Tătar Flavia-Andreea 937

<https://github.com/tatarflavia/Formal-Languages-Compiler-Design-LAB/tree/main/lab2>

Functions used:

1.

*//returns position in hash list aka hash value from hash function***private int** getHashValue(String tokenName){  
 **int** asciiSum = 0;  
 **for** (**int** i = 0; i < tokenName.length(); i++){  
 asciiSum += (**int**)tokenName.charAt(i);  
 }  
 **return** (**int**)(asciiSum%256);  
}

-this function accepts a string which is the identifier name as an input and as an output it returns the position in hash list (hash value from hash function)

-makes sum of ascii values for the string given and it divised it by 256 and takes the remainder from the operation

2.

*//returns position in symbol table***public** AbstractMap.SimpleEntry<Integer,Integer> position(String tokenName){  
 **int** hashValue=**this**.getHashValue(tokenName);  
  
 **if**(**this**.**hashTable**.get(hashValue)==**null**){  
 *//this means we are adding in the head of the linked list of this hash value, cause there is no member yet here added* ArrayList<String> linkedList=**new** ArrayList<>();  
 linkedList.add(tokenName);  
 **this**.**hashTable**.set(hashValue, linkedList);  
 **return new** AbstractMap.SimpleEntry<Integer,Integer>(hashValue, 0);  
 }  
 **else**{  
 *//this means that there is already a hash here, so we put it in the link* List<String> linkedList=**this**.**hashTable**.get(hashValue);  
 linkedList.add(tokenName);  
 **this**.**hashTable**.set(hashValue,linkedList);  
 **return new** AbstractMap.SimpleEntry<Integer,Integer>(hashValue, linkedList.size()-1);  
 }  
}

-this function accepts a string which is the identifier name as an input and as an output it returns the position in hash list and linked list, a pair of them

-gets hash value and checks to see if the linked list from the index of the list= hash value is null or not, if null then we add a new linked list to that pos with the token as a head, and returns the pair . If not null, then we add to the end of the linked list the new token and return the new pair.