

Об'єктно-орієнтовне програмування (КР)

[Особистий кабінет](#) / [Мої курси](#) / [Об'єктно-орієнтовне програмування \(КР\)](#) / [Загальне](#) / [Перелік основних питань](#)

Перелік основних питань

1. Відмінності C++ від C.
2. Несумісності C та C++.
3. Стили програмування.
4. Елементи ООП.
5. Переваги та недоліки ООП.
6. Принципи ООП.
7. Глобальні функції. Означення. Приклад.
8. Користувальні та вбудовані функції. Приклад.
9. Прототип функції. Приклад.
10. Визначення функції. Приклад.
11. Глобальні та локальні змінні.
12. Перевантаження функції. Приклад.
13. Вбудовані функції. Ключове слово inline. Приклад.
14. Розподіл оперативної пам'яті при виклику функції.
15. Поняття об'єкту. Стан об'єкту. Поведінка об'єкту.
16. Операції над об'єктами.
17. Активні та пасивні об'єкти.
18. Відношення між об'єктами. Відношення використання.
19. Поняття синхронізації. [Класифікація](#) об'єктів.
20. Відношення включення між об'єктами.
21. Поняття [класу](#). Складові частини інтерфейсу [класу](#).
22. Змінні-члени та функції-члени [класу](#). Приклад.
23. Відношення [наслідування](#) між [класами](#).
24. Відношення використання між [класами](#).
25. Відношення між [класами](#).
26. Критерії оцінки взаємозв'язку [класів](#) та об'єктів.
27. Оголошення [класу](#). Приклад.
28. Визначення об'єкту. Приклад.
29. Доступ до членів [класу](#). Приклад.
30. Закриті члени [класу](#). Приклад.
31. Відкриті члени [класу](#). Приклад.
32. Відкриті методи доступу. Приклад.
33. Реалізація методів [класу](#). Приклад.
34. Поняття [конструктора](#). Приклад.
35. Поняття [деструктора](#). Приклад.
36. Постійні функції-члени. Приклад.
37. Поняття файлу заголовку.
38. Вбудована реалізація методів [класу](#). Приклад.
39. Структури в мові C++.
40. [Вказівники](#). Оператор непрямого доступу.
41. Стек та динамічна пам'ять.
42. Оператор new .
43. Оператор delete.
44. Створення та видалення об'єктів в області динамічної пам'яті.
45. Доступ до змінних-членів об'єкту в динамічній пам'яті.
46. [Вказівник](#) this.
47. [Вказівники](#) та константи.
48. Арифметичні операції з [вказівниками](#).
49. Означення [посилання](#). Основні його властивості.
50. Передача функції аргументів за [посиланням](#). Приклад.
51. Повернення з функції більше ніж одного значення. Приклад.
52. Передача у функцію [посилання](#) на постійний об'єкт.

53. Перевага посилань над [вказівниками](#).
54. Перевизначення унарних операцій. Приклад.
55. Перевизначення бінарних операцій. Приклад.
56. [Перевизначена операція](#) додавання. Приклад.
57. [Перевизначена операція](#) присвоєння. Приклад.
58. [Перевизначена операція](#) доступу до члена [класу](#). Приклад.
59. Загальні правила перевизначення.
60. Дружні функції. Приклад.
61. Визначення базового та похідного [класів](#) при наслідуванні. Приклад.
62. Робота [конструктора](#) та [деструктора](#) при наслідуванні. Приклад.
63. Пере визначення функцій в похідному [класі](#) при наслідуванні. Приклад
64. Типи [наслідування](#). Приклад.
65. Визначення віртуальної функції. Приклад.
66. [Віртуальна функція](#). V-таблиця. Недоліки віртуальних функцій.
67. [Поліморфізм](#). Варіанти вирішення проблеми [наслідування](#). Приклад.
68. [Множинне наслідування](#). Приклад.
69. Означення віртуального базового [класу](#). Приклад.
70. Правило домінування при використанні віртуальних базових [класів](#).
71. [Класи](#) можливостей. Приклад.
72. Абстрактні типи даних. Приклад.
73. Чисті віртуальні функції. Приклад.
74. Означення [шаблону](#). Приклад.
75. Оголошення [шаблону](#). Приклад.
76. Функції [шаблону](#). Приклад.
77. [Шаблони](#) та спеціалізація. Приклад.
78. [Шаблони](#) та дружні функції. Приклад.
79. Стандартна бібліотека [шаблонів](#) STL.
80. Означення [контейнера](#). [Класифікація](#) [контейнерів](#).
81. Послідовні [контейнери](#). Види послідовних [контейнерів](#).
82. [Контейнер](#) вектор. Приклад.
83. [Контейнер](#) список. Приклад.
84. [Контейнер](#) стек. Функції роботи зі стеком.
85. [Класифікація](#) помилок.
86. Варіанти, які можливі при виникненні помилок.
87. Ідея використання виключень.
88. Етапи застосування виключень
89. Використання блоку **try**
90. Використання блоку **catch** .
91. Обробка виключень.
92. Реалізація виключень при виникненні декількох помилок.
93. Об'єктно-орієнтований аналіз, мова моделювання.
94. Об'єктно-орієнтований аналіз, процес, метод.
95. Об'єктно-орієнтований аналіз. Циклічний процес проектування.
96. Об'єктно-орієнтований аналіз. Каскадний процес проектування.
97. Об'єктно-орієнтований аналіз. Екстремальне програмування.
98. Об'єктно-орієнтований аналіз. Аналіз вимог, визначення прецеденту.
99. Об'єктно-орієнтований аналіз. Аналіз вимог, визначення виконавців.
100. Об'єктно-орієнтований аналіз. Створення моделі сфери діяльності.
101. Об'єктно-орієнтований аналіз. Основні зв'язки, що документуються при аналізі (входження, асоціація, узагальнення).
102. Об'єктно-орієнтований аналіз. Визначення сценаріїв.
103. Об'єктно-орієнтований аналіз. Визначення критеріїв.
104. Об'єктно-орієнтований аналіз. Діаграми взаємодій.
105. Об'єктно-орієнтований аналіз. Візуалізація.
106. Об'єктно-орієнтований аналіз. Артефакти.
107. Об'єктно-орієнтоване проектування.
108. Взаємозв'язок [класів](#) при об'єктно-орієнтованому проектуванні.
109. Схема послідовності дій при об'єктно-орієнтованому проектуванні.
110. Схема взаємодії при об'єктно-орієнтованому проектуванні.
111. Схема станів при об'єктно-орієнтованому проектуванні.
112. Статичні дані-члени. Приклад.
113. Статичні функції-члени. Приклад.

[◀ Методичні вказівки до виконання курсової роботи](#)

Перейти до...

[Система оцінювання ▶](#)

[✉ Зверніться до служби підтримки сайту](#)

Ви зайшли під ім'ям Легеза Іван (Вийти)

О-О-П

Підсумок збереження даних

Завантажте мобільний додаток

Ця сторінка: General type: incourse. Context Сторінка: Перелік основних питань (context id 3625515). Page type mod-page-view.