

1. Koji su osnovni principi OOP?
2. Koji tipove veza između različitih klasa postoje u objektno orijentisanim programskim jezicima?
3. Kakvu vezu između klasa opisuje nasleđivanje?
4. Kakvu vezu između klasa opisuje kompozicija?
5. Kakvu vezu između klasa opisuje asocijacija?
6. Šta je klijent-server model?
7. Šta je serverska aplikacija i koja je njena uloga u klijent/server modelu?
8. Šta je klijentska aplikacija i koja je njena uloga u klijent/server modelu?
9. Kako se obavlja komunikacija između klijenta i servera?
10. Koji su osnovni protokoli koji se koriste za komunikaciju u web aplikacijama?
11. Šta su portovi? Šta oni omogućavaju u slučaju da se na računaru pokreće više mrežnih aplikacija?
12. Koji je osnovni paket i osnovne klase koje se koriste za razvoj mrežnih aplikacija u Javi?
13. Šta su i čemu služe **Socket-i**?
14. Koja je razlika između ServerSocket-a i Socket-a?
15. Šta je **server loop**?
16. Šta je HTTP protokol?
17. Koje su osnovne HTTP metode? I kada se koja od njih koristi?
18. Šta su response statusi? Koji je smisao grupa response statusa (100+, 200+, 300+, 400+, 500+)?
19. Šta je i koja je namena Spring frameworka?
20. Šta je, kako se koristi i šta omogućava korišćenje anotacija u Spring-u?
21. Šta je i šta omogućava "ubrizgavanje zavisnosti" (**Dependency injection**) u Spring-u?
22. Šta su "upravljani objekti" (**manage objects, managed beans**) u Springu?
23. Navesti nekoliko najbitnijih modula Spring frameworka
24. Koje osnovne principe pri razvoju softvera treba težiti da se koriste?
25. Koji su osnovni ISO-jevi tipične (troslojne) arhitekture web aplikacija?
26. Šta je MVC arhitektura web aplikacije?
27. Koje komponente čine **model** u MVC?
28. Koje komponente čine **view** u MVC?
29. Koje komponente čine **controller** u MVC?
30. Šta je Spring Boot i koja je prednost njegovog korišćenja u odnosu na "obični" Spring?
31. Koja je osnovna arhitektura (slojevi) Spring Boot aplikacije?
32. Koje su osnovne komponente Spring (Boot) aplikacije i koja je uloga svake od tih vrsta komponenti?
33. Koja je putanja obrade zahteva (kroz koje komponente prolazi) u Spring Boot aplikaciji?
34. Koje su osnovne Spring Boot anotacije i šta one omogućavaju?
35. Šta je Spring Initializr i čemu služi?
36. Šta je i čemu služi Servlet Dispatcher u Spring Boot aplikaciji?
37. Koje klase učestvuju u funkcionisanju sloja za perzistiranje podataka u Spring Boot aplikacije?
38. Šta je Java JPA?
39. Šta je objektno-relaciono mapiranje?
40. Koje su, čemu služe JPA anotacije?
41. Šta je Spring Data JPA?
42. Koja je osnovna uloga **Controller** komponente?
43. Kako se unutar kontrolera definiše koju putanju opslužuje koja *handler* metoda?
44. Čemu služi @RequestMapping anotacija?
45. Čemu služi objekat tipa **ModelAndView**?
46. Čemu služi @RequestParam anotacija?
47. Koja je razlika između @PathVariable i @RequestParam?
48. Čemu služi @ResponseBody anotacija. U kom formatu se najčešće šalju podaci u body-ju zahteva?
49. Čemu služi @RestController anotacija i koja je ključna razlika u odnosu na "standardni" kontroler?
50. Koja je namena servisnog sloja u Spring Boot aplikaciji?
51. Čemu služi @Autowired anotacija u Spring Boot komponentama?
52. Šta je i čemu služi HTML?
53. Koja je minimalna struktura HTML dokumenta (koje elemente minimalno sadrži)?

54. Šta je tag, šta element, a šta atribut u HTML?
55. Koja je razlika između blok i inline elemenata u HTML-u?
56. Kako (kojim elementom) se formiraju veze (linkovi) između HTML dokumenata (i drugih online resursa).
57. Kojim elementom se u HTML dodaje informacija o slici koju treba prikazati korisniku?
58. Koji formati slika se koriste na webu?
59. Kako se u HTML dodaju video i audio sadržaji?
60. Kako je SVG i šta on omogućava?
61. Čemu služi Canvas element?
62. Kako se u HTML-u formiraju tabele?
63. Kako se u HTML formiraju korisničke forme za unos podataka?
64. Zašto je **name** atribut obaveza za sve input, textarea i select elemente u formama?
65. Šta je CSS i čemu služi?
66. Šta su CSS selektori i koje osnovne vrste selektora razlikujemo?
67. Šta su i čemu služe pseudo selektori?
68. Kako se formiraju selektori za potomke?
69. Šta su apsolutne, a šta relativne jedinice mera i koje se koriste u CSS-u?
70. Šta je responsive design?
71. Da bi se postigao responsive design da li je poželjno koristiti relativne ili apsolutne jedinice za definisanje izgleda stranice?
72. Šta su i čemu služe **media-query**
73. Šta je i čemu služi JavaScript?
74. Zašto je JavaScript dinamički tipiziran jezik?
75. Koja je razlika između operatora == i === u JavaScript-u?
76. Na koje načine je moguće definisati JavaScript funkciju?
77. Šta su callback funkcije?
78. Koji je najčešći način za kreiranje objekata u JavaScriptu? Na koje još načine je to moguće uraditi?
79. Šta su anonimne funkcije?
80. Šta su samopozivajuće funkcije?
81. Čemu služi i kako se koristi spread operator?
82. Čemu služi i kako se koristi operator za destrukuirisanje?
83. Šta je i čemu služi AJAX?
84. Šta je i čemu služi XMLHttpRequest objekat?
85. Šta je i čemu služi Fetch API?
86. Šta je i čemu služi Promise objekat?