프로그래밍 언어론

언어 설계 Report

2017.04.26

20154073

강다현

Q.. 여러분이 개발하고 싶은 새로운 언어를 다음과 같이 설계하고 답하시오.

🏵BNF

START < statement-list > END

< var > 🡪 A | B | C | D | E

< const > 🡪 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< expr > 🡪 <expr> + < term > | <expr> - < term > | < term >

< term > 🡪 < term > \* < factor > | <term> / < factor > |

< term > % < factor > | < factor >

< factor > 🡪 < factor > \*\* < exp > | < exp >

< exp > 🡪 (< expr >) | < var > | < const >

< compare > 🡪 == | != | < | <=

< statement > 🡪 < assign > | < loop > | < control >

< statement-list > 🡪 < statement >, < statement-list > | < statement >;

< condition > 🡪 < exp > < compare > < exp >

< assign > 🡪 <var> = <const> | <var> = <var> | <var> = <expression>

< loop > 🡪 WHILE < condition> DO < statement-list > |

RECURSION < control >DO < statement-list > |

FOR ( < assign >, < condition >, <expr> ) DO < statement-list >

< control > 🡪 IF <condition> THEN < statement-list >

[ ELSE < statement-list > ]

🏵EBNF

START < statement-list > END

< var > 🡪 A | B | C | D | E

< const > 🡪 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< expr > 🡪 <term> { ( + | - ) < term > }

< term > 🡪 < factor > { ( \* | / | % ) < factor > }

< factor > 🡪 < exp > { \*\* < exp > }

< exp > 🡪 (< expr >) | < var > | < const >

< compare > 🡪 == | != | < | <=

< condition > 🡪 < exp > < compare > < exp >

< statement > 🡪 < assign > | < loop > | < control >

< statement-list > 🡪 < statement > { , < statement-list > }

< assign > 🡪 <var> = ( < const > | < var > | < expr > )

< loop > 🡪 WHILE < condition> DO < statement-list > |

RECURSION < control >DO < statement-list > |

FOR ( < assign >, < condition >, <expr> ) DO < statement-list >

< control > 🡪 IF <condition> THEN < statement-list > [ ELSE < statement-list > ]

🏵 언어에 속한 한 문장을 제시하고, 이 문장이 언어에 속하는지를 다음의 각 방법에 의해서 보여라

START

A = 1

WHILE A < 10 DO B = A, A = A + 2

END

🖉최좌단 유도

=> START < statement-list > END

=> START < statement >, < statement-list > END

=> START < assign> , < statement-list > END

=> START < var > = < const > , < statement-list > END

=> START A = < const > , < statement-list > END

=> START A = 1, < statement-list > END

=> START A = 1, < statement> END

=> START A = 1, < loop > END

=> START A = 1, WHILE < condition > DO < statement-list > END

=> START A = 1, WHILE < exp > <compare> < exp > DO < statement-list > END

=> START A = 1, WHILE < var > <compare> < exp > DO < statement-list > END

=> START A = 1, WHILE A <compare> < exp > DO < statement-list > END

=> START A = 1, WHILE A < < exp > DO < statement-list > END

=> START A = 1, WHILE A < < const > DO < statement-list > END

=> START A = 1, WHILE A < 10 DO < statement-list > END

=> START A = 1, WHILE A < 10 DO < statement >, < statement-list > END

=> START A = 1, WHILE A < 10 DO < assign >, < statement-list > END

=> START A = 1, WHILE A < 10 DO < var > = < var > , < statement-list > END

=> START A = 1, WHILE A < 10 DO B = < var > , < statement-list > END

=> START A = 1, WHILE A < 10 DO B = A , < statement-list > END

=> START A = 1, WHILE A < 10 DO B = A , < statement > END

=> START A = 1, WHILE A < 10 DO B = A , < assign > END

=> START A = 1, WHILE A < 10 DO B = A , < var > = < expr > END

=> START A = 1, WHILE A < 10 DO B = A , A = < expr > END

=> START A = 1, WHILE A < 10 DO B = A , A = < term > + < term > END

=> START A = 1, WHILE A < 10 DO B = A , A = < factor > + < term > END

=> START A = 1, WHILE A < 10 DO B = A , A = < exp > + < term > END

=> START A = 1, WHILE A < 10 DO B = A , A = < var > + < term > END

=> START A = 1, WHILE A < 10 DO B = A , A = A + < term > END

=> START A = 1, WHILE A < 10 DO B = A , A = A + < factor > END

=> START A = 1, WHILE A < 10 DO B = A , A = A + < exp > END

=> START A = 1, WHILE A < 10 DO B = A , A = A + < const > END

=> START A = 1, WHILE A < 10 DO B = A , A = A + 2 END

🖉파스트리

