

1. У чому полягає суть успадкування?
2. Що таке клас-предок?
3. Що таке клас-нащадок?
4. Для чого використовують класи-предки?
5. Скільки класів-предків може мати клас?
6. Як описати клас-нащадок?
7. Що таке перевизначення методів?
8. Як перевизначити метод класу-предка?
9. Як здійснити доступ до перевизначених методів класу-предка?
10. Чи має об'єкт-нащадок доступ до перевизначених методів класу-предка?
11. Які члени класу-предка є доступними для класу-нащадка?
12. Чи має можливість клас-нащадок змінювати область видимості членів класу-предка?
13. Як описати члени класу-предка, щоб вони були доступними для класу-нащадка і недоступними у програмі-клієнті?
14. Як описати члени класу-предка, щоб вони не були доступними як для класу-нащадка, так і для програми-клієнта?
15. Чи може змінна типу класу-предка приймати значення об'єкта-нащадка?
16. До яких полів і методів об'єкта-нащадка може здійснити доступ об'єкт-предок?
17. Які види масивів можна створювати в C#?
18. Як оголошуються різні види масивів?
19. Оператор `foreach` – його синтаксис та приклади роботи з масивами.
20. Елементи класу: `Rank`, `Length`, `GetLength`, `Clear`, `Sort`, `Clone`, `IndexOf`, `Reverse`, `GetLowerBound`, `GetUpperBound`, `GetType`. Їх опис (що вони виконують).
21. Загальний вигляд опису одновимірного масиву.
22. Ініціалізатори одновимірних масивів.
23. Виділення пам'яті для одновимірних масивів.
24. Введення/виведення одновимірних масивів.
25. Звертання до елементів одновимірного масиву.
26. Копіювання одновимірних масивів.
27. Опис прямокутних багатовимірних масивів.
28. Виділення пам'яті для прямокутних багатовимірних масивів.
29. Ініціалізатори прямокутних масивів.
30. Звертання до елементів прямокутних масивів.
31. Введення/виведення прямокутних масивів.
32. Опис багатовимірних масивів з різними довжинами рядків.

33. Виділення пам'яті для багатовимірних масивів з різними довжинами рядків.
34. Що таке інтерфейс?
35. Що спільного між інтерфейсами та абстрактними класами?
36. Чим відрізняються інтерфейси від абстрактних класів?
37. Який інтерфейс є базовим для всіх інтерфейсів?
38. Що повинен містити клас, який реалізує інтерфейс?
39. Чи обов'язково клас, який реалізує інтерфейс має містити реалізації всіх його методів?
40. Чи можна змінній типу інтерфейс надавати значення об'єкта типу класу, в якому реалізовано цей інтерфейс?
41. Чи може інтерфейс мати багато реалізацій?
42. У чому відміна колекцій від масивів?
43. Назвіть основні властивості та методи класу `System.Array`.
44. Наведіть приклади опису масивів та колекцій.
45. Як передавати та повертати масиви та колекції з методів.
46. Поясніть принцип роботи циклу `foreach`.
47. Розкажіть про плюси та мінуси використання двозв'язних списків.
48. Наведіть практичні приклади ефективного використання розглянутих колекцій.