Link for github repo:

<https://github.com/AndreiFoidas/LFTC>

<https://github.com/AndreiFoidas/LFTC/tree/main/src/main/java>

# Grammar

Is a class containing 4 fields: terminals (List of Strings), nonTerminals (List of Strings), startingSymbol (String) and productions (Hashmap were the key is a List of Strings and the value is a List containing a List of Strings), where each field is equivalent to the theoretical definition.

## Operations

* isCFG(): verifies if the grammar is context free by verifying if the left hand side of the productions (the key) includes a single nonTerminal and that the right hand side of the productions only contain terminals, nonTerminals or epsilon
* isValid(): verifies that the grammar is valid by checking to see if the starting symbol is a nonterminal, the left hand side (the keys) of the productions are all nonTeminals and that the right hand side (the values) of the productions are terminals, nonTerminals or epsilon
* getProductionsForNonTerminal(nonTerminal): it will return a List containing a List of Strings representing all the productions that have that nonterminal in the left hand side (as the key)

## Tests



