Основні поняття ООП. Історія мови С++

ООП це є методологія я ка базується на представленні програми. У вигляді сукупності об’єктів. Об’єкти є екземплярами класу, клас є новий тип даних.

ООП використовує конструкції об’єктів, а не алгоритми, як мова С. Класи можуть утворювати ієрархію на принципах наслідування (успадкування). Розрізняють 5 основних стилів програмування:

1. Процедурно-орієнтовний (алгоритми, мова С);
2. Об’єктно-орієнтований (класи, С++, С#);
3. Логічно-орієнований (предикати);
4. Орієнтований на правилах (якщо-> то);
5. Орієнтований на обмеженнях (СШІ);

Для ООП властиві 4 основні та 3 додаткові елементи:

1. Основні:

* Абстрагування – виділення таких суттєвих характеристик об’єктів, які відділяють його від інших для подальшого розгляду та аналізу. (Кімната і автомобіль)
* Обмеження доступу (інкапсуляція) – процес захисту окремих елементів об’єкту, що не стосується основних характеристик.
* Модульність – декомпозиція або розбиття системи на окремі модулі, що частково зменшує складність програми.
* Ієрархія на основі успадкування.

1. Додаткові:

* Типізація – це обмеження, що накладаються на клас об’єктів та запобігає взаємозаміні їх
* Паралелізм – Можливість одночасного існування та функціонування множини об’єктів. Надає об’єктам властивість знаходитись в активному або пасивному стані.
* Стійкість – властивість об’єкту існувати в часі та просторі незалежно від процесу, що його породив. Переміщення об’єкту з адресного простору.

Лекція №2

Переваги ООП:

1. Використання виразних засобів (об’єктно-орієнтованих програмних мов).
2. Можливість повторного використання окремих складових ПЗ (класи).
3. Створення більш відкритих систем.
4. Зниження ризиків при розробці.
5. Активізація пізнавальних здібностей програміста.

Недоліки ООП:

1. Вплив на характеристики системи (швидкодія та пам’ять).
2. Початкові затрати на навчання.

Основні принципи ООП:

1. Інкапсуляція (public, private, protected)
2. Наслідування (Базовий –> Похідний –> Похідний похідного)
3. Поліморфізм (Розробка одноіменних функцій з різною реалізацією)

Історія мови С++

Simula67, BCPL

|

C

|

C++

З мови Simula67 взято поняття класу. С++ називали “C з класами”. За замовчування всі данні класу є private, а данні структури – public. Вперше мова С++ була згадана у 1983 році. Розробник – Ріккі Массіті.

Мета С – позбавити необхідність програмування на Assembly.

Мова С – гнучка, компактна та має відносно низький рівень, С підходить для програмування більшості системних задач. Працює “всюди і на всьому”. Добре узгоджена з середовищем UNIX.

Міжнародний стандарт мови С++ був створений акредитованим комітетом стандартів, що діє під керівництвом ANSI (American National Standart Institute)

Стандарти мови С++ починаються від року її створення. Останній стандарт – С++ 20. Затверджений 20-го вересня 2020 року. Він вводить додатково поняття концепції співпрограми, новий оператор порівняння.

Відмінності С++ від С:

1. Підтримка ООП через класи.
2. Підтримка узагальненого програмування через шаблони.
3. Доповнення до стандартної бібліотеки (контейнери)
4. Обробка виняткових ситуацій.
5. Простір імен.
6. Вбудовані функції.
7. Перевантаження операцій.
8. Перевантаження імен функцій.
9. Посилання
10. Оператори управління динамічною пам’яттю.
11. Коментарі “//” у мові С++ з’явилися з мови BCPL, а пізніше коментарі вигляду “/\*\*/” з’явилися у с++

Несумісності С та С++. Більшість конструкцій в мові С дозволена в С++ і їх значення не змінюється. Несумісними є:

1. Заборонено використання в якості ідентифікаторів наступні ключові слова:

class, const, new, delete, friend, inline, operator, overload, public, private, protected, this, virtual.

1. F() – таке оголошення функції, як члена класу, дозволяє працювати всередині функції зі змінними того ж класу, без їх спеціальної передачі у функцію.
2. В мові С зовнішнє ім’я можна оголошувати декілька разів, в С++ тільки один раз.
3. Імена класів займають той самий адресний простір, що і інші імена, тому давати однакові назви класам змінним та функціям заборонено.

Підготовка до програмування

Мова С++ потребує попереднього проектування програми на етапі, що передує кодуванню.

Аналіз прогблеми -> моделювання –> написання коду

Переваги С++

1. Стандартизованість
2. Швидкодія
3. Ефективність при використанні пам’яті, часу, енергії
4. Масштабованість (програми розробляються для різних систем і платформ)
5. Можливість роботи на низькому рівні з пам’яттю, адресами, портами.
6. Можливість створювати узагальнені алгоритми для різних типів даних.
7. Підтримка різних стилів програмування та технологій. Використання мета-програмування на основі шаблонів та макросів.

Недоліки С++

1. Наявність безлічі можливостей призводить до порушення принципів безпеки.
2. Погано підтримується модульність.
3. Підключення через директиву “include” уповільнює виконання програми.
4. Недостатня інформація про типи даних при компілюванні програми.
5. Неінтуїтивні деякі перетворення типів. (операція над знаковим і беззнаковим елементом дає беззнаковий результат)
6. Препроцесор С++ є примітивного рівня, він успадкований від мови С і в деяких випадках перешкоджає реалізації мета-програмування
7. Хоча С++ де-юре вважається мультипарадигменою мовою, насправді відсутня підтримка функціонального програмування.

Об’єктно-орієнтований аналіз

Об’єкти реального світу.

1. Тип змінної визначає її розмір в пам’яті. Тип інформації, яку вона може зберігати та операції, які можуть над нею виконуватися.
2. Об’єкти мають свою поведінку і це визначає, як об’єкт впливає або підлягає впливу у стосунках з іншими об’єктами. Ця поведінка буде відображена через функції
3. На основі властивостей і поведінки буде створений новий тип даних. Він має всі функційні можливості та права вбудованих типів
4. В поведінці об’єкту виділяється поняття операція. Розрізняють 5 основних операцій над об’єктами:

* Модифікація (модифікатор) – змінює стан об’єкту шляхом доступу або запису.
* Селектор – надає доступ для визначення стану об1єкту без його зміни.
* Літератор – дає доступ до вмістимого об’єкту по частинах
* Конструктор – створює об’єкт та за потребами проводить ініціалізацію.
* Деструктор – знищує об’єкт та звільняє пам’ять, яку він займав.

1. За поведінкою об’єкти бувають активні та пасивні.

* Активні об’єкти не підлягають впливу інших об’єктів та розміщуються в каналі управління. Якщо система має декілька каналів управління, то одночасно може бути декілька активних об’єктів.
* Пасивні об’єкти змінюють свій стан тільки під впливом активних об’єктів

1. Також вирізняють таке поняття як порівняння об’єктів
2. Індивідуальність об’єкта – властивість, що вирізняє його від інших об’єктів (наприклад, імя)

Відношення між об’єктами

1. Відношення використання

Кожен об’єкт може виконувати одну з трьох ролей:

* Вплив - об’єкт впливає на інші, але на нього ніщо не впливає
* Виконання – на об’єкт впливають інші об’єкти, але він ні на кого не впливає
* Посередництво – об’єкт як сам може впливати, так і на нього можуть впливати.

При передачі повідомлень між об’єктами вони повинні синхронізуватись.

В послідовних системах ця синхронізація вирішується через підпрограми

В паралельних системах ситуація складніша – мають бути враховані виняткові ситуації. Звідси має бути врахована наступна класифікація об’єктів:

* Об’єкт-транслятор – пасивний об’єкт з одним каналом управління.
* Блокований об’єкт – пасивний об’єкт з декількома каналами управління.
* Паралельний об’єкт – активний об’єкт з декількома каналами управління.

1. Відношення включення

Перевага відношення включення у зменшенні кількості на певному рівні опису

Перевага відношення використання у незалежності

Клас!

Клас – це новий тип даних, множина об’єктів, які пов’язані спільністю структури та поведінки.

При формуванні класу визначають інтерфейсу частину. Вона може бути поділена на три складові:

1. Загальнодоступна. В ній визначаються елементи, видимі для всіх об’єктів інших класів.
2. Захищена. Визначаються елементи, видимі для об’єктів цього класу, та об’єктів, які успадковують даний клас.
3. Відособлена частина. Визначаються об’єкти, приховані для об’єктів інших класів.

Клас складається з набору властивостей (даних) та їх обробки (методів). Членами класу можуть бути об’єкти інших класів. Між класами доступні такі відносини:

1. Успадкування. На основі існуючого типу утворюється новий тип даних. Основними елементами успадкування є:

* Поліморфізм. Це є елемент теорії типізації, який дозволяє використовувати однакове ім’я для об’єктів різних класів. При цьому ці класи мають мати спільний базовий клас.
* Множинний поліморфізм
* Множинне наслідування

1. Відношення використання:

* Об’єкт одного класу є елементом іншого класу. При цьому об’єкт існує в області видимості цього іншого класу.
* Па

1. Відношення наповнення – побудова збірного класу.
2. Метаклас – побудова класу класів, де класи трактуються як об’єкти.

Екземпляром кожного класу є об’єкт за аналогією змінної певного типу.

Переважно класи статичні, всі їх особливості та змісти визначені в процесі компіляції.

Об’єкти навпаки – динамічно створюються та знищуються в процесі виконання програми, при цьому можуть займати як статичну так і динамічну пам’ять. Для оцінки класів і об’єктів виділяють 5 критеріїв:

1. Взаємозалежність – визначає ступінь глибини зв’язків між окремими модулями.
2. Зв’язнісь – Визначає ступінь функціональної взаємодії між елементами окремого модуля.
3. Достатність – наявність в класі всього необхідного для логічної та ефективної поведінки.
4. Повнота – наявність в інтерфейсній частині класу всього необхідного для взаємодії з користувачем
5. Простота – відсутність надлишковості.

Оголошення класу

class Cat {

char name[];

int age;

void Meow();

};

При оголошенні класу пам’ять не резервується.

При оголошенні повідомляє компілятору про оголошення класу. Розмір визначається в залежності від розміру елементів класу. Для методів класу пам’ять не резервується

Новоутворений тип даних має на рівні з вбудованими типами. Нові типи використовуюються для створення нових об’єктів.