

Об'єктно-орієнтоване та узагальнене програмування мовою C++

1. Які є способи передавання аргументів функціям?
2. Поясніть синтаксис і семантику передавання аргументів за значенням.
3. Поясніть синтаксис і семантику передавання аргументів за адресою.
4. Поясніть синтаксис і семантику передавання аргументів через посилання.
5. Що таке локальні змінні, статичні локальні змінні функції? (Оголошення, використання, видимість, час існування, приклади).
6. Передавання одновимірного масиву функції (з указанням розміру, з указанням діапазону). Пояснити на прикладі.
7. Що таке *клас*, *об'єкт*? *поле даних*, *змінна класу*? *метод (екземпляра)*, *метод класу*?
8. Що таке *конструктор*, *деструктор*?
9. Які різновиди конструкторів (у C++) Ви знаєте?
10. Які завдання вконує деструктор? Чи може він бути віртуальним?
11. У яких випадках використовується конструктор за замовчуванням? Наведіть його синтаксис.
12. У яких випадках використовується конструктор визначення? Наведіть його синтаксис.
13. У яких випадках використовується конструктор копіювання? Наведіть його синтаксис.
14. У яких випадках перевизначають конструктор копіювання? Наведіть його синтаксис.
15. У яких випадках перевизначають оператор присвоєння? Наведіть його синтаксис.
16. У яких випадках перевизначають деструктор? Наведіть його синтаксис.
17. Які члени класу генеруються автоматично? У яких випадках автоматичні члени класу не забезпечують потрібної функціональності?
18. Керування доступом до елементів класу C++. Правила видимості елементів класу.
19. Оголошення класу і визначення методів класу. Розташування програмного коду у файлах.
20. Діапазон видимості класу. Оголошення в класі. Доступ до членів класу.
21. Статичні поля класу. Оголошення, ініціалізація, використання.
22. Статичні методи. Оголошення, визначення, використання.
23. Навіщо перевантажувати оператори? Синтаксис прототипу методу, що перевантажує оператор. Як компілятор опрацьовує вираз з оператором?
24. Вкажіть три головні способи перевантаження оператора.
25. Що таке дружня конструкція? Наведіть приклад використання.
26. У яких випадках оператор (для роботи з класом) можна перевантажити тільки дружньою функцією? Наведіть короткий приклад.
27. У яких випадках оператор (для роботи з класом) можна перевантажити зовнішньою функцією (без статусу дружньої)? Наведіть короткий приклад.
28. Перетворення типів: «клас → вбудований тип» і «вбудований тип → клас».
29. Відкрите наслідування: синтаксис, призначення, видимість, що моделює.
30. Закрите наслідування: синтаксис, призначення, видимість, що моделює.
31. Захищене наслідування: синтаксис, призначення, видимість, що моделює.
32. Множинне наслідування: синтаксис, призначення, безпроблемне використання.
33. Множинне Наслідування зі споріднених класів, віртуальні батьківські класи.
34. Включення об'єкта (об'єктів) до складу іншого об'єкта: синтаксис, можливості моделювання, використання.
35. Синтаксис конструктора підкласу. Порядок виконання його коду.
36. Призначення абстрактного класу. Синтаксис абстрактного методу.
37. Призначення і синтаксис віртуального методу. Порядок виклику віртуального методу.
38. Сумісність за присвоєнням «надклас-підклас», «надклас*-підклас*», «надклас&-підклас&».
39. Перевірка та динамічне перетворення типу об'єкта.
40. Відмінності у синтаксисі та можливостях використання включення і закритого наслідування.
41. У яких випадках використовують шаблони функцій (у C++)? Наведіть приклад оголошення.
42. Явне та неявне створення екземплярів шаблону функції (у C++).
43. Явна спеціалізація шаблону функції (у C++).
44. Що таке шаблон класу (у C++)? У яких випадках його використовують?
45. Оголошення та використання шаблону класу з параметром типом.
46. Оголошення та використання шаблону класу з параметром «не типом».

47. У яких випадках компілятор за шаблоном класу генерує визначення класу? Наведіть приклади.
48. Використання шаблона класу в якості аргумента шаблона, рекурсивне використання.
49. Спеціалізації шаблона класу (у C++).
50. Використання шаблона класу в якості базового (у C++). Наведіть приклади.
51. Використання шаблона класу в якості компоненти (у C++). наведіть приклади.
52. Як узагальнюють алгоритми опрацювання контейнерів (у C++)?
53. Використання класів характеристик, політик для оголошення шаблона класу (у C++).
54. Головні (чотири) групи шаблонів, що складають бібліотеку STL.
55. Що таке ітератор? Для чого його використовують? Де оголошують? Класифікація ітераторів.
56. Що таке функтор? Для чого його використовують? Класифікація функторів.
57. Що таке об'єкт-функція? Наведіть приклад оголошення і використання.
58. Охарактеризуйте стандартні функтори.
59. Алгоритми заповнення інтервалів і генерування значень, алгоритми підрахунку.
60. Алгоритми копіювання та перестановок.
61. Алгоритми пошуку та заміни.
62. Алгоритми порівняння та вилучення.
63. Алгоритми впорядкування та споріднені дії.
64. Застосування дії до кожного елемента контейнера.
65. Спільні риси послідовних контейнерів.
66. Ітератори вставляння для послідовних контейнерів.
67. Поточкові ітератори (оголошення, приклади використання).
68. Контейнер вектор (оголошення, функціональні можливості, приклади використання).
69. Контейнер список (оголошення, функціональні можливості, приклади використання).
70. Контейнер дек (оголошення, функціональні можливості, приклади використання).
71. Контейнер однозв'язний список (оголошення, можливості, приклади використання).
72. Контейнер стек (оголошення, можливості, приклади використання, доступ до реалізації).
73. Контейнер черга (оголошення, можливості, приклади використання, доступ до реалізації).
74. Контейнер пріоритетна черга (оголошення, можливості, приклади використання, доступ до реалізації, роль предиката в оголошенні та функціонування).
75. Спільні риси асоціативних контейнерів.
76. Контейнери множина, мультимножина (оголошення, можливості, приклади використання, роль предиката в оголошенні та функціонування).
77. Контейнери відображення, мультивідображення (оголошення, можливості, приклади використання, роль предиката в оголошенні та функціонування, особливості ітерування).
78. Клас string (влаштування, споріднені шаблони, можливості, приклади використання).
79. Числові масиви (оголошення, можливості, приклади використання).
80. Клас complex (оголошення, можливості, приклади використання).

Вміння писати програми:

1. Оголосити клас
 - a. правильно використати режими видимості;
 - b. правильно розташувати код у файлах;
 - c. уміти оголосити конструктори (різні), деструктор, розуміти, коли вони працюють;
 - d. уміти перевантажувати оператори (присвоєння, порівняння, арифметичні, введення/виведення) за допомогою методів, зовнішніх функцій, дружніх функцій;
 - e. вміти створювати статичні та динамічні екземпляри класу, надсилати їм повідомлення;
 - f. вміти створювати масиви об'єктів, завантажувати їх з файла, зберігати до файла.
2. Оголосити підклас (підкласи) за допомогою відкритого наслідування
 - a. правильно оголошувати конструктори підкласу (з викликом конструкторів базового класу), деструктор;
 - b. перевизначати методи (віртуальні), використовувати наявні, визначати нові;
 - c. розуміти особливості доступу до членів базового класу з різною видимістю;
 - d. перевизначати оператори для ієрархії класів, зокрема, оператори введення і виведення;

- e. розуміти та вміти використовувати розширені правила сумісності споріднених типів для передавання у функції (методи), для створення поліморфних колекцій;
 - f. розуміти як працюють та вміти використовувати поліморфні повідомлення;
 - g. розуміти призначення і оголошення абстрактних класів, вміти їх використовувати для побудови ієрархії типів.
3. Використовувати композицію та/або агрегацію об'єктів
- a. вміти включати об'єкт(и) до складу іншого;
 - b. використовувати закрите та захищене наслідування для реалізації включення;
 - c. проектувати та використовувати власні контейнери.
4. Використовувати узагальнені функції
- a. оголошення шаблону функції, використання, коли та як відбувається створення екземпляра функції;
 - b. спеціалізації (повні та часткові шаблони функції) для певного типу;
 - c. вміти оголосити та використати шаблон функції, оголосити його явну спеціалізацію.
5. Узагальнені типи
- a. оголосити та використати шаблон класу, параметризований типом та/або параметром-нетипом; наслідувати шаблони;
 - b. вміти застосовувати явну спеціалізацію шаблонів;
 - c. розуміти, коли та як відбувається створення оголошення класу за його шаблоном;
 - d. використовувати характеристики класів для створення шаблонів класів.
6. Бібліотека STL
- a. розуміти, для яких задач які типи контейнерів підходять найкраще; вміти підібрати контейнер за його властивостями;
 - b. вміти використовувати послідовні контейнери для зберігання колекцій значень, вміти їх перебирати, знаходити потрібне, змінювати, додавати/вилучати елементи, з'ясовувати загальні властивості контейнера;
 - c. вміти використовувати асоціативні контейнери для зберігання колекцій унікальних значень, відображень, вміти їх перебирати, знаходити потрібне, змінювати, додавати/вилучати елементи, з'ясовувати загальні властивості контейнера;
 - d. вміти виконувати перетворення контейнерів одного типу до іншого, завантажувати їх з файла, зберігати до файла;
 - e. вміти використовувати стек, чергу, пріоритетну чергу за безпосереднім призначенням;
 - f. вміти опрацьовувати текстову інформацію за допомогою рядкових контейнерів;
 - g. вміти використовувати стандартні алгоритми та ітератори для опрацювання контейнерів, взаємодії з потоками тощо (копіювання, трансформації, впорядкування, дії з множинами, пошук, генерування та заповнення інтервалів, заміни, перестановки, вилучення, злиття, числові алгоритми; ітератори контейнерів, потокові, вставляння);
 - h. вміти оголошувати об'єкт-функції та лямбда-вирази і використовувати їх для налаштування алгоритмів на необхідні перетворення, чи умови, використовувати для оголошення контейнерів тощо;
 - i. вміти використовувати тип комплексного числа та числові масиви для обчислень.