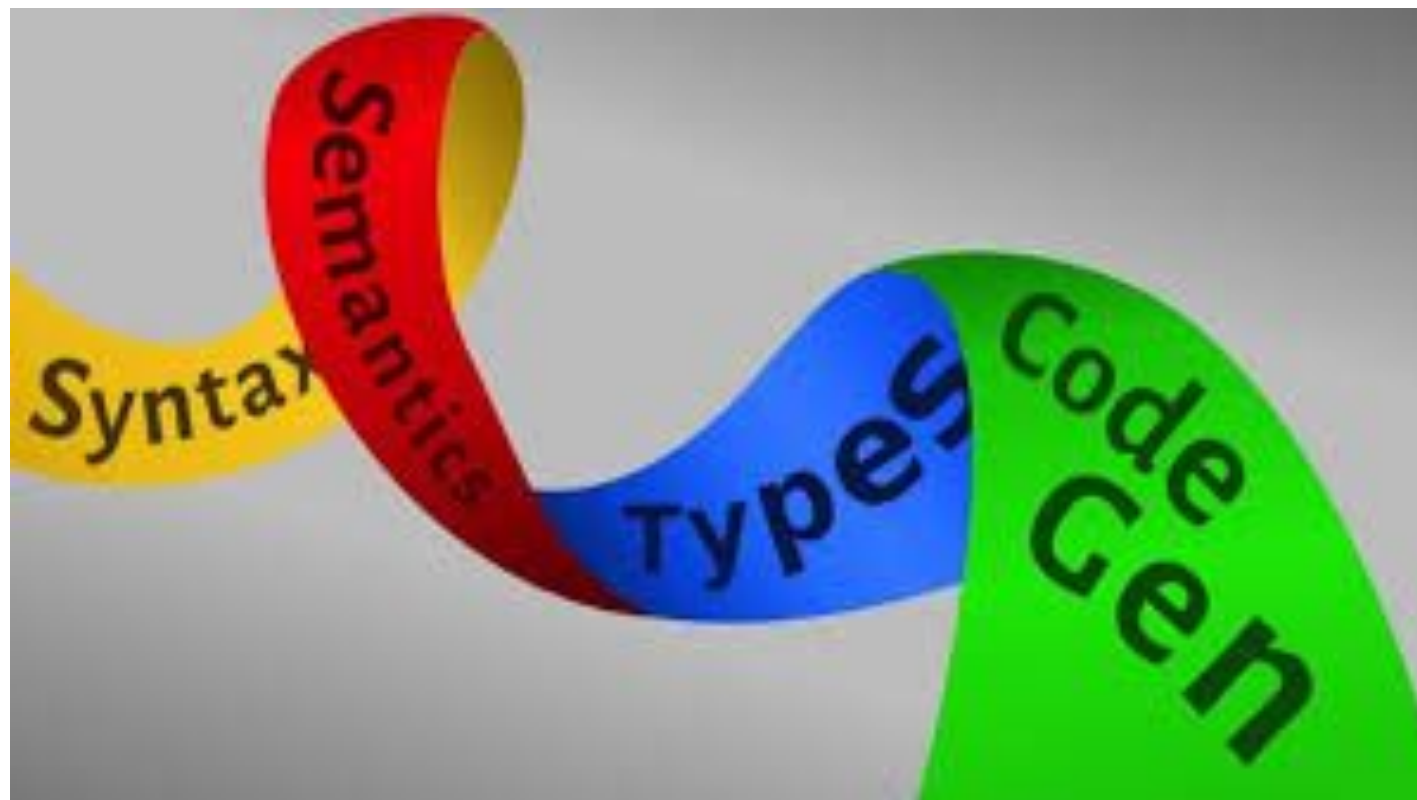


Огляд курсу «Програмування мовних процесорів»



Про що курс?



- Курс "Програмування мовних процесорів" фокусується на вивченні технологій **створення власних мов програмування**. Студенти отримають знання про процес лексичного та синтаксичного аналізу, семантичного аналізу та генерації коду.
- Курс включає:
 1. **Лексичний та синтаксичний аналіз**: Вивчення методів розпізнавання та обробки лексем та синтаксичних конструкцій у програмному коді (регулярні вирази, скінченні автомати, формальні граматики).
 2. **Семантичний аналіз**: Розгляд визначення значень та взаємозв'язків між елементами мови програмування.
 3. **Генерація коду**: Вивчення процесу перетворення вихідного коду програми в машинний код або код іншої мови програмування.

Фази побудови мовних процесорів

Лексичний аналіз (сканування):

Розпізнавання та виділення лексем зі вихідного тексту програми.

Синтаксичний аналіз:

Побудова синтаксичного дерева, перевірка відповідності коду граматиці мови програмування.

Семантичний аналіз:

Визначення семантичного значення програмного коду, виявлення семантичних помилок.

Генерація проміжного коду:

Створення проміжного коду або представлення вихідного коду внутрішньою формою для подальшої обробки.

Оптимізація коду:

Покращення ефективності та продуктивності програмного коду шляхом застосування різних оптимізацій.

Генерація вихідного коду (або інтерпретація):

Створення фінального вихідного коду або виконання програми на льоту для інтерпретаторів.



Основні завдання лексичного аналізу

1. **Розпізнавання лексем:** Визначення груп символів, які утворюють окремі лексеми, такі як ідентифікатори, числа, оператори і т. д.
2. **Виділення токенів:** Призначення кожній лексемі токена, який має асоційований тип та інформацію, необхідну для подальшого синтаксичного та семантичного аналізу.
3. **Вилучення зайвих символів:** Видалення з тексту програми непотрібних елементів, таких як пробіли, коментарі та інші символи, що не мають семантичного значення.



Основні завдання синтаксичного аналізу

1. Побудова синтаксичного дерева:

Синтаксичний аналізатор (парсер) перевіряє, чи послідовність токенів відповідає правилам граматики мови програмування, та будує синтаксичне дерево, яке представляє структуру програми.

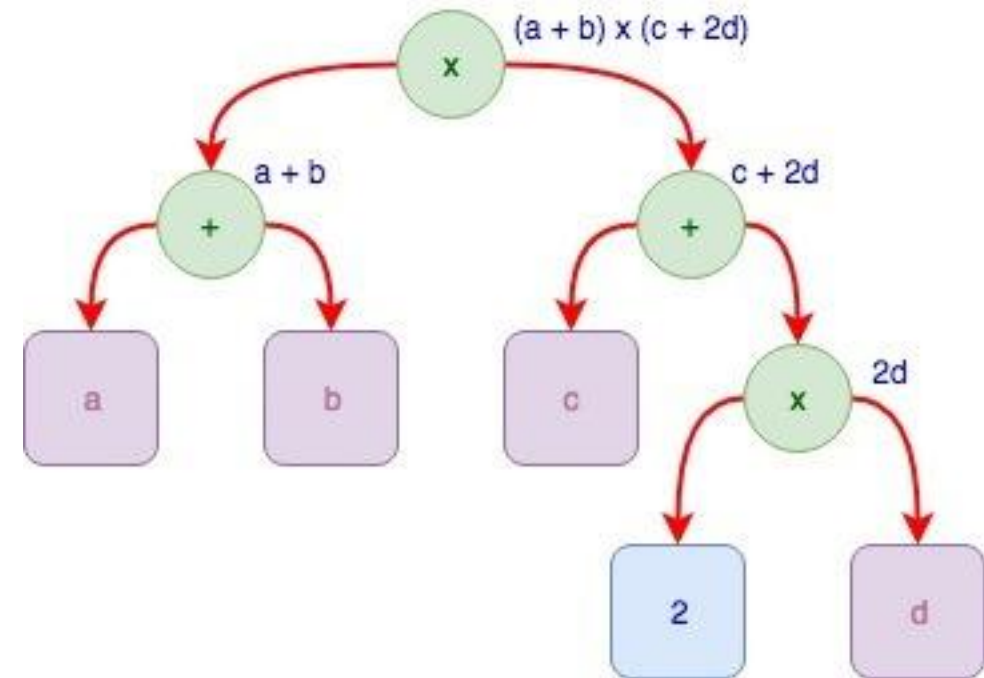
2. Перевірка правильності синтаксису:

Визначення та розпізнавання коректного синтаксису мови програмування.

Синтаксичний аналізатор виявляє синтаксичні помилки у вихідному кодї, якщо такі є.

3. Визначення порядку виконання операторів:

Визначення правильного порядку виконання операторів з урахуванням пріоритетів та асоціативності.



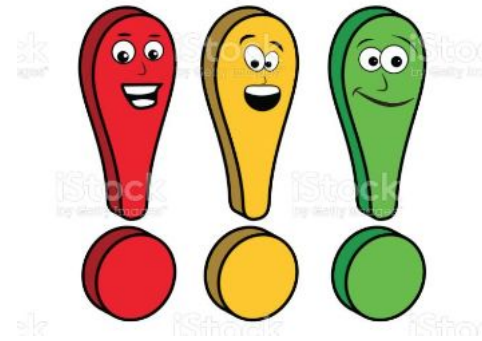
Генерація коду

Генерація коду — частина процесу [компіляції](#), коли спеціальна частина [компілятора](#), кодогенератор, конвертує синтаксично коректну програму в послідовність інструкцій, які можуть виконуватися на машині. При цьому можуть застосовуватися різні, в першу чергу машинно-залежні оптимізації

```
#include <stdio.h>
int main() {
    printf("Hello, Compilation!");
    return 0;
}
```

```
0100100101101110
0110001101100101
0010000011011100
1010110101110100
0111010001111001
0110001000101100
```

Вибирайте цей курс!!!



Ви завжди мріяли про власну мову програмування? Тепер у вас є шанс втілити свої ідеї у життя!

- Приєднуйтеся до нашого захоплюючого курсу з **створення власних мов програмування** та дізнайтеся, як зробити свій внесок у світ інновацій.
- Освоюйте найновіші технології компіляції та інтерпретації.
- Розширте свої кар'єрні можливості завдяки унікальним навичкам у розробці мовних процесорів.
- Розпочніть свою захоплюючу подорож у світ власної розробки мов програмування.

