

## Лабораторна робота №1

### Тема: Проектування лексичного аналізатора (сканера)

Мета роботи: Навчитися реалізовувати лексичний аналіз на основі регулярних виразів, кінцевих автоматів.

## 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Першим етапом аналізу програми мовою програмування є групування послідовності символів в одиниці мови.

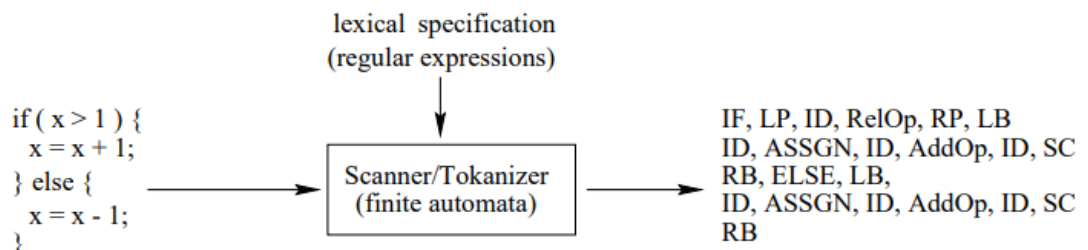
Лексичний аналіз (ЛА) - процес аналізу вхідної послідовності символів на розпізнані групи - лексеми, з метою отримання на виході ідентифікованих послідовностей, які звуться токенами.

Лексичний аналіз необхідний для спрощення наступних етапів обробки мови. Засобами реалізації ЛА є кінцеві автомати та регулярні вирази. Вони описують послідовності символів, які відповідатимуть токенам. Приклади токенів наведено у таблиці.

тип токена	приклади лексем	опис
num	257	число
id	Ident951	ідентифікатор
relop	<=	операція відносини
string	«Символ»	рядок

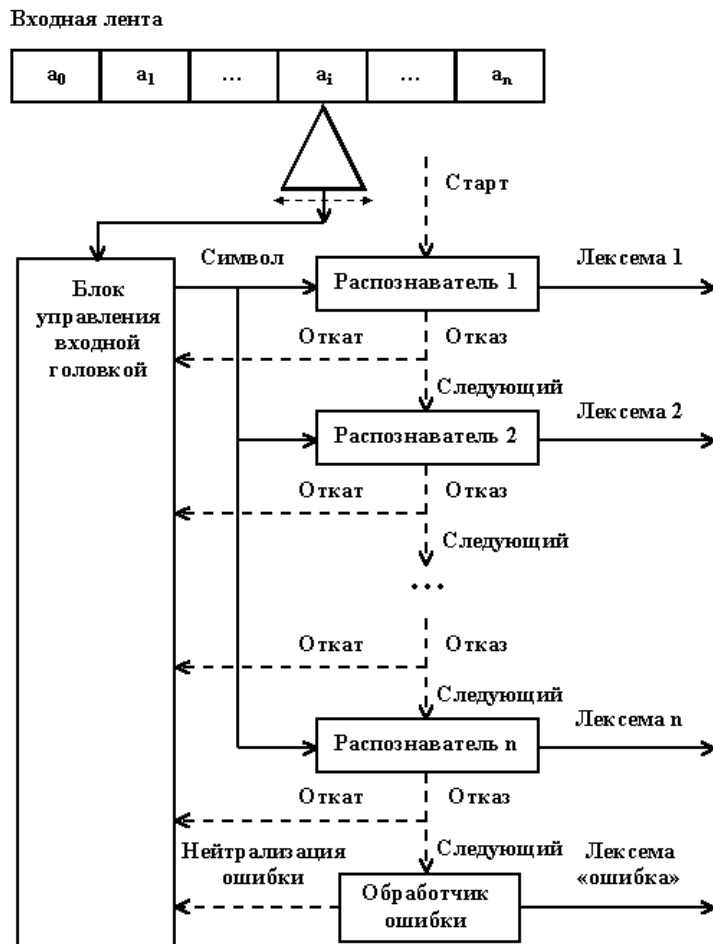
Кожен токен описується шаблоном - регулярним виразом, що реалізується на основі кінцевого автомата. Шаблон визначає множину рядків, що ідентифікуватиметься токеном. Наприклад, всі рядки, що є числами, повинні бути ідентифіковані як токен - num.

У процесі роботи ЛА всі розпізнані рядки замінюються на відповідні токени. Приклад такого перетворення наведено на малюнку нижче.



Входом ЛА є програма мовою програмування. Результатом роботи лексичного аналізатора є послідовність токенів, які є одиницями аналізу на наступних етапах обробки, та таблиця символів, де зберігаються розпізнані лексеми для класів ідентифікаторів і чисел.

Загальну схему роботи ЛА наведено нижче.



## 2 ВИКОНАННЯ РОБОТИ

1. Вибрати мову програмування та конструкції мови, які будуть присутні у фрагменті програма, яка подаватиметься на вхід ЛА.
2. Описати класи рядків у вигляді регулярних виразів.
3. Розробити програму, що реалізує роботу ЛА на основі регулярних виразів або кінцевих автоматів.
4. Протестувати програму для різних вхідних даних. Звернути увагу на роботу у разі подання на вхід помилкових даних.
5. Оформити звіт, який має містити регулярні вирази, що описують лексику мови; рядки, подані на вхід ЛА, результати роботи ЛА, код програми.

## 3 ВАРІАНТИ МОВНИХ КОНСТРУКЦІЙ

1. Мова, що представляє оператор циклу `for` з тілом циклу у вигляді оператора присвоювання зі змінною або елементом масиву в лівій частині оператора присвоювання та арифметичним виразом у правій. До арифметичного виразу входять змінні, цілі числа, числа з крапкою.

2. Мова, що представляє оператор `if` з тілом у вигляді оператора присвоювання зі змінною або елементом масиву в лівій частині оператора та арифметичним виразом у правій. До арифметичного виразу входять змінні, цілі числа, числа з крапкою.
3. Мова, що представляє оператор `if else`, з тілом у вигляді оператора присвоювання, зі змінною або елементом масиву в лівій частині оператора та арифметичним виразом у правій. До арифметичного виразу входять змінні, цілі числа, числа з крапкою.
4. Мова, що представляє оператор циклу `while` з тілом циклу у вигляді оператора присвоювання, зі змінною та елементом масиву в лівій частині оператора присвоювання та арифметичним виразом у правій. Заголовок пишеться із зазначенням умови виконання циклу. До арифметичного виразу входять змінні, цілі числа, числа з крапкою.
5. Мова, що представляє оператор циклу `do while` з тілом циклу у вигляді оператора присвоювання зі змінною та елементом масиву в лівій частині оператора присвоювання та арифметичним виразом у правій. Заголовок пишеться із зазначенням умови виконання циклу. До арифметичного виразу входять змінні, цілі числа, числа з крапкою.