



Вступ у програмування мовою С

Лектор:

к.т.н., доцент кафедри прикладної математики

Рижа Ірина Андріївна

Про що ця лекція

- Розглянемо еволюцію мов програмування.
- Наведемо характеристики мови С, її переваги та недоліки.
- Опишемо структуру С (С++) програми.

Еволюція мов програмування

- Пристрої сучасних ЕОМ засновані на принципах двійкової арифметики, де для подання чисел використовуються всього дві цифри - 0 і 1.
- Найбільш природним способом керування ЕОМ є кодування інформації для ЕОМ у вигляді бітових послідовностей.

Керування першими ЕОМ

- Алфавіт мови безпосереднього кодування містив усього дві цифри.
- Програмування в машинних кодах вимагало досконалого знання системи команд машини і великої уваги.
- Крім того, процес програмування в кодах малоефективний.

Еволюція мов програмування

Кодування і стає програмуванням

- З метою оптимізації програмування в двійкових кодах було розроблено спеціальну систему кодування двійкових машинних команд багатобуквенними мнемонічними скороченнями.
- Програмування в мнемонічних командах є зручнішим для програміста, оскільки містять додаткову інформацію для програміста в порівнянні з важко помітними послідовностями нулів та одиниць.
- Текст такої програми стає абсолютно незрозумілим обчислювальній машині і вимагає спеціальної програми - перекладача (**транслятора**), яка б замінювала мнемонічний код вихідної команди двійковою командою.

Мови програмування високого рівня

-- мови, які вимагають попереднього перекладу.

- Вважається, що ці мови в певному сенсі більш близькі до природних мов.

Еволюція мов програмування

Підпрограми

- це послідовності команд, призначені для багаторазового використання в одній програмі.
- реалізує можливість побудови великої програми з окремих фрагментів програмного коду.
- вимагає спеціальної програми - **компонувальника**, яка забезпечує формування єдиної програми з окремих фрагментів-підпрограм і її розміщення в пам'яті ЕОМ.

Транслятор + компоновальник = компілятор.

Процес трансляції

Програма - це послідовність інструкцій, призначених для виконання комп'ютером.

- Файл з вхідним текстом програми обробляється транслятором, який здійснює переклад програми з мови програмування у зрозумілу машині послідовність кодів.

1. Лексичний аналіз

- Програма поділяється на пропозиції, кожна з яких ділиться на елементарні складові (лексеми).
- Кожна лексема розпізнається (ім'я, ключове слово, літерал, символ операції або розділювач) і перетворюється у відповідне двійкове представлення.

2. Синтаксичний аналіз

- На цьому етапі з лексем збираються вирази, а з виразів - оператори.

3. Поетапна генерація коду

- Відбувається заміна операторів мови високого рівня послідовностями машинних команд.
- Результат перетворень вхідного тексту програми записується у вигляді двійкового файлу («.obj»).

Процес компонування

Компонувальник

- відповідає за те, щоб конкретному операнду виразу відповідала певна область пам'яті;
- узгоджує адреси у всіх фрагментах коду, з яких збирається готова до виконання програма;
- додає коди так званих бібліотечних функцій (вони забезпечують виконання конкретних дій - обчислення, виведення інформації на екран дисплея і т.д.), а також код, що забезпечує розміщення програми в пам'яті, її коректний початок і завершення.

Файли, що містять перетворену компонувальником програму називаються **завантажувальними або виконавчими** файлами.

Підходи до побудови програмних продуктів

1. Принцип структурного програмування (представлений мовою C)

полягає у покращенні читабельності програм та за рахунок використання операторів розгалуження.

- Цей принцип гарно поєднується з принципом проектування алгоритму «зверху-вниз», коли задача розбивається на логічно пов'язану послідовність елементарних (більш простих) задач.

2. Об'єктно-орієнтований підхід (представлений мовою C++)

акцент робиться не на структуру програми, а на дані, дозволяє розробляти такі форми даних та операцій над ними (класи), які відповідають специфіці задачі.

- Класи при цьому описують необхідну форму даних, а конкретною структурою даних, створеною у відповідності до існуючого класу є об'єкт.

3. Узагальнене програмування (програмування на основі шаблонів)

- дає можливість використання раніше створених програм, які вже не залежать від типів даних.

Характеристики мови C

Cі (C)

- універсальна, процедурна, імперативна мова програмування загального призначення:
- була розроблена для написання системного програмного забезпечення (операційної системи) у 1972 році Денісом Рітчі та Кеном Томпсоном у стінах Bell Labs;
- наразі досить часто використовується для написання прикладного (застосовного) програмного забезпечення;
- вірогідно є найпопулярнішою мовою програмування за кількістю написаного на ній програмного забезпечення, доступного під вільними ліцензіями коду та числа програмістів, які її знають.

Переваги мови С

Конструктивні особливості

Конструкція мови С добре підходить для:

- низхідного планування (від загального до часткового);
- структурного програмування;
- модульного проектування.

Все це дозволяє отримувати **надійні і зрозумілі** програми.

Ефективність

- Конструкція С продуктивно використовує можливості комп'ютерів, на яких вона встановлена.
- Програми на С відрізняються компактністю і швидкістю виконання.

Переваги мови C

Переносимість

- Програму, написану на C для однієї системи, можна виконувати на іншій системі з мінімальними змінами;
- Компілятори мови C існують майже для усіх платформ.

Орієнтація на програмістів

- Мова C орієнтована на задоволення потреб програмістів.
- Вона надає доступ до обладнання та дозволяє маніпулювати окремими фрагментами пам'яті.
- Передбачає багатий вибір операцій, які дозволяють лаконічно висловлювати свій підхід до розв'язання завдань.

Недоліки мови C

Недоліки

- Оскільки C проектувалась для використання у системному програмуванні, то вона забезпечує простий, прямий доступ до адреси будь-якого об'єкту (наприклад, карти пам'яті, пристрою контролю регістрів).
- C дозволяє здійснення низки операцій, котрі часто не є бажаними, а тому чимало помилок у коді не виявляється компілятором, і не можуть бути очевидними під час виконання.
- Ефективне використання C вимагає від програміста більше навиків, досвіду та зусиль, ніж це потрібно для деяких інших мов програмування.

Програмне середовище MS Visual Studio

Microsoft Visual Studio

— низка продуктів фірми Microsoft, що включають інтегроване середовище розробки (IDE) програмного забезпечення і ряд інших інструментальних засобів.

Як встановити Microsoft Visual Studio???

1. Перейти на сайт <https://www.microsoft.com/>
2. Вибрати закладку:

Усі продукти Microsoft -> Visual Studio -> Download Visual Studio Community 2019

Структура C (C++) програми

Структура програми:

- директиви препроцесора;
- оголошення глобальних змінних і констант;
- оголошення агрегованих типів даних;
- оголошення функцій;
- означення функцій, що були оголошені або ні;
- тіло головної функції;
- означення функцій.

Функція `main()`

Головна функція `main()`

Проект може мати декілька файлів з розширенням “.cpp”, але обов’язково має бути один і тільки один файл, що містить функцію `main()`.

- Як і для кожної функції на мові C/C++, перед іменем цієї функції повинно стояти зарезервоване слово, яке відповідає за тип результату функції. Для головної функції це може бути одне з двох слів: `void` або `int`.
- Після заголовка функції у фігурних дужках записуються інструкції програми.
- Послідовність виконавчих інструкцій, записаних у фігурних дужках, називають **тілом функції**.
- За межами функцій інших інструкцій, ніж описових, бути не повинно.

Коментування програми

Коментарі

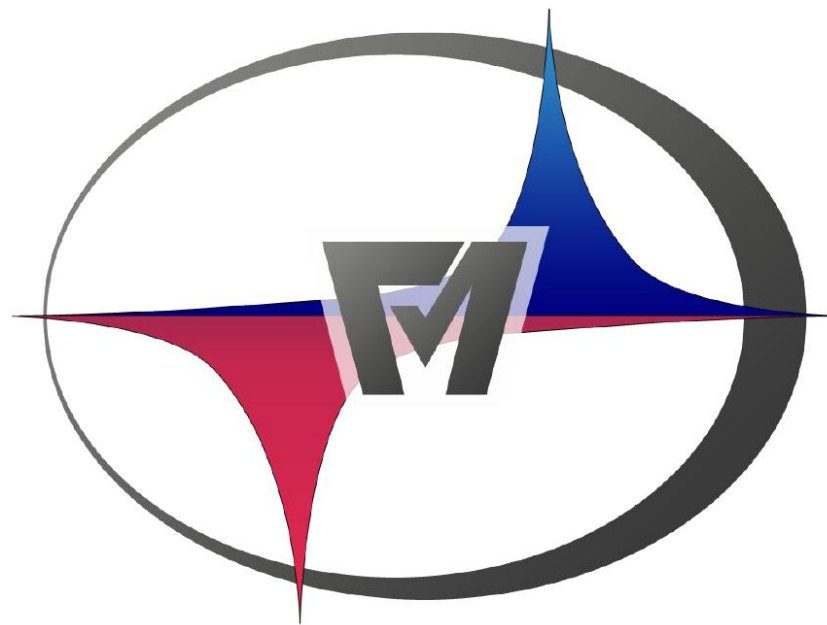
– це довільна послідовність символів, взятих у дужки `/*.....*/`, або таких, що слідує за символами `//` і обмежена кінцем стрічки.

- Коментарі допомагають краще розуміти текст програми користувачу, але ігноруються компілятором.
- Поганий коментар може бути гіршим, ніж ніякий.

Коментування програми

Вартує віддавати перевагу

1. Коментарю для кожного вхідного файлу, який інформує, що знаходяться в ньому, дає посилання на довідники і посібники, загальні рекомендації з використання і т.п.
2. Коментарям для кожної нетривіальної функції, у якому сформульоване її призначення, використовуваний алгоритм (якщо він неочевидний) і, можливо, щось про прийняті в ній припущеннях щодо середовища виконання;
3. Невелике число коментарів у тих місцях, де код програми не очевидний.



Кафедра прикладної математики

<http://amath.lp.edu.ua>