

Мови програмування.

Мова програмування – формальна знакова система, призначена для опису алгоритмів в формі, яка зручна для виконавця (наприклад, ЕОМ, тобто комп'ютера).

Для машин першого покоління програми створювалися в машинному коді (або на машинному мовою). Це означає, що кожна операція, яку могла виконати ЕОМ, мала код (свій номер). Програміст, формуючи директиву для комп'ютера, повинен був вказати код операції та місця розташування інформації, над якою треба було виконати дану операцію. Наприклад, директива додавання двох чисел могла виглядати так:

001 0345 0363 0211.

У представленому коді набори чисел означають наступне:

001 – операція додавання,

0345 – адреса першого числа,

0363 – адреса другого числа,

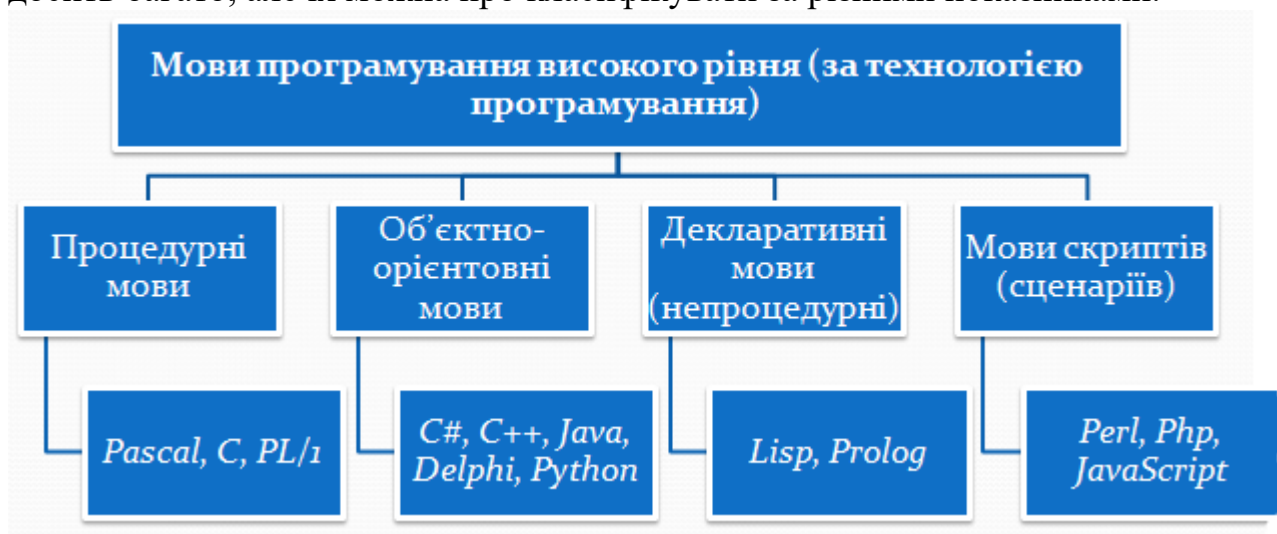
0211 – адреса для запису результату.

Вчені спробували змінити ситуацію:

1) спростити процес програмування,

2) зробити код програми читаним і інформативним при його розгляді.

З цією метою розроблялись мови програмування яких на сьогодні існує досить багато, але їх можна про класифікувати за різними показниками.



В загальному випадку мови програмування поділяють на:

- мови низького рівня;
- мови високого рівня.

До групи мов **низького рівня** входять машинні мови і мови символічного кодування: Автокод, Асемблер. Оператори цієї мови – це ті ж машинні команди, але записані мнемонічними кодами, а в якості операндів використовуються не конкретні адреси, а символічні імена. Всі мови низького рівня орієнтовані на певний тип комп'ютера, отже є машинно-залежними.

Машинний код		Асемблер	
0005	B4 09	7	<u>mov AH, 09h</u>
0007	BA0000r	8	<u>mov DX, offset msg</u>
00A	CD 21	9	<u>int 21 h</u>

Машинно-орієнтовані мови – це мови, набори операторів і образотворчі засоби яких істотно залежать від особливостей ЕОМ (внутрішнього мови, структури пам'яті і т.д.).

Машинно-орієнтовані мови дозволяють використовувати всі можливості та особливості машинно-залежних мов:

- висока якість створюваних програм (компактність і швидкість виконання);
- можливість використання конкретних апаратних ресурсів;
- передбачуваність об'єктного коду та замовлень пам'яті;
- для складання ефективних програм необхідно знати систему команд та особливості функціонування даної ЕОМ;
- трудомісткість процесу складання програм (особливо на машинних мовах і ЯСК), погано захищеного від появи помилок;
- низька швидкість програмування;
- неможливість безпосереднього використання програм, складених на цих мовах, на ЕОМ інших типів.

До мов програмування **високого рівня** відносять мови програмування, які дозволяють оперувати безпосередньо ідентифікаторами (іменами) змінних та параметрів та наближені до природної мови.

Ці мови машинно-незалежні, тому що вони орієнтовані не на систему команд тієї чи іншої ЕОМ, а на систему операндів, характерних для запису певного класу алгоритмів. Однак програми, написані на мовах високого рівня, займають більше пам'яті і повільніше виконуються, ніж програми на машинних мовах.

Серед мов високого рівня можна виділити Фортран (перекладач формул – був розроблений в середині 50-х років програмістами фірми ІВМ і в основному використовується для програм, що виконують природно – наукові та математичні розрахунки), Алгол, Кобол (комерційний мову – використовується, в першу чергу, для програмування економічних задач), Паскаль, Бейсік (був розроблений професорами Дармутського коледжу Джоном Кемені і Томасом Курц.), Сі (Денніс Рітч - 1972 році), Пролог (в основі мови лежить апарат математичної логіки) і т.д.

Мови програмування також можна класифікувати на процедурні і непроцедурного.

У **процедурних** мовах програма явно описує дії, які необхідно виконати, а результат задається тільки способом отримання його за

допомогою деякої процедури, яка представляє собою певну послідовність дій.

Серед процедурних мов виділяють в свою чергу структурні і операційні мови. У структурних мовах одним оператором записуються цілі алгоритмічні структури: розгалуження, цикли і т.д. В операційних мовах для цього використовуються кілька операцій. Широко поширені такі структурні мови: Паскаль, Сі, Ада, ПЛ / 1. Серед операційних відомі Фортран, Бейсік, Фокал.

Непроцедурне (декларативне) програмування з'явилося на початку 70-х років 20 століття. До непроцедурного програмування відносяться функціональні і логічні мови.

У **функціональних** мовах програма описує обчислення деякої функції. Зазвичай ця функція задається як композиція інших, більш простих, ті в свою чергу діляться на ще більш прості завдання і т.д. Один з основних елементів функціональних мов – рекурсія. Оператора привласнення і циклів в класичних функціональних мовах немає.

У **логічних** мовах програма взагалі не описує дій. Вона задає дані і співвідношення між ними. Після цього системі можна задавати питання. Машина перебирає відомі і задані в програмі дані і знаходить відповідь на питання. Порядок перебору не описується в програмі, а неявно задається самим мовою. Класичним мовою логічного програмування вважається Пролог. Програма на Пролозі містить, набір предикатів-тверджень, які утворюють проблемно-орієнтовану базу даних і правила, які мають вигляд умов.

До не процедурних мов відносяться так звані проблемно орієнтовані мови програмування, які спрямовані на певний тип задач, наприклад, мова середовища Matlab спрямована на задачі математичного моделювання.

Можна виділити ще один клас мов програмування – об'єктно-орієнтовані мови високого рівня. На таких мовах не описують докладної послідовності дій для вирішення завдання, хоча вони містять елементи процедурного програмування. Об'єктно-орієнтовані мови, завдяки багатому призначеному для користувача інтерфейсу, пропонують людині вирішити задачу в зручній для нього формі.

Перший об'єктно-орієнтована мова програмування Simula був створений в 1960-х роках Нігаардом і Далом.

Програму, написану на мові програмування високого рівня, ЕОМ не розуміє, оскільки їй доступний тільки машинний мову. Тому для перекладу програми з мови програмування на мову машинних кодів використовують спеціальні програми – транслятори.

Трансляція – це переклад. Якщо ми подивимося, як працюють перекладачі з іноземних мов, то теж побачимо, що існує два підходи: синхронний переклад і переклад текстів.

Синхронний перекладач «ловить» почуту фразу і переводить її якомога ближче до сенсу. У момент перекладу він не знає, яка фраза піде далі, і не встигає враховувати фрази, висловлені раніше. Він ніби в будь-який момент

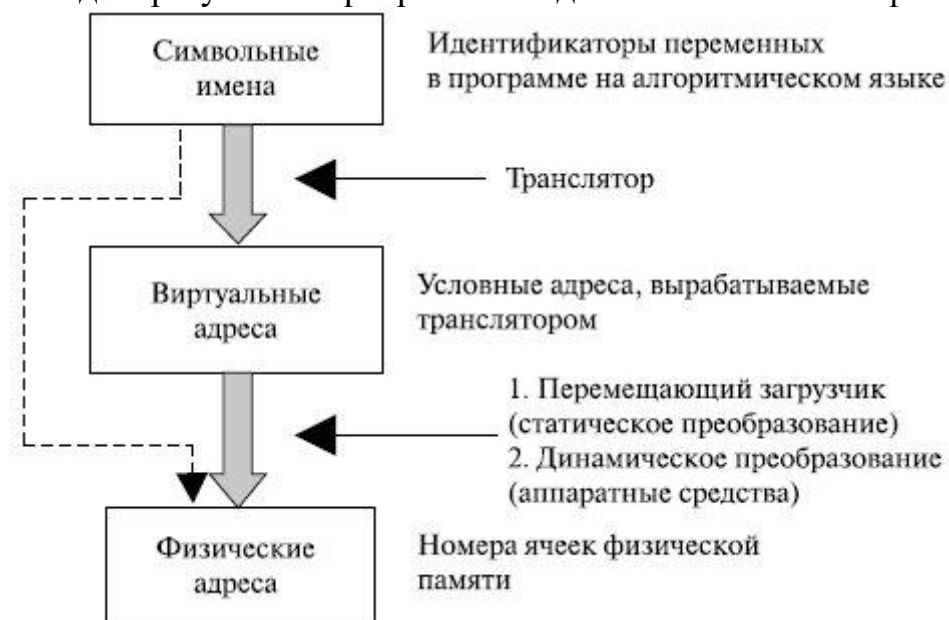
часу працює з однією конкретною фразою. Для перекладу будь-яких специфічних понять йому доводиться вибудовувати цілі конструкції. Фрази синхронного перекладу зазвичай довше, ніж їх оригінал, і не відрізняються витонченістю.

Літературний перекладач спочатку кілька разів прочитає весь текст, що підлягає перекладу. Він заздалегідь знайде в ньому все особливо складні для перекладу місця, побудує потрібні конструкції і буде ними користуватися щоразу, коли виникне необхідність. При перекладі однієї конкретної фрази перекладач повинен враховувати особливості всієї книги в цілому і навіть особливості інших творів того ж автора.

Існує три види транслятора: інтерпретатори, компілятори і асемблери (переводять програму, записану на мові асемблера, в програму на машинній мові).

Інтерпретатори працюють як синхронні перекладачі. Вони беруть один оператор з програми, транслують його в машинний код (або в якийсь проміжний код, близький до машинного коду) і виконують його. Якщо якийсь оператор багаторазово використовується в програмі, інтерпретатор щоразу буде сумлінно виконувати його переклад так, як ніби зустрів його вперше.

Компілятори обробляють програму в кілька прийомів. Спочатку вони кілька разів переглядають вихідний текст (зазвичай він називається вихідним кодом), знаходять спільні місця, Виконують перевірку на відсутність помилок синтаксису і внутрішніх протиріч, і лише потім переводять текст в машинний код. В результаті програма виходить компактною і ефективною.



С

Співробітник фірми Bell Labs Деніс Рітчі створив мову Сі в 1972 р під час спільної роботи з Кеном Томпсоном, як інструментальний засіб для реалізації операційної системи Unix. Мова С замислювався як мова проміжний між мовами високого і низького рівня. Від компілятора

намагалися домогтися продуктивності, близькою до продуктивності Ассемблера, але в той же час зберегти можливість перенесення програм між комп'ютерними платформами, характерну для мов високого рівня.

C – сучасна мова. Включає в себе ті керуючі конструкції, які рекомендовані теорією і практикою програмування.

C – ефективна мова. Її структура дозволяє найкращим чином використовувати можливості сучасних персональних комп'ютерів. Програмування на цій мові відрізняється компактністю і швидкістю виконання.

C – переносима і мобільна мова.

Cі – потужна і гнучка мова. Велика частина операційної системи Unix, компілятори і інтерпретатори мов Фортран, Паскаль і Бейсік написані саме з його допомогою.

C – зручна мова. Досить структурована, щоб підтримувати хороший стиль програмування і в той же час не пов'язана жорсткими обмеженнями.

Надзвичайну популярність мова C придбала завдяки тому, що, як і Паскаль, є мовою структурного програмування, але дозволяє генерувати більш продуктивний і компактний робочий код.

C ++.

Мова C ++ був розроблений в 1980 р Бьерном Страуструпом в компанії Bell Labs.

Очевидно, що дуже багато C++ запозичив у мови C. Ці запозичення забезпечили C++ потужними засобами низького рівня, що дозволяють вирішувати складні завдання системного програмування. Але C++ відрізняється від C в першу чергу різним ступенем уваги до типам і структурам даних. Це пов'язано з появою понять класу, похідного класу і віртуальної функції. Це дає C++ більш ефективні можливості для контролю типів даних і забезпечує модульність програми.

Мова C++ є засобом об'єктного програмування, новітньої методики проектування та реалізації програм, яка в поточному десятилітті, швидше за все, замінить традиційне процедурне програмування.

Останнім часом з'явилися середовища швидкого проектування на C++, які надають готові компоненти для введення і виведення інформації і підтримують концепцію віртуального проектування додатків Windows.

Для мови C ++ в даний час розроблені потужні інтегровані середовища (Builder C ++, Visual C ++), які підтримують концепції ООП.

При створенні C++ прагнули зберегти сумісність з мовою C. Більшість програм на C справно працюватимуть і з компілятором C++. C++ має синтаксис, заснований на синтаксисі C.

Нововведеннями C++ порівняно з C є:

1. **підтримка об'єктно-орієнтованого програмування через класи;**
2. **підтримка узагальненого програмування через шаблони;**
3. **доповнення до стандартної бібліотеки;**
4. **додаткові типи даних;**

5. **обробка винятків;**
6. **простори імен;**
7. **вбудовані функції;**
8. **перевантаження операторів;**
9. **перевантаження імен функцій;**
10. **посилання і оператори управління вільно розподіленою пам'яттю.**

У 1998 році ратифіковано міжнародний стандарт мови C++: ISO/IEC 14882 «Standard for the C++ Programming Language». Поточна версія цього стандарту — ISO/IEC 14882:2003.