

プロジェクト演習

課題2

担当したソースコード

nextch
printitem
simplestat
statements
strliteral

初回の資料に書いた通り：

ソースコードの説明は、それを利用しようとする人に対して、

- どんな機能を提供しているのか

- どうやって使うのか

- 使う時の制約や注意点は何か、

などを知らせてくれるものであるべきです。

このレポートはそうではありません。前回からあまり進展も見られないようです。

nextch.c

tip構造体は何のためにある？

getTextBuffer関数ではtip構造体のそれぞれの変数を0で初期化している。

freeTextBuffer関数ではtip構造体のメモリ領域を解放している。

printTextBuffer関数ではisEOFが真か偽によって出力を変えている。

この関数の役割は？

get_new_line関数ではtip構造体を引数とし、tip構造体の各変数のそれぞれに値を代入している。

nextbyte関数では読み込んだ文字列を1文字ずつ解析している中で次の文字が存在しているかを判別する役割を持つ。 nextbyte() と nextch() はどう違う？

undoch関数ではchとtipを引数とし、tipのungetchが0のときungetchにchを代入している。

このプログラムは全体で、読み込んだ文字列を1文字ずつ解析する処理の次の文字への処理の移行の制御を行っている。 ?

nextch.h

nextch.cで用いる構造体、関数名を定義している。

printitem.c

どんな数値？

shorten関数ではs,limitを引数とし、配列bufに格納する数値を計算し返り値として出力する。

itemToString関数ではbuffer,pを引数とし、tの数値によってテキストを表示する。 どんなテキスト？

printItem関数ではpを引数にとし、pの状態によってデバッグ処理を行っていると思われる。

デバッグ処理とは何？

simplestat.c

なぜ？

checkAssignment関数ではkindの状態によって操作を中断させる。

assignStatement関数ではahead,terminatorを引数としている。checkAssignment関数を呼び出した後、getItemを代入したsの状態によって処理を行い、statmpを返り値としている。

returnStatement関数ではterminatorを引数にしている。構造体rtmpの3つの要素に数値を代入している。statmpを返り値としている。 前回から変更なく、内容も意味不明です。

inputStatement関数では構造体stpの要素に数値を代入している。stpを返り値としている。

forStatement関数では構造体stp,fopのそれぞれの要素に数値を代入している。stpを返り値としている。

statement.c

それぞれの返り値の型は何で、何を表していますか

statInitialize関数では構造体symset_tの各要素の初期化を行っている。

ifStatement関数ではduskulでif構文を解析する場合の処理の制御を行っている。stpを返り値としている。

whileStatement関数ではduskulでwhile構文を解析する場合の処理の制御を行っている。stpを返り値としている。

callStatement関数ではduskulでcall構文を解析する場合の処理の制御を行っている。stpを返り値としている。

printStatement関数ではduskulでprint構文を解析する場合の処理の制御を行っている。stpを返り値としている。

fetchStatement関数ではaheadを引数にしている。aheadの状態によって次に処理を移す関数を制御している。

varsDeclareList関数ではduskulでvarを用いて変数を宣言する際の制御を行っている。varsを返り値としている。

codeblock関数ではtermset,rtnflagを引数としている。termset状態、rtnflagの真偽によって適切な処理を行っている。rootを返り値としている。

statement.h

statement.cで用いる関数名、ポインタを定義している。

strliteral.c

関数addLiteralStringでは入力された文字列を、そのサイズに適したメモリを用意して保存している。

getsymbol.c内の関数get_stringで読み込んだ文字列を受け取っている。

strliteral.h

strliteral.cで用いる関数名、ポインタを定義している。