

Як засобами UML представити клас?

Клас можна представити в UML діаграмі класів у вигляді прямокутника з назвою класу всередині.

Які зв'язки між класами пропонує UML і як вони реалізуються у C++?

UML пропонує наступні зв'язки між класами: композиція, агрегація, асоціація, залежність, спадкування. У C++ ці зв'язки реалізуються за допомогою відповідних механізмів мови програмування.

У чому полягає концепція успадкування?

Успадкування - це механізм ООП, який дозволяє створювати нові класи на основі вже існуючих класів, успадковуючи їх властивості і методи.

Чим відрізняється відкрите, захищене і закрите успадкування?

Відкрите успадкування (public inheritance) дозволяє доступ до публічних членів базового класу з публічних і захищених методів похідного класу.

Захищене успадкування (protected inheritance) дозволяє доступ до публічних і захищених членів базового класу з захищених методів похідного класу.

Закрите успадкування (private inheritance) дозволяє доступ до публічних і захищених членів базового класу з закритих методів похідного класу.

Які переваги й недоліки множинного успадкування?

Переваги множинного успадкування: більша повторна використаність коду, можливість використовувати властивості декількох базових класів у похідному класі.

Недоліки множинного успадкування: можливість конфліктів імен, виникнення складності при розробці і підтримці програмного забезпечення.

Для чого визначають віртуальний базовий клас?

Віртуальний базовий клас використовується для уникнення проблеми подвійного копіювання при успадкуванні. Якщо декілька класів успадковують від одного базового класу, а сам базовий клас успадковується декількома класами, то може виникнути проблема дублювання базового класу в пам'яті. Віртуальний базовий клас дозволяє уникнути цієї проблеми шляхом забезпечення того, що лише одна копія базового класу буде збережена в пам'яті.

Для чого створюють ієрархії винятків?

Ієрархії винятків використовуються для обробки винятків у програмах. Вони дозволяють групувати винятки за типом і обробляти їх відповідно до типу. Це дозволяє зробити програму більш структурованою і зрозумілою.

У чому полягає концепція поліморфізму?

Поліморфізм - це здатність об'єкта вести себе по-різному залежно від контексту. В C++ це може бути досягнуто за допомогою віртуальних функцій та успадкування.

Чим відрізняється поліморфізм часу компіляції від поліморфізму часу виконання?

Поліморфізм часу компіляції визначається на етапі компіляції програми, коли виконується вибір функції на основі типу даних, які вона отримує. У поліморфізмі часу виконання вибір функції відбувається відповідно до реального типу об'єкта на етапі виконання програми.

Що таке віртуальна функція?

Віртуальна функція - це функція в базовому класі, яка може бути перевизначена в похідних класах. Вона вказує компілятору, що виклик методу повинен бути здійснений відповідно до типу об'єкта, на який посилається вказівник чи посилання, а не відповідно до типу вказівника чи посилання. Це дозволяє досягти поліморфізму, коли виклик методу здійснюється залежно від типу об'єкта, на який посилається вказівник чи посилання, а не від типу вказівника чи посилання. Для визначення віртуальної функції використовується ключове слово `virtual`.

Які класи вважаються поліморфними?

Поліморфними вважаються класи, що мають щонайменше один віртуальний метод.

Чому в поліморфних класах доцільно визначати віртуальні деструктори?

У поліморфних класах доцільно визначати віртуальні деструктори, щоб при видаленні об'єкта базового класу через вказівник на нього, який вказує на похідний клас, викликалась правильна реалізація деструктора похідного класу.

Що таке таблиця віртуальних методів?

Таблиця віртуальних методів - це таблиця, яка містить адреси віртуальних функцій класу, в якому вони були визначені. Кожен об'єкт, створений з похідного класу, має покажчик на цю таблицю.

Що таке суто віртуальна функція?

Суто віртуальна функція - це віртуальна функція без визначення в базовому класі. Це дає можливість відкласти визначення функції до моменту, коли вона буде потрібна в похідному класі.

Що таке абстрактний клас?

Абстрактний клас - це клас, який має щонайменше один чисто віртуальний метод. Чисто віртуальний метод - це метод, який не має реалізації в базовому класі і призначений для перевизначення в похідному класі.

Для чого використовують шаблонні функції?

Шаблонні функції використовуються для написання функцій, які можуть працювати з різними типами даних без необхідності створення окремої функції для кожного типу.

Для чого використовують шаблонні функції?

Шаблонні функції використовують для створення функцій, які можуть працювати з різними типами даних.

Чи можна шаблону функцію окремо реалізувати для певного типу?

Так, можна окремо реалізувати шаблону функцію для певного типу за допомогою явно вказання типу викликуваний шаблон.

Навіщо параметр шаблону в шаблонній функції вказувати явно?

Параметр шаблону в шаблонній функції можна вказувати явно, щоб забезпечити більш точне визначення типу аргументу, коли тип залежить від параметра шаблону.

Як створити шаблон класу?

Шаблон класу створюють за допомогою ключового слова `template` і указівки параметрів шаблону, які вказуються в `<>`. Приклад: `template <typename T> class MyClass {}`.

Якими можуть бути параметри шаблону?

Параметрами шаблону можуть бути будь-які типи даних, включаючи інші класи та шаблони.

Для чого використовують цілі параметри шаблону?

Цілі параметри шаблону використовуються для передачі значень констант часу компіляції в шаблон. Це може бути корисним для збереження пам'яті та оптимізації виконання програми.

Що таке інстанціювання шаблону?

Інстанціювання шаблону - це процес генерації конкретного класу або функції на основі шаблону з відповідними аргументами шаблону

Як здійснюється перетворення об'єктів інстанційованих типів?

Перетворення об'єктів інстанційованих типів здійснюється за допомогою явного або неявного приведення типу. Якщо тип не співпадає, то може виникнути виключення.