1. Pocetno stanje
2. Pokazuje gdje tok pocinje. Svaki dijagram aktivnosti moze imati jedno i vise pocetnih stanja  ○
3. Pokazuje gdje tok počinje. Svaki dijagram aktivnosti mora imati vise  početnih stanja ○
4. Pokazuje gdje tok počinje. Svaki dijagram aktivnosti mora imati  
   jedno i samo jedno početno stanje ●

1. Alati koji su nam na raspolaganju za kreiranje dijagrama aktivnosti su
2. Transakcija ☐
3. Odlucivanje ☑
4. Zavrsna tacka ☑
5. Pocetna tacka ☑
6. Aktivnost ☑
7. Prelaz ☑
8. Organizacijska jedinica ☑
9. Sinkronizacija ☑
10. Okidac ☐
11. Stanje objekta ☑
12. Da li poruke destroy i self destroy mogu biti rekurzivne?
13. Da ○
14. Ne ●
15. Slozeno stanje koristi podstanja za detaljniji opis.
16. Da ○
17. Ne ●
18. U kojoj fazi razvoja softvera je dijagram komponenti koristan?
19. Analize i dizajna ●
20. Analize ○
21. Dizajna ○
22. Ponudjeno sucelje je
23. Komponente sadrže jednu ili više klasa koje implementiraju usluge   
    koje komponenta pruža putem sučelja ●
24. Komponenta također može putem sučelja zahtijevati uslugu od   
    drugih komponenti ○
25. Koji pogled predstavlja sekvencijalni dijagram?
26. Logicki ●
27. Fizicki ○
28. Dijagram saradnje mozete pretvoriti u sekvencijalni dijagram i obrnuto
29. Tacno ●
30. Netacno ○
31. Zahtjevano sucelje je
32. Komponenta također može putem sučelja zahtijevati uslugu od   
    drugih komponenti ●
33. Komponente sadrže jednu ili više klasa koje implementiraju   
    usluge koje komponenta pruža putem sučelja ○
34. Komponente mogu biti neovisne ili međusobno povezane vezama ovisnosti i vezama generalizacije ukoliko jedna komponenta nasljeđuje osobine druge komponente
35. Netacno ○
36. Tacno ●
37. Komponente mogu koristiti jedan ili vise interface-a.
38. Tacno ●
39. Netacno ○
40. Dijagrami stanja su korisni za modeliranje
41. Jednostavnih poslovnih entiteta ☐
42. Objekata koji se ponasaju razlicito ovisno o stanju u kojem se   
    nalaze ☑
43. Jednostavnih objekata ☐
44. Realizaciju slucajeva upotrebe ☑
45. Modeliranje elemenata korisnickog sucelja ☑
46. Ponasanje podsistema ili modula aplikacije ☑
47. – kompleksnih podsistema ili modula aplikacije ☑
48. Kompleksnih objekata ☑
49. Interakciju podsistema ili modula aplikacije ☐
50. Oznaci ispravne tvrdnje
51. Prelaz od starta do junction pointa moze postojati ☑
52. Prelaz od stanja do junction pointa moze postojati ☑
53. Prelaz od starta do stanje moze postojati ☑
54. Prelaz od enda do stanja moze postojati ☐
55. Prelaz od starta do enda moze postojati ☐
56. Prelaz od stanja do starta moze postojati ☐
57. Komponente imaju vecu odgovornost od klasa
58. Netacno ○
59. Tacno ●
60. Za slozeno stanje mogu se izraditi dijagrami stanja koji detaljnije opisuju   
    njegovo ponasanje
61. Netacno ○
62. Tacno ●
63. U dijagramu aktivnosti krajnje stanje
64. Pokazuje gdje tok zavrsava, mora postojati vise od jednog krajnjeg   
    stanja ○
65. Pokazuje gdje tok završava, mora postojati samo jedno krajnje   
    stanje ○
66. Pokazuje gdje tok završava, može postojati više od jednog krajnjeg   
    stanja ●
67. Pokazuje gdje tok pocinje , može postojati više od jednog krajnjeg   
    stanja ○
68. Navedi implementacijske dijagrame
69. Kolaboracijski dijagram ☐
70. Dijagram aktivnosti ☐
71. Dijagram komponenti ☑
72. Sekvencijalni dijagram ☐
73. Dijagram stanja ☐
74. Dijagram rasporeda ☑
75. Odaberi tacne aktivnosti mozemo definirati za stanje u dijagramu stanja obizrom na trenutak izvrsavanja aktivnosti?
76. Finish ☐
77. Done ☐
78. Complete ☐
79. Leave ☐
80. Exit ☑
81. Did ☐
82. Entry ☑
83. Do ☑
84. Odaberi minimalnu brojnost ulaznih i izlaznih tranzicija iz/u racvanje u UML dijagramima stanja?
85. Minimalna brojnost za ulaz je jedan, a za izlaz 1 ili vise ●
86. Minimalna brojnost za ulaz je jedan, a za izlaz striktno jedna ○
87. Minimalna brojnost za ulaz je jedna ili vise, a za izlaz striktno jedna ○
88. U kojoj fazi razvoja softvera je dijagram aktivnosti koristan
89. Samo u fazi dizajna ○
90. U fazi analize i dizajna ○
91. Samo u fazi analize ●
92. Oznaci tacne tvrdnje za cvoriste/node u dijagramu rasporeda
93. Posjeduje memoriju i sposobnos procesuiranja. ☑
94. Može sadržavati komponente ☑
95. Može sadržavati druga čvorišta ☐
96. Fizički je element koji predstvalja server, računar, štampač ili   
    drugi hardverski uređaj ☑
97. Čvorište može biti hardverski ili softverski resurs ☑
98. Da li se jedna poruka moze nalaziti na dvije instance veze
99. Da ○
100. Ne ●
101. Koji je najmanji dio interakcije u sekv.
102. Strelica ○
103. Dogadjaj ●
104. Entiteti ○
105. Federacija ○
106. Oznaci ispravnu tvrdnju za junction point (definiranje cvorista)
107. Ima jedan i samo jedan ulazni prelaz, te jedan i samo jedan   
     izlazni prelaz ○
108. Ima jedan i samo jedan ulazni prelaz i vise izlaznih prelaza ○
109. Ima jedan i samo jedan izlazni prelaz i vise ulaznih prelaza ○
110. Ima nekoliko ulaznih i izlaznih prelaza ●
111. Na horizontalnoj liniji sekv. dijagrama
112. Rasporedjeni su objekti koji medjusobno komuniciraju ●
113. Predstavljeno je vrijeme ○
114. Poruka je komunikacija koja se dogada kada jedan objekt poziva metodu drugog u zahtjevu za informacijom
115. Da ●
116. Ne ○
117. Dijagrami stanja se koriste kada je više objekata uključeno u interakciju
118. Da ○
119. Ne ●
120. Stanja mogu biti
121. Prosta ☑
122. Teska ☐
123. Slozena ☑
124. Jednostavna ☐
125. <<subsystem>> je najveci oblik komponente
126. Tacno ●
127. Netacno ○
128. Promjene na jednom cvoristu direktno uticu na drugo cvoriste
129. Tacno ●
130. Netacno ○
131. Interakcija u sekv. dijagramu se koristi za
132. Opis dinamickih aspekata sistema ●
133. Opis statickih aspekata sistema ○
134. Sihrona poruka u sekvencijalnom dijagramu
135. Ne ocekuje odgovor ○
136. Ocekuje odgovor ●
137. Oznaci tacnu tvrdnju namjene dijagrama rasporeda
138. Uspostavlja vezu između hardvera i softvera ☑
139. Grafički predstavlja raspored komponenti na serverima i   
     identificira arhitekturu sistema ☑
140. Daje preciznije podatke o fizičkoj implementaciji i interakciji   
     komponenti ☑
141. Dopunjuje dijagram komponenti ☑
142. Nedjeljivo stanje ne sadrzi poredjenja stanja
143. Tacno ●
144. Netacno ○
145. Najmanji dio interakcije u sistem sekvencijalnom dijagramu je
146. Aktivnost ○
147. Dogadjaj ●
148. Akcija ○
149. Akcije su manji koraci koji se desavaju unutar aktivnosti
150. Tacno ●
151. Netacno ○
152. Ucesnici se na kolaboracijskom dijagramu predstavljaju putem
153. Paralelopipeda ○
154. Kocke ○
155. Tetraedera ○
156. Pravougaonika ●
157. Osnovni elementi dijagrama saradnje / kolaboracije
158. Instanca veze ☑
159. Poruka ☑
160. Zmaj ☐
161. Ucesnik (actor) ☑
162. Objekat ☑
163. Entitet ☐
164. Akcija ☐
165. Sistem ☐
166. Sta opisuj dijagram aktivnosti?
167. Sta sistem radi ○
168. Kako sistem radi ●
169. Konektori se koriste
170. za prikaz kako se interfejsi komponenti slažu sa internim   
     dijelovima i kako interni dijelovi sarađuju ●
171. za prikaz kako se interfejsi komponenti slažu sa vanjskim   
     dijelovima i kako interni dijelovi izgledaju ○
172. Aktivnost je zadatak koji se izvrsava u toku rada sistema
173. Tacno ●
174. Netacno ○
175. Dijagram stanja opisuje
176. Ponašanje objekata i pripada skupu dijagrama UML-a koji   
     opisuju ponašanje ○
177. Ponašanje klasa i pripada skupu dijagrama UML-a koji   
     opisuju ponašanje ●
178. ponašanje klasa i pripada skupu dijagrama UML-a koji   
     opisuju interakciju ○
179. Ponašanje objekata i pripada skupu dijagrama UML-a koji   
     opisuju ponašanje ○
180. Odaberi tacna svojstva instance veze u dijagramu saradnje
181. Dvije instance veze mogu postojati između istih pošiljaoca i   
     primaoca ☑
182. Kada kopiramo instancu veze, njena poruka je istovremeno   
     kopirana ☑
183. Kada obrišete jedinu instacu veze jedne poruke, odgovarajuća   
     poruka je automatski izbrisana a ☑
184. Moze biti rekurzivna ☑
185. Komponente mogu biti velicine
186. Pseudosistema ☐
187. Podsistema ☑
188. Modula ☑
189. Klase ☑
190. Na vertikalnoj liniji sekvencijalnog dijagrama
191. Rasporedeni su objekti koji medusobno komuniciraju ○
192. Predstavljeno je vrijeme ●
193. Dijagram aktivnosti je jedini dijagram sa procesnim pogledom na sistem
194. Tacno ●
195. Netacno ○
196. Dijagram stanja ima strukturu grafa cija cvorista su
197. Stanja ☑
198. Prelazi ☑
199. Relacije ☐
200. Sta su paralelne poruke
201. Poruke koje se pozivaju u isto vrijeme ●
202. Poruke koje se pozivaju u isto vrijeme, ali razlicito doba ○
203. Navedi tacne tvrdnje dijagrama komponenti
204. Komponente međusobno ili sa drugim klasama mogu biti   
     povezane vezama ovisnosti i vezama asocijacije ☑
205. Statički dijagram objektno orijentiranog modela ☑
206. Predstavlja strukturu softvera u obliku grafa čija čvorišta su   
     komponente zajedno sa sučeljem komponenti ☑
207. Potpuna notacija za opise prijelaza je
208. Dogadjaj[izraz]/operacija ●
209. Izraz[dogadjaj]/operacija ○
210. Operacija[izraz]/dogadjaj ○
211. Aktivnosti su manji koraci koji se desavaju unutar akcije
212. Tacno ○
213. Netacno ●
214. Cvor grananja ima
215. Jedan ili vise ulaznij tok i dva ili više izlazna toka ○
216. Jedan ili vise ulaznij tok i jedan i samo jedan izlazi tok ○
217. Jedan ulazni tok i dva ili više izlazna toka ●
218. Koliko ce poruka biti puta pozvana \*[0..7]
219. 8 ●
220. 0 ○
221. 7 ○
222. 1 ○
223. Prelazom sa stanja na stanje moze se izvrsiti vise aktivnosti
224. Tacno ○
225. Netacno ●
226. Dijagram kolaboracije
227. Naglasava kako se interakcija desava u vremenu ☐
228. Prikazuje saradnju medu objektima u specificnim situacijama ☑
229. Naglasava odnos objekata i njihovu topologiju u situacijama ☑
230. Pogodan je da se objasni proces u programskoj logici ☑
231. Sta predstavlja uslov za paralelno obavljanje tranzicija koje izlaze iz tacke sinkronizacije u dijagramu stanja?
232. Moraju sve biti bez uslova, ili da ako ima uslova da su svi   
      ispunjeni ●
233. Jedna moze biti sa ispunjenim uslovom, a druga bez ○
234. Da li su sekvencijalni i kolaboracijski dijagrami semanticki ekvivalentni
235. Da ●
236. Ne ○
237. Tipovi konektora
238. Destruktivni konektori ☐
239. Delegacijski konektori ☑
240. Konektori relacija ☐
241. Konektori spajanja ☑
242. Koji su UML – interakcijski dijagrami
243. Dijagram stanja ☐
244. Sistem sekvencijalni dijagram ☑
245. Dijagram komponenti ☐
246. Sekvencijalni dijagram ☑
247. Kolaboracijski dijagram ☑
248. Dijagram rasporeda ☐
249. Komponente ne trebaju da budu enkapsulirane
250. Tacno ○
251. Netacno ●
252. UML omogucava dva pogleda na komponente
253. Pogled kroz bijelu kutiju ☑
254. Pogled kroz sivu kutiju ☐
255. Pogled kroz crnu kutiju ☑
256. Pogled kroz crnu kutiju pokazuje
257. Kako komponente izgledaju izvana, ukljucujuci zahtijevane i   
     ponudjene intefejse komponenti ●
258. Kako komponenta izgleda s unutrasnje tacke, uključujući   
     zahtijevane i ponuđene interfejse komponenti ○
259. Prelaz spaja dva stanja i pokazuje tok kontrole
260. Tacno ○
261. Netacno ●
262. Asinhrona poruka u sekvencijalnom dijagramu
263. Ocekuje odgovor ○
264. Ne ocekuje odgovor ●
265. Da li moze postojati vise instanci veza izmedju dva objekta
266. Da ●
267. Ne ○
268. Konvencija imenovanja ucesnika u sekv. dijagramu
269. Naziv objekta navodi se velikim pocetnim slovom, a naziv klase   
     malim slovom ○
270. Naziv objekta navodi se malim slovom, a naziv klase velikim   
     pocetnim slovom ●
271. Naziv objekta , kao i naziv klase navodi se velikim slovom ○
272. Stanje je ograničenje ili situacija u životnom ciklusu objekta gdje objekt, pod nekim uvjetom (guard) izvršava aktivnosti ili čeka na događaj.
273. Tacno ●
274. Netacno ○
275. Tip fizičke komponente određuje se na osnovu njenih osobina i može biti:
276. Servis ☐
277. Tabela u bazi podataka ☑
278. Izvrsni kod ☑
279. Zaglavlje ☐
280. Kompjuter ☐
281. Biblioteka ☑
282. Web stranica ili web forma ☑
283. Dokument ☑
284. Datoteka ☑
285. Oznaci elemente dijagrama stanja
286. Prelaz iz jednog stanja u drugo stanje (transition) ☑
287. Do ☐
288. Entery ☐
289. Pocetno stanje (start) ☑
290. Zavrsno stanje (end) ☑
291. Exit ☐
292. Cvoriste u dijagramu (junction point) ☑
293. Stanje (state) ☑
294. U okviru softverskog proizvoda dijagram stanja predstavlja stanja kroz koja objekt prolazi u toku svog životnog ciklusa
295. Tacno ●
296. Netacno ○
297. Ako se zeli izbjeci prikazivanje sucelja komponente, onda se uspostavlja   
     izmedju komponenti
298. Veza ljubavi ○
299. Veza uslovljenosti ○
300. Veza asocijacije ○
301. Veza ovisnosti ●
302. Koje su osnovne karakteristike dijagrama stanja
303. Izolacija ☐
304. Trajnost ☑
305. Atomicnost ☐
306. Stabilnost ☑
307. Implementacijski dijagrami prikazuju strukturu programskog koda i strukturu implementacije u realnom vremenu
308. Tacno ●
309. Netacno ○
310. Da li moze postojati vise poruka na jednoj instanci veze
311. Da ●
312. Ne ○
313. Dijagrami rasporeda se koriste za
314. Dokumentiranje rasporeda softvera na hardverske cvorove ☑
315. Modeliranje fizickih elemenata i komunikacijskih puteva  
     izmedju njih ☑
316. Izradu plana arhitekture sistema ☑
317. Predstavljanje strukture softvera u obliku grafa ☐
318. Port komponente prikazuje dio interne strukture komponente
319. Tacno ●
320. Netacno ○
321. Interface kompontente ukazuje na to da li
322. Komponenta sadrzi implementaciju usluge koju pruza putem   
     odredjenog interface-a ●
323. Komponenta je dio objekta koji ce se moci prikazati pomocu viseg   
     interface-a ○
324. Pomocu cega se UML rasporedjeni softver prikazuje
325. Artefakta ●
326. Objekata ○
327. Klasa ○
328. Odaberi tacne tvrdnje dijagrama stanja
329. Stanje izaziva veliku zabrinutost u vezi OHR-a ☐
330. Dijagram stanja pokazuje kako i pod kojim uvjetima objekti   
     mijenjaju stanje ☑
331. Stanje pokazuje uslove objekta u jednom trenutnku vremena ☑
332. Sta cini potpis na poruce
333. Naziv poruke ☑
334. Lista parametara ☑
335. Lista poruka ☐
336. Aktivnosti izrade sekvencijalnog dijagrama su
337. Definiranje poruka ☑
338. Definiranje klasa ☐
339. Definiranje indikatora sinhronizacije ☑
340. Definiranje aktera i objekata ☑
341. Fizicka komponenta <<artifacts>> je sve ono sto se moze pohraniti na hard disk
342. Tacno ●
343. Netacno ○
344. Koje dijagrame koristimo kao izvor informacija za kretanje dijagrama saradnje
345. Dijagram komponenti ○
346. Sekvencijalni dijagram ●
347. Dijagram stanja ○
348. Kolaboracijski dijagram ○
349. Sekvencijalni dijagrami cine
350. Akcije ☐
351. Aktivnosti ☐
352. Objekti-akteri ☑
353. Objekti ☑
354. Kojim znakom se u dijagramu saradnje oznacava brojac
355. Tackom . ○
356. Plusem + ○
357. Zvjezdicom \* ●
358. Minusom - ○
359. Odaberi tacne tvrdnje dijagrama aktivnosti
360. Koristi se za dizajn poslovnih i softverskih procesa ☐
361. Koristi se za analizu poslovnih i softverskih procesa ☑
362. Koristi za analizu kompleksnih operacija, poslovnih pravila  
     te slucajeva upotrebe ☑
363. Modelira dinamicke aspekte sistema i predstavlja ponasanje  
     sistema ☑
364. Ekvivalentan je organigramu ili dijagramu toka podataka ☑
365. Koja je uloga stereotipa u cvorovima/node-a
366. Objasnjava veze izmedju cvorova ○
367. Objasnjava ulogu cvora ●
368. Povezuje dva cvora ○
369. Dijagram stanja se zasniva na teoriji
370. Grafova ○
371. Automata ●
372. Struna ○
373. Relativnosti ○
374. Tipovi dogadjaja
375. Time event ☑
376. Change event ☑
377. Signal eventi ☑
378. Call event ☑
379. Reply event ☐
380. <<artifacts>> - fizicke datoteke koje implementiraju komponentu
381. Tacno ●
382. Netacno ○
383. Guard condition je logicki izraz
384. Ako je guard condition prisutan, mora biti true u poretku da se   
     tranzicija (prelaz) dogodi ☑
385. kada događaj pobuđuje (okida) tranziciju (prelaz) ☑
386. ako je “guard condition” prisutan, moze, ali i ne mora biti true u   
     poretku da se tranzicija (prelaz) dogodi. ☐
387. guard condition je prikazan duž strelice prelaza ☑
388. Odaberi minimalnu brojnost ulaznih i izlaznih tranzicija iz/u udruzivanje u UML dijagramima stanja
389. Minimalna brojnost za ulaz je jedan ili vise, te za izlaz jedan ●
390. Minimalna brojnost za ulaz je jedan, te za izlaz jedan ili vise ○
391. Sta opisuju interakcijski dijagrami?
392. Sta sistem radi ○
393. Kako sistem radi ●
394. Krajnje stanje pseudonima – Prelaz stanja izvrsava se automatski odmah nakon kreiranja objekta
395. Netacno ●
396. Tacno ○
397. Sta predstavlja ime ucesnika o[i]?
398. Objekt imena o koji se selektira indeksom ●
399. Klasa imena o koji se se selektira indeksom ○
400. Anonimni object imena o koji se selektira indeksom ○
401. Tacka odluke
402. Ima jedan i samo jedan ulazni tok i jedan izlazni tok podataka ○
403. Ima jedan i samo jedan ulazni tok i vise izlaznih tokova podataka ●
404. Ima vise ulaznih tokova i jedan i samo jedan izlazni tok podataka ○
405. Ima vise ulaznih i vise izlaznih tokova podataka ○
406. Signalni dogadjaj je
407. Asinhroni - predajnik ne čeka da prijemnik primi signal ili reaguje   
     na njega. ●
408. Sihroni - predajnik čeka da prijemnik primi signal ili reaguje na   
     njega. ○
409. U dijagramu saradnje moguce je izmjeniti smjer poruke ukoliko se radi o konstruktoru ili destruktoru
410. Tacno ○
411. Netacno ●
412. U kojoj fazi razvoja softvera je dijagram stanja koristan
413. Samo u fazi dizajna ○
414. U fazama analize i dizajna ●
415. Samo u fazi analize ○
416. Sta predstavlja ime ucesnika o:C
417. Anonimni object imena C, klase o ○
418. Objekt imena o, klase C ●
419. Anonimni object imena o, klase C ○
420. Objekt imena C, klase o ○