

توضیحات و کد پروژه تحلیلگر لغوی پاسکال

2024 شماره دانشجویی: 4000711328 : تاریخ

رضا خداحیمی : نویسنده

مقدمه

این سند توضیحات دقیقی درباره دو فایل کلیدی در پروژه تحلیلگر لغوی پاسکال ارائه می‌دهد: `tokenizer.py` و `tokenizer.lex`. هر دو فایل برای تحلیل کدهای پاسکال به توکن‌های معنادار تبدیل می‌کنند. همکاری می‌کنند.

توضیحات فایل: `tokenizer.lex`

هسته تحلیلگر لغوی در این پروژه است. این فایل قوانین مربوط به توکنیزه کردن `tokenizer.lex` را مشخص می‌کند. در زیر ساختار و توضیحات هر بخش آمده است.

کد شبهه: `tokenizer.lex`

بخش کد	توضیحات
<code>%{ ... %}</code>	کد C برای مقداردهی متغیرها و ساختار توکن
<code>%option noyywrap</code>	جلوگیری از بسته شدن خودکار ورودی
<code>\n</code>	افزایش شمارنده خط و ریست کردن شمارنده ستون
<code>[\t]+</code>	شناسایی توکن‌های فضای خالی
<code>//.*</code>	شروع می‌شوند // شناسایی نظرات که با
<code>if, then, else</code>	شناسایی کلمات کلیدی پاسکال
<code>[0-9]+(\.[0-9]+)?([eE][+-]?[0-9]+)?</code>	شناسایی مقادیر عددی (از جمله اعداد اعشاری)
<code>[a-zA-Z_][a-zA-Z0-9_]*</code>	شناسایی شناسه‌های پاسکال
<code><=, >=, ==, !=, <, ></code>	شناسایی عملگرهای رابطه‌ای
<code>.</code>	شناسایی هر توکن ناشناخته (خطاها)

توضیحات فایل: tokenizer.py

اسکرپت پایتون است که با تحلیلگر لغوی Lex ارتباط برقرار می‌کند. این فایل tokenizer.py فایل تحلیلگر تولید شده توسط Lex را اجرا کرده و خروجی را برای طبقه‌بندی توکن‌ها پردازش می‌کند.

کد شبهه: tokenizer.py

توضیحات	بخش کد
تعریف نوع توکن‌ها مانند IF, ELSE, RELOP, و غیره	class TokenType:
نمایش توکن با نوع، مقدار، خط و ستون	class Token:
اجرای تحلیلگر لغوی Lex و دریافت خروجی	run_lex_tokenizer(input_file)
پردازش خروجی Lex به اشیاء توکن	parse_tokens(lex_output)
تابع اصلی برای اجرای تحلیلگر و پردازش توکن‌ها	tokenize(input_file)
مثال کد برای اجرای تحلیلگر	Example usage (if __name__ == __main__):

نتیجه‌گیری

ستون فقرات پروژه تحلیلگر لغوی پاسکال tokenizer.py و tokenizer.lex به طور کلی، فایل‌های را تشکیل می‌دهند. فایل Lex قوانین تحلیل را تعریف می‌کند و اسکرپت پایتون خروجی را پردازش کرده و با رابط کاربری گرافیکی ارتباط برقرار می‌کند.