dokazati svaku logičku posljedicu skupa {G}.
- a. Sustav je kompletan.
 - b. Sustav je poluodrediv.
 - c. Sustav je zadovoljiv.
 - d. Sustav je odrediv.
 - e. Sustav je ispravan.
26. Definiraj predikate i preslikaj rečenicu prirodnog jezika u ispravnu formulu predikatne logike:
- "Svako dijete je mlađe od svoje majke."

a.

$\forall x \forall y ((\text{dijete}(x) \wedge \text{majka}(y, x)) \Rightarrow \text{mlađi}(x, y))$

b. $\text{mlađi}(x, y) = x$ je mlađi od y

c. $\text{majka}(x, y) = x$ je majka od y

d. $\text{dijete}(x) = x$ je dijete

e. $\text{mlađi}(x) = x$ je mlađi

27. Za model sustava M sa stanjima (s_1, s_2, s_3, \dots) u slučaju da za sve putove

$s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ gdje za svako stanje s_i duž puta, vrijedi $M, s_i \models \varphi$ vrijedi tvrdnja:

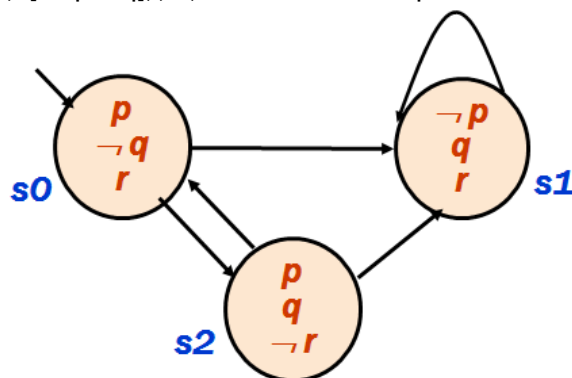
a. $M, s_1 \models \text{AF } \varphi$

b. $M, s_1 \models \text{EG } \varphi$

c. $M, s_1 \models \text{AG } \varphi$

d. $M \models \text{EG } \varphi$

28. Odredite istinitost tvrdnje $(A[\neg p \vee q])(s_1)$ za model sustava prema slici.



a. NE

b. Formula je neispravna

c. DA

29. Opišite CTL formulom: Bez obzira u kojem stanju sustav se nalazi, novo postavljene **zahtjev** će konačno biti **prihvaćen** (kad-tad).

a. $\text{AG} (\text{zahtjev} \rightarrow \text{AF prihvaćen})$

b. $\text{AF} (\text{zahtjev} \rightarrow \text{AF prihvaćen})$

c. $\text{AG} (\text{zahtjev} \rightarrow \text{prihvaćen})$

d. niti jedan od navedenih

e. $\text{A} (\text{zahtjev} \vee \text{prihvaćen})$

30. Odredite ispravno tumačenje CTL formule: $\text{CTL} \models \neg \text{AX AG } \varphi$

a. sada ili za svako slijedeće stanje vrijedi $\text{AF } \varphi$

b. sada i na svim putovima počevši od slijedećeg

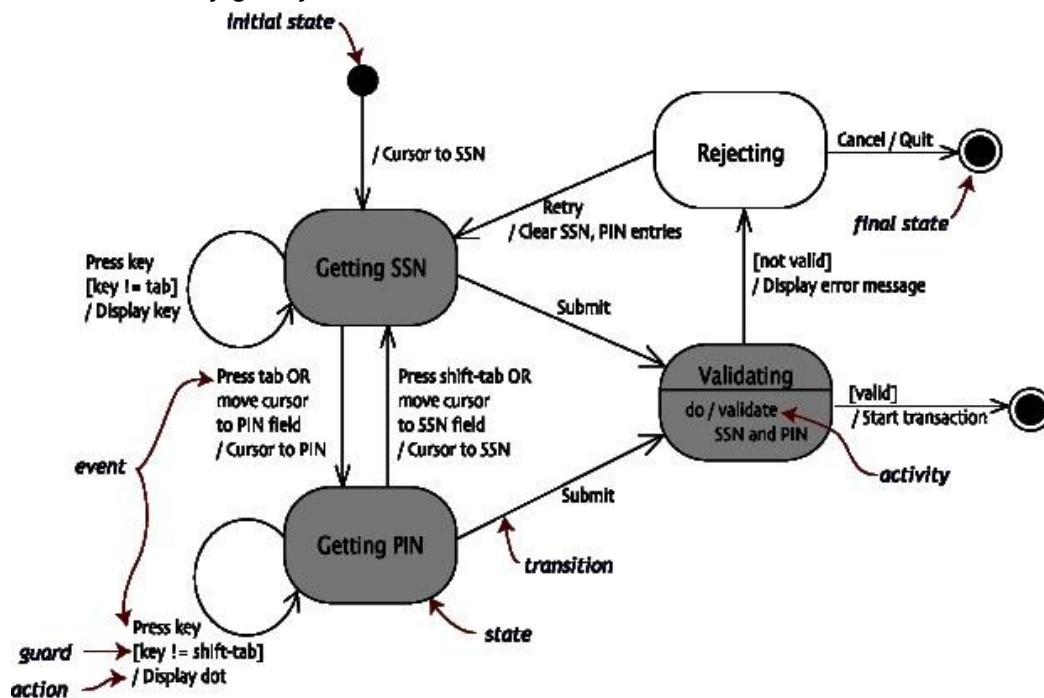
c. sada i na jednom putu počevši od slijedećeg

d. sada ili za svako slijedeće stanje vrijedi $\text{EF } \varphi$

e. niti jedan od navedenih

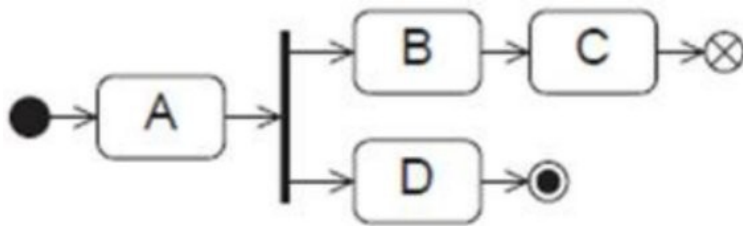
31. UML dijagrami koji se temelje na prikazu fizičkih modularnih i zamjenjivih jedinica kojima se može pristupiti samo kroz sučelja su:
- Dijagrami paketa (engl. package diagram)
 - Dijagrami objekata (engl. object diagram)
 - Dijagrami razreda (engl. class diagram)
 - Dijagrami komponenti (engl. component diagram)**
32. Za karakteristike temeljnih sastavnica UML dijagrama razmjesta (engl. deployment diagram) vrijede tvrdnje:
- Ovisnosti između komponenti se prikazuju punom crtom
 - Čvorovi su programski artefakti, a komponente sklopovski artefakti sustava
 - Čvorovi su sklopovski artefakti, a komponente programski artefakti sustava**
 - Ovisnosti između komponenti se prikazuju crtkanom strelicom**
 - Proširenje stereotipovima je dozvoljeno samo na vezama.

33. Prikazani UML dijagram je:



- sekvencijski dijagram (engl. sequence diagram)
- dijagram aktivnosti (engl. activity diagram)
- dijagram objekata (engl. object diagram)
- dijagram obrazaca uporabe (engl. use-case diagram)
- dijagram razreda (engl. class diagram)
- dijagram stanja (engl. statechart diagram)**

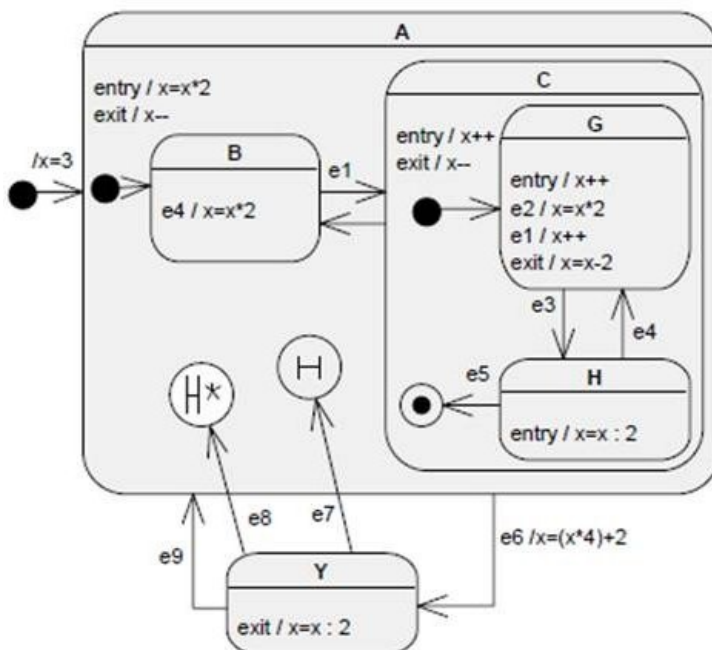
34.



Moguće sekvence izvođenja dijagrama stanja sa slike su:

1. $A \rightarrow B \rightarrow C \rightarrow D$
2. $A \rightarrow B \rightarrow D$
3. $A \rightarrow D$

35.



Stanje u kojem se nalazi automat nakon slijeda događaja **e1 e2 e3 e6 e7** je: **G**.

36. Princip oblikovanja povećanja ponovne uporabivosti u pristupu oblikovanju je komplementaran principu:

- a. Planiraj zastaru
- b. Povećaj uporabu postojećeg**
- c. Oblikuj za prenosivost
- d. Oblikuj po ugovoru
- e. Oblikuj za ispitivanje

37. Svi razredi u radnom okviru OCSF su apstraktni.

Točno

Netočno

38. Koje metode korisnik OCSF radnog okvira može (ali ne mora) implementirati za klijentsku stranu?



- a. `sendToServer()`
- b. `handleMessageFromServer()`
- c. `closeConnection()`
- d. `openConnection()`
- e. `connectionException()`

39. Od navedenih koji razred(i) pripada(ju) poslužiteljskoj strani?

- a. `AbstractServer`
- b. `AbstractClient`
- c. `ConnectionToClient`

40. Koje tvrdnje vrijede za ispitivanje programske potpore?

- a. Ispitivanje je proces pokazivanja odsustva pogrešaka.
- b. Cilj ispitivanja je pokazati da program ispravno obavlja željene funkcije.
- c. Ispitivanje je proces izvođenja programa sa svrhom pronalaženja pogrešaka.
- d. Ispitivanje je proces uspostave povjerenja ispravnog rada.

41. Zadaća ispitivanja pod pritiskom (engl. *stress testing*) je:

- a. Ispitivanje memorijskih zahtjeva
- b. Ispitivanje performansi
- c. Određivanje stabilnosti sustava

42. Za metriku pokrivanja stupnja gotovosti ispitivanja upotrebljavamo:

- a. Vrijeme izvođenja.
- b. Pokrivenost putova.
- c. Pokrivenost grana.
- d. Broj ispitnih slučajeva.
- e. Postotak programskih elemenata koji su izvedeni.

f. Pokrivenost linija koda.

43. Odredite ispravnu tvrdnju. Ako u formalnom sustavu $\{\mathcal{T}, L\}$ izvodimo neki teorem w_i (tj. tražimo odgovor da li je w_i teorem ili ne), a ne postoji algoritam koji će uvijek u konačnom vremenu pokazati da w_i nije teorem ako on zaista nije teorem.

a. Sustav je poluodrediv.

b. Sustav je kompletan.

c. Sustav je odrediv.

d. Sustav je ispravan

e. Sustav je zadovoljiv.

44. Uz pretpostavku definiranog značenja predikata: (x, y) , čija je semantika x jednako y , definirajte minimalan broj ostalih potrebnih predikata i preslikajte rečenicu u formalizam predikatne logike: "Ante ima barem dvije sestre."

a. $\text{sestra}(x, y) - x$ je sestra od y

b. $\forall x \exists y [\text{sestra}(x, \text{Ante}) \wedge \text{sestra}(y, \text{Ante}) \Rightarrow \neg(x, y)]$

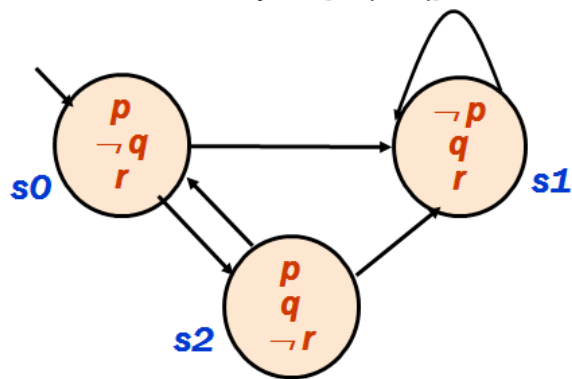
c. $\exists x \exists y [\text{sestra}(x, \text{Ante}) \wedge \text{sestra}(y, \text{Ante}) \wedge \neg(x, y)]$

d. $\forall x [(\exists y M(y, x) \Rightarrow \exists z O(z, x))]$

e. $\exists y \exists x [\text{sestra}(x, \text{Ante}) \wedge \neg(x, y) \wedge \text{sestra}(y, \text{Ante})]$

f. Ante – konstanta

45. Odedite istinitost tvrdnje $(EF[\neg p \vee q])(s_0)$ za model sustava prema slici.



a. DA

b. Formula je neispravna

c. NE

46. Opišite CTL formulom: Bankomat će uvijek zaključati karticu nakon tri uzastopna neuspješna unosa PIN-a.

a. $A (\text{unos} \vee \text{zaključana_kartica})$

b. niti jedan od navedenih

c. $AF (\text{unos} \Rightarrow AF \text{ zaključana_kartica})$

d. $AG (\text{unos} \Rightarrow AX AX AX \text{ zaključana_kartica})$

e. $AG (\text{unos} \Rightarrow \text{zaključana_kartica})$