GÖKBERK ERÜST-------PROJECT1 DESCRIPTION

2012400084

Program takes the directories of input and output file as arguments. First it takes whole input files as a string. Then it divides it according to given grammar and calls the functions(which are the non-terminals of myLang) recursively. When the recursion is done the functions returns an abstract code for the input File. According to this abstract code the assembly code is generated.

While dividing the string basically program checks the grammar of myLang and decide whether it is a statement or statement list, etc. For example when the scanner sees while it is automatically checks for do and the part between them is become an expression.

I counted all begins and ends because they are essential to decide the where the partial code belongs; is it an expression or statement, etc. I also counted parenthesis' because there can be expressions like "(5 + 3 + (4*5))" which I need the give priority.

After dividing the input string the functions returns the abstract code of them according to description of project. When I put the whole returns in a string then the method called abstractToAssembly makes them assembly code. I used two scanners to read the abstract code first one reads a line and second one reads one by one in the line.

Then I add the myprint and myread functions to the output string then I print it out the output file which is given by the arg[1] .

Programs run correctly.