

1. Що таке паралельні обчислення?
2. Яка вимога повинна виконуватися для досягнення паралелізму?
3. Яка із класифікацій обчислювальних систем є найбільш поширеною?
4. Чим характеризуються мультипроцесорні системи?
5. Чим характеризуються мультикомп'ютерні системи?
6. Що таке "комп'ютерний кластер"?
7. Що є притаманним для метакомп'ютера?
8. Що таке декомпозиція задачі?
9. Що використовується під час проектування та написання паралельних програм підхід на основі ітеративного паралелізму?
10. Що використовується під час проектування та написання паралельних програм підхід на основі рекурсивного паралелізму?
11. Що використовується під час проектування та написання паралельних програм модель "виробники та споживачі"?
12. Що використовується під час проектування та написання паралельних програм модель "Клієнти та сервери"?
13. Що використовується під час проектування та написання паралельних програм модель "Керівник та робітники"?
14. Що використовується під час проектування та написання паралельних програм модель "Взаємодіючі рівноправні"?
15. Що таке дрібномодульна неподільна дія?
16. Що використовується для вирішення задачі критичної секції?
17. Що таке об'єкт взаємного виключення «mutex»?
18. Для чого використовується синхронізація у паралельних програмах?
19. Що таке «семафор» у паралельному програмуванні?
20. Що таке «монітор» у паралельному програмуванні?
21. Що таке критична секція у термінах паралельної програми?
22. За якої умови два процеси не впливають один на одного?
23. Для чого використовується алгоритм квитка у паралельних обчисленнях?

24. Для чого використовується алгоритм поліклініки у паралельних обчисленнях?
25. За яких умов паралельна програма переходить у стан гонок?
26. Яку проблему описує задача про виробників та споживачів?
27. Для чого використовуються кільцеві буфера у паралельних обчисленнях?
28. Яку проблему описує задача про філософів, що обідають?
29. У чому полягає суть взаємного блокування процесів?
30. Що потрібно зробити щоб запустити у Java потік на виконання?
31. Що потрібно зробити з кодом, що має бути виконаний у новому потоці у Java?
32. Як у Java можна призупинити виконання потоку на заданий проміжок часу?
33. Чим у Java відрізняється звичайний потік від потоку демона?
34. Які існують способи створення нового потоку засобами мови Java?
35. Яке призначення методу `run()` з інтерфейсу `Runnable`?
36. Чому під час створення нового потоку необхідно надавати перевагу реалізації інтерфейсу `Runnable`, а не наслідуванню від класу `Thread`?
37. Яким чином запустити створений потік на виконання?
38. Як можна згрупувати декілька потоків між собою?
39. Яку функціональну роль відіграє головний потік у мові Java?
40. У яких станах можуть перебувати потоки у мові Java?
41. Яким чином можна дізнатися поточний стан потоку?
42. Яким чином можна призупинити виконання певного потоку на вказаний проміжок часу?
43. У якому випадку може бути згенерований виняток `InterruptedException`?
44. Яким чином зробити так, щоб певний потік гарантовано завершав своє виконання останнім?
45. Як можна перевірити, чи певний потік ще не завершив своє виконання?
46. У яких випадках може бути корисним використання методу `yield()`?
47. До чого призводить виклик методу `yield()` із об'єкта потоку?

48. Як можна змінити пріоритет потоку у програмі?
49. Які значення є коректними під час задання пріоритету потоку?
50. Яке значення рівня пріоритету мають потоки по замовчуванню?
51. Який потік називається потоком-демоном?
52. Що потрібно для створення потоку-демона?
53. Що таке “монітор” у термінології потоків мови Java?
54. Які у Java існують механізми захисту критичної секції коду від паралельного доступу?
55. Який вид блокування називається реентрабельним?
56. Що потрібно для блокування потоку через об'єкт класу ReentrantLock?
57. Що потрібно для розблокування потоку через об'єкт класу ReentrantLock?
58. Яке призначення ключового слова synchronized у мові Java?
59. Які методи дозволяється оголошувати із ключовим словом synchronized?
60. Яке обмеження притаманне вбудованим блокуванням у мові Java?
61. Яким чином можна оголосити синхронізований блок засобами мови Java?
62. У яких випадках доцільніше для синхронізації використовувати об'єкти типу ReentrantLock?
63. Які існують механізми міжпотоківих комунікацій у мові Java?
64. Яким чином виконується міжпотоківих комунікація з використанням неявного монітора об'єкта?
65. Яким чином виконується міжпотоківих комунікація з використанням високорівневих методів взаємодії між потоками?
66. Яка існує відмінність у виклику методів signal() та signalAll()?
67. Яка існує відмінність у виклику методів notify() та notifyAll()?
68. У яких випадках рекомендується перевизначити методи wait() та notify() з класу Object?
69. Що використовується для створення умовних об'єктів типу Condition?
70. Який режим синхронізованого доступу до спільних даних називається не блокуючим?
71. У яких випадках використовуються оператор volatile?

- 72. У яких випадках у багатопотокових програмах рекомендується використовувати оператор `final`?**
- 73. У чому полягає перевага використання пулу потоків?**
- 74. Як у Java можна створити пул потоків?**
- 75. Яка відмінність між способами створення потоків з використанням інтерфейсів `Runnable` та `Callable`?**
- 76. Яке призначення методу `call()` з інтерфейсу `Callable`?**
- 77. Яким чином передати пулу потоків на виконання список задач?**
- 78. У яких випадках необхідне використання планувальників задач?**
- 79. Для чого використовується закон Джина Амдала в паралельних обчисленнях?**
- 80. Згідно із законом Джина Амдала:**