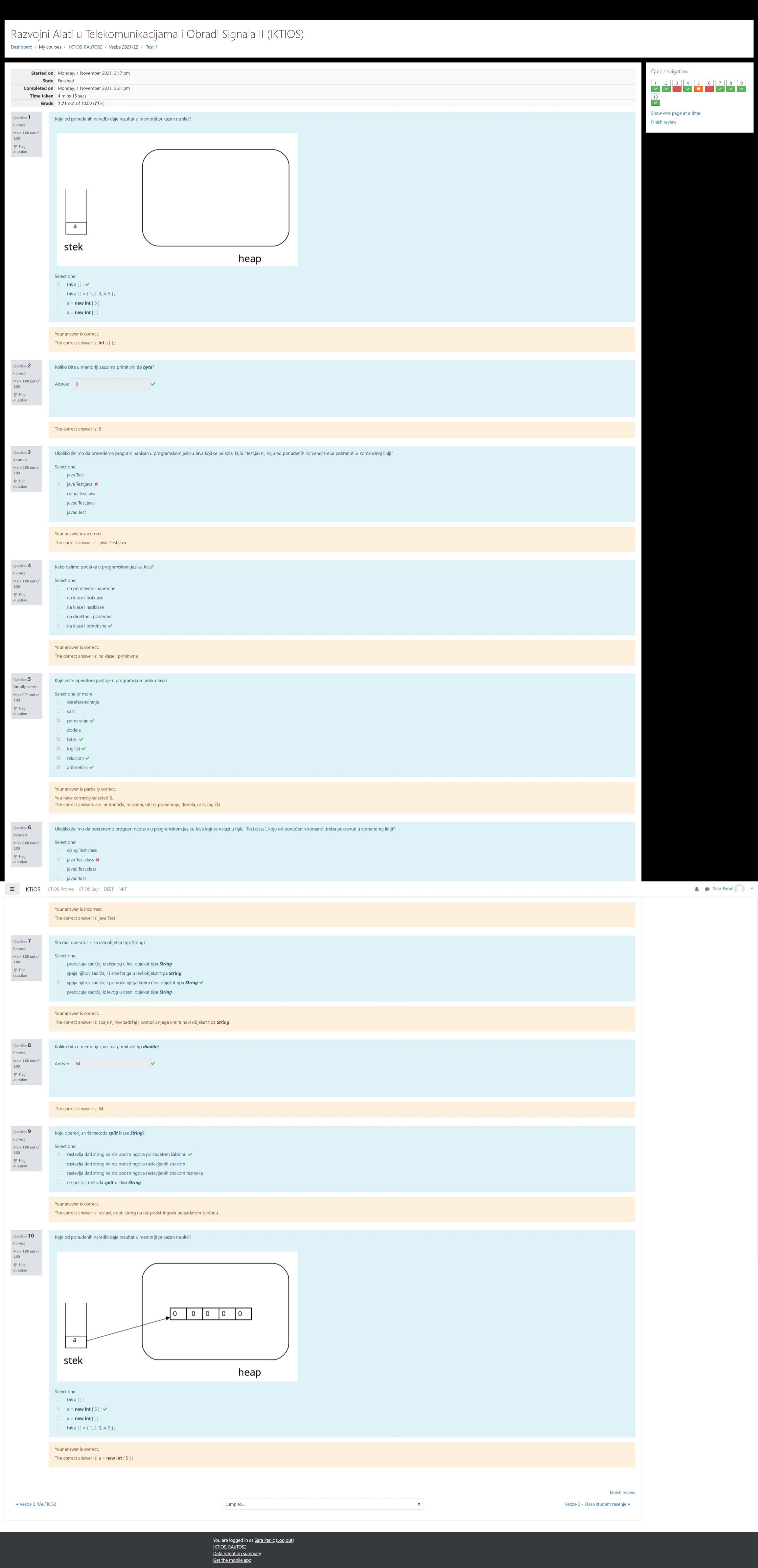
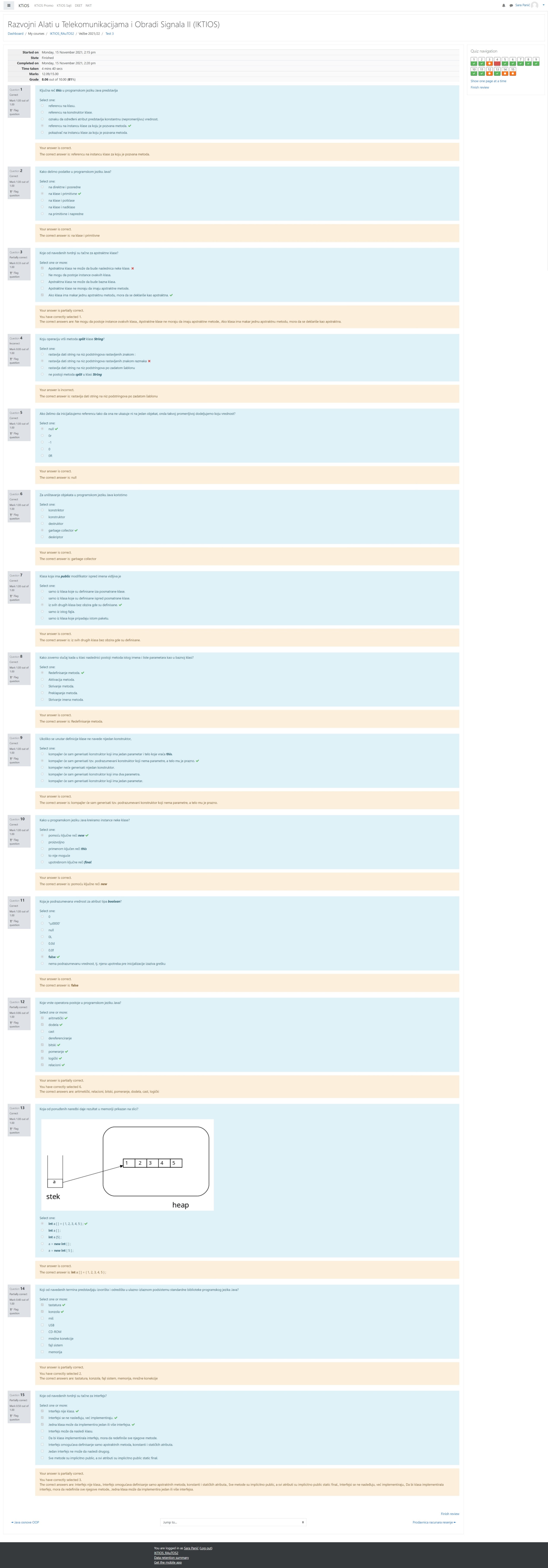
**TESTOVI 1-6**

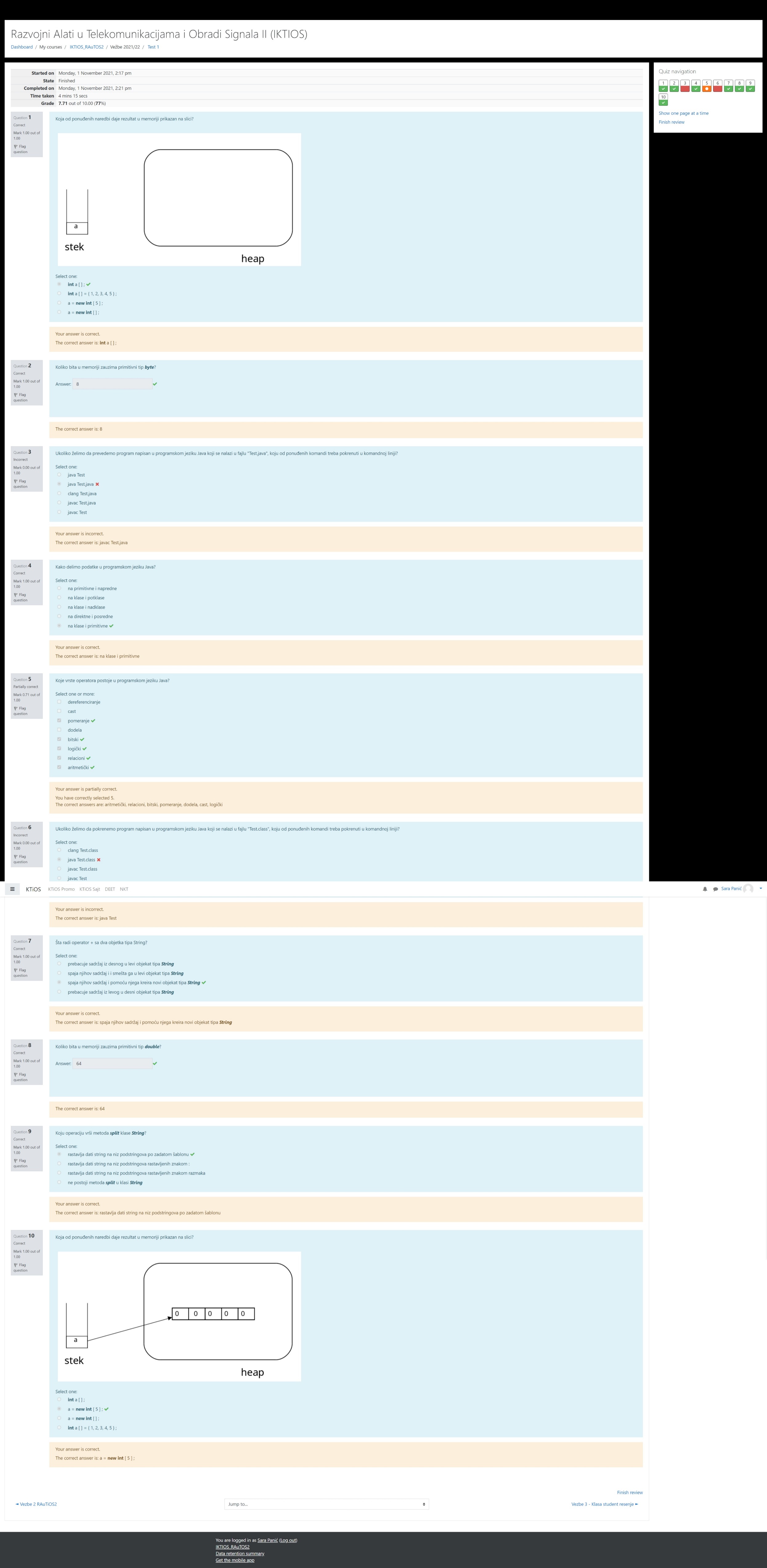
1. Koja od ponuđenih naredbi daje rezultat u memoriji prikazan na slici



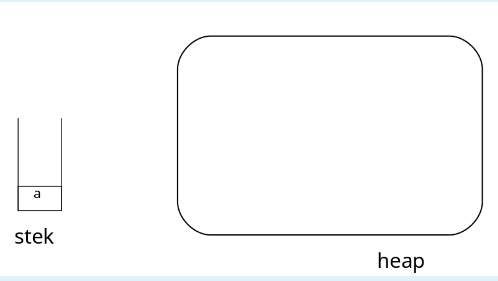
1. int a[];
2. int a[] = {1,2,3,4,5};
3. a = new int [5]
4. a = new int[];
5. Koja od ponuđeni naredbi daje rezultat u memoriji prikazan na slici?



1. int a[];
2. a = new int [5];
3. a = new int [];
4. int a [] = {1,2,3,4,5};
5. Koja od ponuđeni naredbi daje rezultat u memoriji prikazan na slici?



1. int a[];
2. a = new int [5];
3. a = new int [];
4. int a [] = {1,2,3,4,5};
5. Koja od ponudjenih naredbi daje rezultat u memoriji prikazan na slici?



1. Int a[] = {1,2,3,4,5};
2. a=new int [];
3. int a[];
4. a=new int[5];
5. Koliko bita u memoriji zauzima primitivni tip byte?8
6. Koliko bita u memoriji zauzima primitivni tip short? 16
7. Koliko bita u memoriji zauzima primitivni tip float? 32
8. koliko bita u memoriji zauzima primitivni tip **int**? 32
9. koliko bita u memoriji zauzima primitivni tip **boolean** u JVM? 32
10. Koliko bita u memoriji zauzima primitivni tip double?64
11. koliko bita u memoriji zauzima primitivni tip **long**? 64
12. Ukoliko želimo da prevedemo program napisan u programskom jeziku Java koji se nalazi u fajlu „Test.java“, koju od ponuđenih komandi treba pokrenuti u komandnoj liniji?
13. java Test
14. java Test.java
15. clang Test.java
16. javac Test.java
17. javac Test
18. Kako delimo podatke u programskom jeziku Java?
19. Na primitivne I napredne
20. Na klase I potklase
21. Na klase I nadklase
22. Na direktne I posrdne
23. Na klase I primitivne
24. Koje vrste operatora postoje u programskom jeziku Java?
25. Deferenciranje
26. Cast
27. Pomeranje
28. Dodela
29. Bitski
30. Logički
31. Relacioni
32. Aritmetički
33. Ukoliko želimo da pokrenemo program napisan u programskom jeziku Java koji se nalazi u fajlu „Test.java“, koju od ponuđenih komandi treba pokrenuti u komandnoj liniji?
34. clang Test.class
35. java Test.class
36. javac Test.class
37. javac Test
38. java Test
39. Prevodjenjem program napisanog u programskom jeziku Java, koji se nalazi u fajlu “Test.java”, nastaje koji od ponudjenih fajlova?
40. Test
41. javaTest.class
42. javaTest
43. Test.exe
44. Test.class
45. Ukoliko želimo da pokrenemo program napisan u programskom jeziku Java koji se nalazi u fajlu "Test.class", koju od ponuđenih komandi treba pokrenuti u komandnoj liniji?
46. javac Test
47. javac Test.class
48. clang Test.class
49. java Test.class
50. java Test
51. Šta radi operator + sa dva objekta tipa String?
52. Prebacuje sadržaj iz desnog u levi objekat tipa String
53. Spaja njihov sadržaj I smešta ga u levi objekat tipa Strin
54. Spaja njihov sadržaj I pomoću njega kreira novi objekat tipa String
55. Prebacuje sadržaj iz levog u desni objekat tipa String
56. Koju operaciju vrši metoda split klase String?
57. Rastavlja dati string na niz podstringova po zadatom šablonu
58. Rastavlja dati string na niz podstringova rastavljenih znakom :
59. Rastavlja dati string na niz podstringova rastavljenih znakom razmaka
60. Ne postoji metoda split u klasi String
61. Koja je podrazumevana vrednost za atribut tipa boolean?
62. 0.0d
63. false
64. ‘\u0000’
65. 0L
66. 0
67. Nema podrazumevanu vrednost,tj njena upotreba pre inicijalizacije izaziva grešku
68. Null
69. 0.0f
70. Za uništavanje objekata u programskom jeziku Java koristimo:
71. Konstruktor
72. Deskriptor
73. Garbage collector
74. Destruktor
75. Konstriktor
76. Ukoliko se unutar definicije klase ne navede nijedan konstruktor
77. Kompajler neće genersati nijedan konstruktor
78. Kompajler će sam generisati konstruktor koji ima jedan parametar I telo koje vraća **this**
79. Kompajler će sam generisati tzv podrazumevani konstruktor koji nema parametre a telo mu je prazno
80. Kompajler će sam generisati konstruktor koji ima dva parametra
81. Kompajler će sam generisati konstruktor koji ima jedan parametar
82. Koja je podrazumevana vrednost za atribut tipa byte?
83. 0
84. (byte) 0
85. ‘\u0000’
86. 0.0d
87. 0L
88. Null
89. 0.0f
90. False
91. Koja je podrazumevana vrednost za atribut tipa long?
92. '\u0000'
93. 0L
94. false
95. 0.0d
96. 0.0f
97. nema podrazumevanu vrednost, tj. njena upotreba pre inicijalizacije izaziva grešku
98. null
99. 0
100. Ključna reč **this** u programskom jeziku Java predstavlja
101. Referencu na klasu
102. Referencu na konstruktor klase
103. Oznaku da određeni atribut predstavlja konstantnu (nepromenljivu) vrednost
104. Pokazivač na instancu klase za koju je pozvana metoda
105. Referencu na instancu klase za koju je pozvana metoda
106. Ako želimo da inicijalizujemo referencu tako da ona ne ukazuje ni na jedan objekat, onda takvoj promenljivoj dodeljujemo koju vrednost?
107. null
108. -1
109. 0r
110. 0
111. 0R
112. Kako nazivamo instance neke klase u pogramskom jeziku Java?
113. Atributi
114. Metoe
115. Varijanse
116. Objekti
117. Varijable
118. Kako u programskom jeziku Java kreiramo instance neke klase?
119. To nije moguće
120. Pomoću ključne reči new
121. Primenom ključne reči this
122. Proizvoljno
123. Upotrebom ključne reči final
124. U programskom jeziku Java postojanje više metoda sa istim imenom
125. Je dozvoljeno ukoliko se međusobno razlikuju po povratnoj vrednosti
126. Je dozvoljeno ukoliko se međusobno ne razlikuju
127. Je dozvoljeno ukoliko se međusobno razlikuju po broju I ili tipovima parametara
128. Je dozvoljeno ukoliko se međusobno ne razlikuju po povratnoj vrednosti
129. Nije dozvoljeno
130. Koja od navedenih tvrdnji su tačne za apstraktne klase?
131. Apstraktna klasa ne može da bude naslednica neke klase
132. Ne mogu da postoje instance ovakvih klasa
133. Apstraktna klasa ne može da bude bazna klasa
134. Apstraktne klase ne moraju da imaju apstraktne metode
135. Ako klasa ima makar jednu apstraktnu metodu, mora da se deklariše kao apstraktna
136. Klasa koja ima public modifikator ispred imena vidljiva je
137. Samo iz klasa koje su definisane iza posmatrane klase
138. Samo iz klasa koje su definisane ispred posmatrane klase
139. Iz svih drugih klasa bez obzira gde su definisane
140. Samo iz istog fajla
141. Samo iz klasa koje pripadaju istom paketu
142. Kako zovemo slučaj kada u klasi naslednici postoji metoda istog imena I liste parametara kao u baznoj klasi?
143. Redefinisanje metoda
144. Aktivacija metoda
145. Skrivanje metoda
146. Preklapanje metoda
147. Skrivanje imena metoda
148. Koji od navedenih termina predstavljaju izvorišta I odredišta u ulazno izlaznom podsistemu standardne biblioteke programskog jezika Java?
149. Tastatura
150. Konzola
151. Miš
152. USB
153. CD ROM
154. Mrežne konekcije
155. Fajl sistem
156. Memorija
157. Koje od navedenih tvrdnji su tačne za interfejs ?
158. Interfejs nije klasa
159. Interfejsi se ne nasleđuju već se implementiraju
160. Jedna klasa može da implementira jedan ili više interfejsa
161. Interfejs može da nasledi klasu
162. Da bi klasa implementirala interfejs mora da redefiniše sve njegove metode
163. Interfejs omogućava definisanje samo apstraktnih metoda, konstanti I statičkih atributa
164. Jedan interfejs ne može da nasledi drugog
165. Sve metode su implicitno public, a svi atributi su implicitno public static final
166. Od kojih elemenata moze da se sastoji program napisan u programskom jeziku Java?
167. Definicija metoda
168. ‘import’ dela
169. Deklaracija paketa
170. Definicija globalnih funkcija
171. Deklaracija atributa
172. ‘Include’ dela
173. Deklaracija globalnih prmenljivih
174. Preambule
175. Promenljiva cija vrednost se ne moze menjati deklarise se na koji nacin?
176. final int x;
177. final int x=3
178. const boolean b;
179. const float a=3.14;
180. final char c = r;
181. kako se zove mehanizam za obradu gresaka u programskom jeziku Java?
182. Reference
183. Apstraktne klase
184. Izuzeci
185. Virtuelna masina
186. Grabage collector
187. Ako zelimo da metoda ima promenjiv broj parametara, definisemo jen akoji od ponudjenih nacina?
188. **int** f (**int** p[N]) {…}
189. f(**double**…d) {…}
190. **void** f (int…p) {…}
191. To ne moze u prog jeziku Java
192. **void** f(…q) {…}
193. koji osnovni koncepti na koje se oslanja ulazno-izlazni podsistem programskog jezika Java?
194. Citaci/psiaci (readers/writers)
195. Mesaci (mixers)
196. Punjaci (fillers)
197. Cevke (pipes)
198. Snimaci (recorders)
199. Ucitavaci (loaders)
200. Tokovi (streams)
201. Putevi (roads)
202. U programskom jeziku Java, pristup instancama kalsa moguc je na koje sve nacine?
203. Preko reference
204. Direktno
205. Preko pokazivaca
206. Preko posrednika
207. Preko nizova
208. Ukoliko zelimo da pokrenemo program napisan u programskom jeziku Java, sta nam je neophodno?
209. Java virtuelna masina
210. Kasperski antivirus
211. Ladno pivo
212. Java kompajler
213. Java ‘debugger’
214. U programskom jeziku Java, pristup instancama klasa moguc je na koje sve nacine?
215. Preko posrednika
216. Preko reference
217. Direktno
218. Preko nizova
219. Preko pokazivaca
220. Kodovanje je bilo koje preslikavanje jedne grupe simbola u drugu? True
221. Po teoremi o irelevanciji Yk je dovoljna statistika za određivanje Xk True
222. Ako N→∞i R<I(X;Y), moguće je dizajnirati kod takav da nema greške u prenosu. False
223. Kako se u programskom jeziku Java nazivaju instance klase?
224. varijanse
225. varijable
226. konstante
227. objektivi
228. objekti
229. Da li svaki konačni podskup **S⊆L2**, gde **L2** predstavlja skup svih realnih signala sa konačnom energijom, ima bazu (konačan skup signala preko kog mogu da se definišu svi ostali signali iz **S**). True
230. Ukoliko za svaku kodnu reč važi da su i svi ciklični pomeraji (ciklični pomeraj kodne reči x=[x1,x2,…,xN] za jedno mesto definisan je kao S(x,1)=[xN,x1,x2,…,xN−1]) takođe kodne reči, tada je kod linearan. True
231. U programskom jeziku Java ,, funkcije I promenljive moguce je definisati:
232. Iskljucivo unutar paketa
233. Proizvoljno
234. Promenljive iskljucivo unutar klasa, a funkcije proizvoljno
235. Iskljucivo van klasa
236. Iskljucivo unutar klasa
237. Prednosti koriscenja figure bloka jesu:
238. Grupisanje podlsika u jedan blok
239. Lako referensiranje slike u tekstu pomocu labele
240. Lako skaliranje slika
241. Dodavanje natpisa(eng. Caption)
242. Dinamicko pozicioniranje slika
243. Za prikazivanje slika koristimo paket:
244. Image
245. Picture
246. Figure
247. graphicx
248. Koliko bita u sekundi možemo preneti ukoliko nema šuma:
249. 3600
250. 0
251. 4800
252. ∞ (beskonačno)
253. Koliko jedna klasa može da ima klasa naslednica u programskom jeziku Java?
254. Proizvoljno mnogo
255. Zavisi od broja parametara u konstruktoru klase
256. Nijednu
257. Dve
258. Jednu
259. Koje od sledećih tvrdnji su tačne za statičke atribute I metode?
260. Statički atributi I metode postoje bez obzira na kreiranje objekata klase
261. Statički atributi I metode ne postoje bez kreiranja objekata klase
262. Statičkim atributima I metodama može se pristupiti preko imena klase
263. Statičkim atributima I metodama ne može se pristupiti preko imena klase
264. Statičkim atributima I metodama može se pristupiti preko instanci klase
265. Statičkim atributima I metodama ne može se pristupiti preko instanci klase
266. Statički atributi imaju istu vrenost za sve objekte pripadajuće klase
267. Statički atributi nemaju istu vrednost za sve objekte pripadajuce klase
268. Spektralna efikasnost AWGN kanala iznosi : True
269. Formule unutar teksta pišu se između:
270. begin{equation} I end{equation}
271. “i”
272. dva “$” znaka
273. normalno kao I običan tekst
274. Označi tačne tvrdnje:
275. Zbir redova generatorske matrice su validne kodne reči
276. Ako generatorska matrica ima dimenzije K x N, tada matrica provere parnosti ima dimenzije N – K x N
277. Redovi generatorske matrice su validne kodne reči
278. Redovi matrice provere parnosti su validne kodne reči
279. Zbir redova matrice provere parnosti su validne kodne reči
280. Matrica provere parnosti je generatorska matrica dualnog koda
281. U kom delu memorije se čuvaju instance klasa?
282. zavisi od platforme
283. na hard-disku
284. na steku
285. u statičkoj memoriji
286. u dinamičkoj memoriji (heap)
287. Ključna reč **super** u programskom jeziku Java predstavlja
288. oznaku da određeni atribut predstavlja konstantnu (nepromenljivu) vrednost.
289. pokazivač na instancu klase za koju je pozvana metoda.
290. referencu na instancu klase za koju je pozvana metoda.
291. referencu na baznu klasu u odnosu na instancu klase za koju je pozvana metoda.
292. referencu na konstruktor klase.
293. Koje od navedenih tvrdnji su tačne za apstraktne klase?
294. Ako klasa ima makar jednu apstraktnu metodu, mora da se deklariše kao apstraktna.
295. Ne mogu da postoje instance ovakvih klasa.
296. Apstraktna klasa ne može da bude naslednica neke klase.
297. Apstraktne klase ne moraju da imaju apstraktne metode.
298. Apstraktna klasa ne može da bude bazna klasa.
299. Svaki konvolucioni kod je linearan: True
300. Objekti klase **String**
301. se ne mogu menjati
302. mogu da se menjaju na određenim platformama
303. mogu samo da se uvećavaju
304. se mogu menjati
305. Signal se modeluje matematičkom funkcijom. True
306. Kod je linearan ako i samo ako je zbir bilo koje dve kodne reči takođe kodna reč. True
307. Svaki linearni blok kod može da se opiše preko matrice provere parnosti. True
308. Za bilo koji kanal bez memorije važi True
309. Koliko jedna klasa može da ima roditeljskih klasa u programskom jeziku Java?
310. Zavisi od broja prametara u konstruktoru klase.
311. Dve.
312. Nijednu.
313. Proizvoljno mnogo.
314. Jednu.
315. Koja je podrazumevana vrednost za lokalne promenljive?
316. 0.0d
317. null
318. '\u0000'
319. 0L
320. nema podrazumevanu vrednost, tj. njena upotreba pre inicijalizacije izaziva grešku
321. 0.0f
322. 0
323. False
324. Problem dizajniranja dobrog koda jednak je problemu pakovanja sfera.
325. True
326. False
327. Kodovanje je bilo koje preslikavanje jedne grupe simbola u drugu.
328. True
329. False
330. Šta je signal?
331. Električni ekvivalent poruke
332. Matematička funkcija
333. Merljiva promena fizičke veličine
334. Fizički ekvivalent poruke
335. Svaki konovluicioni kod je linearan.
     1. True
     2. False
336. Spektralna efikasnost AWGN kanala iznosi 1/2log2(1+SNR)
     1. True
     2. False
337. Svaki linearni blok kod može da se opiše preko matrice provere parnosti.
     1. True
     2. False
338. Da li svaki konačni podskup S⊆L2, gde L2 predstavlja skup svih realnih signala sa konačnom energijom, ima bazu (konačan skup signala preko kog mogu da se definišu svi ostali signali iz S).
     1. True
     2. False
339. Po drugoj Šenonovoj teoremi, ako N→∞ i R<I(X;Y) onda
     1. Pe→1
     2. Pe=0
     3. Pe→0
     4. Pe=1
340. Kapacitet BEC kanala iznosi ϵ, gde je ϵ verovatnoća brisanja.
     1. True
     2. False
341. Ako N→∞ i R<I(X;Y), moguće je dizajnirati kod takav da nema greške u prenosu.
     1. True
     2. False
342. Ako je Y(t)=X(t)+N(t), onda je Yk=Xk+Nk, gde je N(t) beli gausov šum, X(t) je slučajno odabran signal iz konačnog skupa realnih signala sa konačnom energijom, a Yk, Nk i Xk su uzorci ovih signala u trenutku t=kT.
     1. True
     2. False
343. 3,1 repetition kod i 3,2 SPC kod su međusobno dualni.
     1. True
     2. False
344. Signal se modeluje matematičkom funkcijom.
     1. True
     2. False
345. Svaka akcija nad komponentama korisnickog interfejsa izaziva:
346. Generisanje objekata klasa nasledjenih iz klase EventObject
347. Pokretanje nove niti u programu
348. Zatvaranje pratece niti u programu
349. Generisanje izuzetka tipa klase nasledjene iz klase Thread
350. Generisanje objekata klasa nasledjenih iz klase EventListener
351. Koji od ponudjenih odgovora predstavlja prednost biblioteke Swing nad AWT bibliotekom?
352. Biblioteka Swing je napisana u Javi pa se ne oslanja na operativni system
353. Biblioteka Swing je napisana u C++ pa je mnogo brza od AWT
354. Biblioteka Swing nije bazirana na MVC(ModelViewController) pa jemnogo jednostavija za koriscenje od AWT
355. Biblioteka Swing nudi kompletnu podrsku za Unicode standard
356. Za razliku od AWT, Swing jednako radi I izgleda na svim platformama
357. Koje od ponudjenih klasa omogucavaju upravljanje prostornim rasporedom grafickih komponenti?
358. FlowLayout
359. BorderLayout
360. InverseLayout
361. GridBagLayout
362. ReflectionLayout
363. GroupLayout
364. GridLayout
365. GarbageLayout
366. Od kojih komponenti se sastoji svaki program ciji graficki interfejs je realizovan primenom Swing biblioteke?
367. Metode koje implementiraju reakciju na dogadjaje nad komponentama korisnickog interfejsa
368. Inicijalizovani blok
369. Globalne funkcije koje implementiraju iscrtavanja sadrzaja na ekranu
370. Staticke metode koje implementiraju iscrtavanje sadrzaja na ekranu
371. Blokovi koda koji reaguju na prekide
372. Korisnicki interfejs u programskom jeziku Java je upravljan
373. Izuzecima
374. Nitima
375. Prekidima
376. Impulsima
377. Dogadjajima
378. Koja od ponudjenih kontejnerskih klasa u Swingg biblioteci pripadaju kontejnerima najviseg nivoa?
379. JDialog
380. JMeny
381. JComponent
382. JFrame
383. JApplet
384. Kako bi se olaksalo definisanje reakcija na dogadjaje za sve \*Listener interfejse uvedene su
385. Katalizator klase
386. Filter metode
387. Selektor metode
388. Filter klase
389. Adapter klase
390. Svaki program u kom se koristi Swing biblioteka pocinje rad sa dve niti. U jednoj niti se izvrsava metoda main, a u drugoj obradjuju dogadjaji. Kako se zove druga nit?
391. EventInterrupting Thread
392. EventManaging Thread
393. EventListening Thread
394. EventDispatching Thread
395. EventDisrupting Thread
396. Sve LaTeX komande pocinju sa  znakom
397. #
398. ~
399. \
400. =
401. Za pisanje matrica koristimo
402. Tabular blok
403. Array blok
404. Bmatrix blok
405. Cases blok
406. Za bilo koji kanal bez memorije vazi P[Y|X] = proizvod P[Yn|Xn] od n=1 do N

TACNO

1. Liste u python se pisu izmedju
2. Obicnih ()
3. ManjeVece <>
4. Viticastih {}
5. Uglastih []
6. Po teoremi o irelevanciji Yk je dovoljna statistika za odredjivanje Xk

TACNO

1. Komanda za podvucena slova (underline text) je
2. Textul
3. Textun
4. Emph
5. Underline
6. Alert
7. Pyton je
8. Dinamicki tipiziran
9. Netipiziran
10. Hibridan
11. Staticky tipiziran
12. Oznaci tacne tvrdnje
13. Redovi generatorske matrices u validne kodne reci
14. Matrica provere parnosti je generatorska matrica dualnog koda
15. Ako generatorska matrica je dimenzije KxN, onda matrica parnosti je dimenzije(N-K)xN
16. Zbir redova generatorske matrice u validne kodne reci
17. Zbir redova matrice provere parnosti su validne kodne reci
18. Redovi matrice provere parnosti su validne kodne reci
19. Svaki tekst se nalazi izmedju
20. Begin{document} I end{document}
21. Begin{frame} I end{frame}
22. Begin{equation} I end{equation}
23. Begin{text} I end{text}
24. Begin{figure} I end{figure}
25. Python sadrzi samo range-based for petlju

TACNO

1. Za ucitavanje slika koristimo komandu ~figure~imshow
2. IncludeGraphics!!!!!!!!
3. Begin{figure
4. Za prikazivanje slika koristimo paket
5. Figure
6. Graphicx
7. Image
8. Picture
9. Kodovanje je bilo koje preslikavanje jedne grupe simbola u drugu

TACNO

1. Staticki tipiziran znaci
2. Tip podataka se proverava prilikom izvrsavanja programa
3. Podatak moze da promeni tip
4. Tip podataka se proverava pre izvrsavanja na osnovu izvornog fajla
5. Podatak nema tip
6. Kod je linearan ako I samo ako je zbir bilo koje dve kodne reci takodje kodna rec

TACNO

1. Za svaka dva signala x(t) I y(t) nesto svih realnih signala sa konacnom energijom, koji imaju ortonormalne koeficijente x I y vazi
2. <x(t),y(t)> = d^2(x(t),y(t))
3. ||x(t)||^2 = ||x||^2
4. <x(t),y(t)> = <x,y>
5. <x(t),y(t)> = d^2(x,y)
6. Koriscenje koje metode clanice klase Thread se preporucuje za suspendovanje izvrsavanje niti
7. Destroy()
8. Resume()
9. Sleep()
10. Stop()
11. Pause()
12. Koje vrste niti postoje u pogledu zavisnosti izvrsavanja u odnosu na nit koja izvrsava funkciju main?
13. Sistemske
14. Konkurente
15. Korisnicke
16. Demonske
17. Graficke
18. Kako se zove metoda u kojoj se nalazi izvrsni kod koji se izvrsava u posebnoj niti?
19. Run()
20. Start()
21. Rum()
22. Init()
23. Go()
24. Kako se zove metoda clanica klase Thread kojom se pokrece nit?
25. Init()
26. Start()
27. Go()
28. Run()
29. Rum()
30. Koji je preporucen nacin zavrsavanja niti?
31. Pozivom metode kill()
32. Pozivom metode stop()
33. Pozivom metode destroy()
34. Zavrsavanjem funkcije start()
35. Zavrsavanjem funkcije run()
36. Na koje sve nacine je moguce kreirati aktivan objekat koji sadrzi vlastitu nit kontrole izvrsavanja
37. Kreiranjem instance klase Thread
38. Kreiranjem instance klase ThreadPool
39. Implementacijom interfejsa Runnable
40. Implementacijom interfejsa ExecutorService
41. Ako N -> ∞ I R<I(X;Y) moguce je dizajnirati kod takav da nema greske u prenosu

NETACNO

1. Python sadrzi DO WHILE

NETACNO

1. Oznaci tacne tvrdnje
2. Svaki symbol bilo koje kodne reci definisan je odgovarajucom kolonom generatorske matrice
3. Svaka kolona matrice provere parnosti definise jednacinu provere parnosti koju jedna kodna rec treba da zadovolji
4. Kolona generatorske matrice definise koje informacione bite je potrebno sabrati d bi dobili kodovani symbol
5. Ukoliko je generatorska matrica data u sistematskom obliku G=  [I|P] tada je matrica provere parnosti data u obliku H = [I] – P]
6. Svaki linearni blok nad poljem F I duzine N je linearni podprostor prostora F^N
7. Svaki red matrice provere parnosti definise jednacinu provere parnosti koju jedna kodna rec treba da zadovolji.
8. U javi definisemo funkcije I promenljive iskljucivo unutar KLASA
9. Skupovi u python se pisu izmedju
10. Obicnih ()
11. ManjeVece <>
12. Viticastih {}
13. Uglastih []
14. Problem dizajniranja dobrog koda jednak je problem pakovanja sfera

TACNO

1. 3,1 REPETITION I 3,2 SPC su medjusobno dualni

TACNO

1. Za podesavanje boje teksta koristimo paket
2. Textcolor
3. Graphicscolor
4. Xcolor
5. Color
6. Za dodavanje paketa koristimo komandu
7. Addpackage
8. Loadpackage
9. Import
10. Load
11. Userpackage
12. Importpackage
13. Svaki LaTeX document pocinje sa
14. Document
15. Enumerate
16. Documentclass
17. Equation
18. Figure
19. Neke validne dodatne opcije za documentclass jesu
20. A4paper
21. Article
22. 12pt
23. Letterpaper
24. Journal
25. Latin
26. Svaki konvolucioni kod je linearan

TACNO

1. Po drugoj Senonovoj teoremi ako Ako N -> ∞ I R<I(X;Y) onda
2. Pe = 0
3. Pe tezi 0
4. Pe tezi 1
5. Pe = 1
6. Spektralna efikasnost AWGN kanala iznosi 1/2log2(1+SNR)

TACNO

1. Svaki linearni blok kod moze da se opise preko matrice provere parnosti

TACNO

1. Da bi druga Senonova teorema vazila, mora N -> ∞

TACNO

1. Signal se modeluje matematickom funkcijom

TACNO

1. Glavne prednosti LaTeXa u odnosu na druge
2. Jednostavno koriscenje nestandardnih fontova
3. Jednostavan rad sa referencama
4. Odvajanje sadrzajaod strila
5. Lako kucanje matematickih izraza
6. Uniformno formatiranje u celom dokumentu
7. Koliko bita u sekundi mozemo preneti ukoliko NEMA suma
8. 0
9. ∞
10. 4800
11. 3600
12. Za definisanje podataka koristimo kljucnu rec var

NETACNO

1. Ako komanda ima dodatne opcije one se pisu izmedju
2. manjeVece <>
3. obicnih ()
4. uglastih []
5. viticastih {}

**NITI**

1. Kako se zove metoda u kojoj se nalazi izvršni kod koji se izvršava u posebnoj niti?
2. go( )
3. run( )
4. init( )
5. rum( )
6. start( )
7. Kako se zove metoda članica klase Thread kojom se pokreće nit?
8. go( )
9. run( )
10. init( )
11. rum( )
12. start( )
13. Koje varijante sinhronizacije podržava programski jezik Java?
14. Sinhronizacija nad fajlom.
15. Sinhronizacija nad blokom.
16. Sinhronizacija nad metodom.
17. Sinhronizacija nad objektom.
18. Sinhronizacija nad klasom.
19. Korišćenje koje metode članice klase Thread se preporučuje za suspendovanje izvršavanja niti?
20. destroy( )
21. stop( )
22. resume( )
23. sleep( )
24. pause(
25. Koji je preporučen način završavanja niti?
26. pozivom metode kill( )
27. pozivom metode destroy( )
28. završavanjem funkcije start( )
29. završavanjem funkcije run( )
30. pozivom metode stop( )
31. Na koje sve načine je moguće kreirati aktivan objekat koji sadži vlastitu nit kontrole izvšavanja?
32. Kreiranjem instance klase Thread.
33. Kreiranjem instance klase ThreadPool.
34. Implementacijom interfejsa Executor.
35. Implementacijom interfejsa ExecutorService.
36. Implementacijom interfejsa Runnable.
37. Koje vrste niti postoje u pogledu zavisnosti izvršavanja u odnosu na nit koja izvršava funkciju main?
38. Demonske.
39. Konkurentne.
40. Grafičke.
41. Sistemske.
42. Korisničke.

**GUI**

1. Koje od ponuđenih kontejnerskih klasa u Swing biblioteci pripadaju kontejnerima najvišeg nivoa?
2. JComponent
3. JDialog
4. JApplet
5. JMenu
6. JFrame
7. Kako bi se olakšalo definisanje reakcija na događaje za sve \*Listener interfejse uvedene su:
8. Adapter klase.
9. Filter klase.
10. Selektor metode.
11. Katalizator klase.
12. Filter metode.
13. Svaki program u kojem se koristi Swing biblioteka počinje rad sa dve niti. U jednoj niti se izvršava metoda "main", a u drugoj se obrađuju događaji. Kako se zove druga nit?
14. Event-Listening Thread
15. Event-Managing Thread
16. Event-Dispatching Thread.
17. Event-Interrupting Thread
18. Event-Disrupting Thread.
19. Svaka akcija nad komponentama korisničkog interfejsa izaziva:
20. generisanje objekata klasa nasleđenih iz klase EventObject.
21. zatvaranje prateće niti u programu.
22. generisanje izuzetka tipa klase nasleđene iz klase Thread.
23. generisanje objekata klasa nasleđenih iz klase EventListener.
24. pokretanje nove niti u programu
25. Korisnički interfejs u programskom jeziku Java je upravljan:
26. događajima.
27. prekidima.
28. impulsima.
29. izuzecima.
30. nitima.
31. Od kojih komponenti se sastoji svaki program čiji grafički interfejs je realizovan primenom Swing biblioteke?
32. Globalne funkcije koje implementiraju iscrtavanje sadržaja na ekranu.
33. Inicijalizacioni blok.
34. Blokovi koda koji reaguju na prekide.
35. Metode koje implementiraju reakciju na događaje nad komponentama korisničkog interfejsa.
36. Statičke metode koje implementiraju iscrtavanje sadržaja na ekranu.
37. Koji od ponuđenih odgovora predstavlja prednosti biblioteke Swing nad AWT (Abstract Window Toolkit) bibliotekom?
38. Biblioteka Swing je napisana u programskom jeziku C++ pa je mnogo brža od AWT.
39. Biblioteka Swing je napisana u Javi pa se ne oslanja na operativni sistem.
40. Biblioteka Swing nudi kompletnu podršku za Unicode standard.
41. Biblioteka Swing nije bazirana na MVC (Model-View-Controller) pa je mnogo jednostavnija za korišćenje od AWT.
42. Za razliku od AWT, Swing jednako radi i izgleda na svim platformama.
43. Koje od ponuđenih klasa omogućavaju upravljanje prostornim rasporedom grafičkih komponenti?
44. GarbageLayout
45. BorderLayout
46. FlowLayout
47. GroupLayout
48. InverseLayout
49. GridLayout
50. ReflectionLayout
51. GridBagLayout