

به نام خدا

طراحی و پیاده‌سازی تحلیلگر لغوی (lexer)



دانشگاه شهید بهشتی، تهران

بهار 1403

پوریا اسداللهی (400405008)

محمد دوکوهکی (400407009)

آیدا حافظیان (99407007)

استاد: دکتر فاطمه یوسفی نژاد

درس: طراحی کامپایلر

تحلیلگر لغوی با زبان C++ و با Clion(minGW) کامپایل شده.

فایل اجرایی(.exe). برنامه در پوشه‌ی cmake-build-debug قرار دارد.

برنامه کد برنامه‌ی ورودی را از فایل command.txt خوانده و در صورت عدم وجود خطا خروجی را در فایل output.txt قرار میدهد.

\*در صورتی که برای اجرای کد برنامه را از دوباره کامپایل کردید بسته به نوع IDE ممکن است فایل تکست ورودی و خروجی در جای متفاوتی ایجاد شوند.

## طراحی

کلاس های توکن زبان مورد نظر را تعریف میکنیم:

- identifier
- int
- double
- operator
- keyword
- string
- delimiter
- comment
- semicolon
- error

این توکن ها به صورت یک enum class در اول برنامه تعریف شده اند

حالا به ازای هر کلاس، یک عبارت منظم مینویسیم:

$$\Sigma = \{a, b, c, \dots, z, A, B, \dots, Z, 0, 1, 2, \dots, 9, (, ), \{, \}, ;, ", ., +, -, *, /, \backslash, =, >, <, !\}$$
$$\text{Letter} = 'a' + 'b' + 'c' + \dots + 'z' + 'A' + \dots + 'Z'$$



## پیاده‌سازی

پیاده سازی با زبان C++ و برنامه‌نویسی شی گرا انجام شده است.

### کلاس توکن (Token class):

- یک توکن را با یک نوع (type) و یک مقدار (value) تعریف می‌کند.
- متدهایی برای چاپ کردن توکن و مقایسه توکن فراهم میکند.

### کلاس لکسر (Lexer class):

- تبدیل یک رشته به توکن‌ها را بر عهده دارد.
- کلمات کلیدی (Keywords) و عملگرها (Operators) در این کلاس تعریف شده‌اند.
- متدی برای انجام عملیات توکن سازی (tokenize) رشته ورودی دارد.

### تابع های کمکی:

- IsOperator: تشخیص عملگر بودن کاراکتر
- IsDelimiter: تشخیص دلیمیتور بودن کاراکتر

### تابع main:

- 1- کد برنامه را از فایل command.txt خوانده و در یک رشته قرار میدهد (در اصل در یک ISS).
- 2- با رشته ورودی یک شی Lexer ساخته و متد tokenize را روی آن صدا می‌زنند.
- 3- در صورت وجود error در ترمینال عبارت "error" را چاپ میکند.
- 4- در صورت عدم وجود error خروجی برنامه را در فایل output.txt چاپ میکند.