Зміст

1 Мови програмування та мовні процесори

1.1 Мови програмування

При вивченні мов програмування, як правило, виділяють три аспекти:

- Прагматичний;
- Семантичний;
- Синтаксичний.

1.1.1 Прагматичний аспект

Прагматичний аспект (прагматика мови програмування) визначає клас задач, на розв'язування яких орієнтується мова програмування. Як правило, прагматичний аспект менш формалізований у порівнянні з семантичним та синтаксичним аспектами.

За класом задач на розв'язування яких орієнтуються мови програмування їх можна поділити передусім на

- процедурні;
- непроцедурні.

Процедурні мови програмування орієнтовані перш за все на опис (визначення) алгоритмів, тобто по суті використовуються для побудови процедур обробки даних. До таких мов ми відносимо всім відомі мови програмування, такі як Pascal, Fortran, C та ін.

Непроцедурні мови програмування на відміну від процедурних неявно визначають процедури обробки даних. Частіше всього такі мови використовуються для побудови завдань на обробку даних. При цьому, за допомогою інструкцій непроцедурної мови програмування визначається що необхідно зробити з даними і явно не визначається як (з використанням яких алгоритмів) необхідно розв'язати задачу. До непроцедурних мов програмування ми відносимо командні мови операційних систем, мови управління в пакетах прикладних програм та ін.

Як процедурні, так і непроцедурні мови програмування можуть орієнтуватися як на декілька класів задач, так і конкретну предметну область. У першому випадку ми будемо говорити про *універсальні* мови програмування (Pascal, Fortran, C), в другому — про *спеціалізовані* мови програмування (Snobol, Lisp).

1.1.2 Семантичний аспект

Семантичний аспект (семантика мови програмування) визначається шляхом конкретизації базових функцій обробки даних, набору конструкцій управління та методами побудови більш "складних" програм на основі "простих".

Наприклад, визначивши як базовий тип даних "рядок" ми повинні запропонувати "традиційний" набір функцій обробки таких даних: порівняння рядків, виділення частини рядка, конкатенацію рядків та ін.

Семантика мови програмування має бути визначена формально, бо інакше у подальшому неможливо буде побудувати відповідний мовний процесор. Станом на сьогодні існують два основних напрямки визначення семантики мов програмування:

- методи денотаційної семантики;
- методи операційної семантики.

Методи *денотаційної семантики* базуються на відповідних алгебрах, методи *операційної семантики* базуються на синтаксичних структурах програм.

1.1.3 Синтаксичний аспект

Синтаксичний аспект (синтаксис мови програмування) визначає набір синтаксичних конструкцій мови програмування, які використовуються

для нотації (запису) семантичних одиниць в програмі. Про синтаксис мови програмування можна сказати як про форму, яка є суть похідною від семантики. Для визначення (опису) синтаксису мови програмування використовуються як механізми, що орієнтовані на синтез, так і механізми, орієнтовані на аналіз.

Задачі аналізу та синтезу синтаксичних структур програм — це дуальні задачі. Їх конкретизацію ми будемо розглядати в наступних розділах.

Виходячи з вищенасказаного, щоб побудувати мову програмування потрібно:

- визначити клас (класи) задач, на розв'язок яких орієнтована мова програмування;
- виділити базові типи даних та функції їх обробки, вказати конструкції управління в програмах. Побудувати механізми конструювання більш складних програм та структур даних на основі більш простих одиниць;
- визначити синтаксис мови програмування.

1.2 Мовні процесори

Мовні процесори реалізують мови програмування. Точніше, мовний процесор призначений для обробки програм відповідної мови програмування. З точки зору прагматики, мовні процесори діляться на

- транслятори;
- інтерпретатори.

Мовний процесор типу транслятор (транслятор) — це програмний комплекс, котрий на вході отримує текст програми на вхідній мові, а на виході видає версію програми на вихідній мові, що називається об'єктною мовою. В більшості випадків як об'єктна мова виступає мова команд деякої обчислювальної машини. Серед трансляторів можна виділити дві програмні системи:

- компілятори — транслятори з мов програмування високого рівня;
- асемблери — транслятори машинно-орієнтованих мов програмування.

Мовний процесор типу інтерпретатор (інтерпретатор) — це програмний комплекс, котрий на вході отримує текст програми на вхідній мові та вхідні дані, які в подальшому обробляються програмою, а на виході видає результати обчислень (вихідні дані).

Оскільки транслятори та інтерпретатори реалізують мови програмування, вони мають спільні риси: їх структура досить схожа, в основу їх реалізації покладено спільні теоретичні результати та практичні методи реалізації.

1.2.1 Структура транслятора

1. Вхідний текст програми
2. Лексичний аналіз
3. Синтаксичний аналіз
4. Семантичний аналіз
5. Оптимізація проміжного коду
6. Генерація коду
7. Вихідний (об'єктний) код

1.2.2 Призначення основних компонентів транслятора

1. *Лексичний аналізатор.*

   Вхід: вхідний текст (послідовність літер) програми.

   Вихід: послідовність лексем програми.

   Лексема — це ланцюжок літер, що має певний зміст. Всі лексеми мови програмування (їх кількість, як правило, нескінчена) можна розбити на скінчену множину класів. Для більшості мов програмування актуальні наступні класи лексем:

   - зарезервовані слова;
   - ідентифікатори;
   - числові константи (цілі та дійсні числа);
   - літерні константи;
   - рядкові константи;

- коди операцій;
- коментарі. Безпосередньо не несуть інформації щодо структури програми. В подальшому не використовуються, тобто не передаються синтаксичному аналізатору.
- дужки та інші елементи програми.

2. *Синтаксичний аналізатор.*

   Вхід: послідовність лексем програми.

   Вихід:

   - "Так" + синтаксична структура (синтаксичний терм) програми,
   - "Ні" + синтаксичні помилки в програмі.

3. *Семантичний аналізатор.*

   Вхід: Синтаксичний терм програми.

   Вихід:

   - "Так" + семантична структура (семантичний терм) програми,
   - "Ні" + семантичні помилки в програмі.

4. *Оптимізація проміжного коду.*

   Вхід: семантичний терм програми.

   Вихід: оптимізований семантичний терм програми.

   Оптимізація — це еквівалентне перетворення програми на основі певних критеріїв. Серед критеріїв оптимізації можна виділити:

   - оптимізацію по пам'яті;
   - оптимізацію по швидкості виконання.

   В залежності від підходів по оптимізації програми можна розглядати такі методи оптимізації:

   - машинно-залежні;
   - машинно-незалежні.

На відміну від машинно-незалежних методів машинно-залежні методи оптимізації враховують архітектурні особливості ЕОМ, наприклад, наявність апаратного стека, наявність вільних регістрів, тощо.

5. *Генерація об'єктного коду.*

   Вхід: семантичний терм програми.

   Вихід: результуючий (об'єктний) код програми.

1.3 Контрольні запитання

1. Які три аспекти як правило виділяють при вивчення мов програмування?
2. Які два поділи мов програмування в залежності від орієнтації на розв'язання тих чи інших класів задач вам відомі?
3. Які традиційні функції обробки типу даних "рядок" вам відомі?
4. Які два класи задач пов'язаних з синтаксичними структурами програм вам відомі?
5. Які два типи мовних процесорів вам відомі?
6. Опишіть структуру транслятора.
7. Що таке лексема?
8. Які два поділи оптимізації ви знаєте?