



درس: طراحی کامپایلر

عنوان: پروژه تحلیل گر لغوی

عضو اول: علیرضا کریمی

شماره دانشجویی عضو اول: ۹۹۳۶۲۳۰۳۵

عضو دوم: علی پورقیصری

شماره دانشجویی عضو دوم: ۹۹۳۶۱۳۰۱۴

استاد: دکتر آرش شفیعی

فهرست مطالب

۱- مقدمه.....	۲
۲- مجموعه متغیر های سراسری.....	۳
۳- کلاس Token.....	۳
۴- توابع مورد استفاده در طول کد.....	۳
۴-۱- is_comment(line).....	۴
۴-۲- space_separator(line).....	۴
۴-۳- space_operator(line).....	۴
۴-۴- find_word_occurrences(word, line).....	۴
۴-۵- analyze_line(line, row).....	۴
۴-۶- analyze_file(file_path).....	۴
۴-۷- main().....	۴
مراجع.....	۴

۱- مقدمه

در فاز اول پروژه، قرار بر این است که یک تحلیلگر لغوی را از صفر با استفاده از زبان پایتون پیاده سازی کنیم. تحلیلگر لغوی در این کد با استفاده از ترکیبی از دیکشنری ها برای تعریف کلمات کلیدی، عملگرها، جداکننده ها، نقل قول ها و کاراکترهای فضای خالی پیاده سازی شده است. کلاس تعریف شده در این کد ویژگی های هر توکن را در بر می گیرد و روش های درون آن بر اساس قوانین از پیش تعریف شده، نوع توکن را تعیین می کنند. تحلیلگر خطوط ورودی را با تقسیم کردن آنها به کلمات، و سپس تکرار روی هر کلمه برای تعیین نوع رمز آن، نشانه گذاری می کند. فضاهایی در اطراف جداکننده ها و عملگرها برای تسهیل توکن سازی درج شده اند. به طور کلی، تحلیلگر لغوی کد خام را پردازش می کند، هر عنصر را به انواع نشانه های خاص دسته بندی می کند و نمایشی ساختاریافته از کد را برای پردازش یا تحلیل بیشتر ارائه می دهد.

۲- مجموعه متغیرهای سراسری

این متغیرهای سراسری در سرتاسر کد برای دسته‌بندی نشانه‌ها در طول تحلیل لغوی استفاده می‌شوند. با استفاده از آنها منطق متمرکز برای تعریف و مدیریت انواع توکن‌ها، افزایش خوانایی، قابلیت نگهداری و انعطاف پذیری کد را میتوان فراهم نمود. این کاراکترها شامل انواع علائم منطقی، نشانه گذاری، بلوک بندی و ... می باشد که معمولاً در سایر زبان‌ها نیز وجود دارند.

۳- کلاس Token

کلاس Token در این کد به عنوان ساختاری برای نمایش توکن‌های فردی در کد استفاده شده است. این کلاس شامل ویژگی‌هایی برای ذخیره واژگان، نوع آن، نام نشانه یا شناسه، و موقعیت آن در کد (ردیف و ستون) است. علاوه بر این، شامل روش‌هایی برای تعیین نوع نشانه بر اساس معیارهای از پیش تعریف شده مانند کلمات کلیدی، عملگرها، جداکننده‌ها، نقل قول‌ها، فضای خالی، شناسه‌ها، ثابت‌ها و نظرات می‌باشد. متد `analyze_lexeme()` توکن را بر اساس این معیارها دسته‌بندی می‌کند و مقادیر مناسب نوع و نام توکن را به آن اختصاص می‌دهد. به طور کلی، کلاس **Token** منطق تجزیه و تحلیل و طبقه‌بندی نشانه‌ها را در فرآیند تحلیل واژگانی یک زبان برنامه‌نویسی در بر می‌گیرد.

۴- توابع مورد استفاده در طول کد

در طول برنامه یکسری توابع استفاده شده است که پروسه تحلیل لغوی از طریق این توابع انجام می‌گیرد. این توابع به شرح زیر می‌باشند:

۴-۱- `is_comment(line)`: این تابع بررسی می‌کند که آیا یک خط کد معین یک کامنت است یا نه، با بررسی اینکه آیا با نشانگر کامنت // شروع می‌شود یا خیر.

۴-۲- `space_separator(line)`: این تابع فضاهای اطراف جداکننده‌ها (مانند پرانتز، پرانتز، براکت، نقطه ویرگول و کاما) را در یک خط کد مشخص درج می‌کند. این مرحله برای توکن‌سازی صحیح کد ضروری است.

۴-۳- `space_operator(line)`: این تابع مانند قبل، فضاهای اطراف عملگرها (مانند عملگرهای حسابی، مقایسه و منطقی) را در یک خط کد معین درج می‌کند. این تضمین می‌کند که هر اپراتور در طول توکنایز شدن به عنوان یک توکن جداگانه در نظر گرفته می‌شود.

۴-۴- `find_word_occurrences(word, line)`: این تابع با در نظر گرفتن هر کاراکتر رزرو شده، همه موارد تکرار یک کلمه را در یک خط کد پیدا می‌کند. لیستی از موقعیت‌هایی را که کلمه در خط ظاهر می‌شود، برمی‌گرداند.

۴-۵- `analyze_line(line, row)`: این تابع یک خط کد را تجزیه و تحلیل می کند، آن را نشانه گذاری می کند و لیستی از اشیاء Token را که نشان دهنده هر نشانه یافت شده در خط است، برمی گرداند. از توابع قبلی برای پیش پردازش خط، پیش از توکن سازی استفاده می کند.

۴-۶- `analyze_file(file_path)`: این تابع یک فایل را خط به خط می خواند، از کامنت ها می گذرد و هر خط بدون نظر را با استفاده از تابع `analyze_line(file_path)` آنالیز می کند. جزئیات هر نشانه موجود در فایل را چاپ می کند، مانند واژگان، نوع نشانه و موقعیت در کد. این تابع به عنوان نقطه ورودی برای توکن کردن کل فایل عمل می کند.

۴-۷- `main()`: این تابع جایست که `analyze_file(file_path)` فراخوانی می شود.

مراجع

- chat.openai.com
- [Introduction of Lexical Analysis - GeeksforGeeks](https://www.geeksforgeeks.org/introduction-of-lexical-analysis/)
- https://www.youtube.com/watch?v=O4Bt_CyZWbI
- <https://medium.com/@pythonmembers.club/building-a-lexer-in-python-a-tutorial-3b6de161fe84>