Link to github:
https://github.com/Akitektuo/University/tree/master/3rd%20year/FLCT

Link to lab 1b pull request:
https://github.com/anamariadem/lftc-mini-language/pull/2

Symbol table implementation using hash table

The symbol table class takes into account the initial capacity and a load factor. The capacity is by default set to 17, it has been proven that using a primitive number for modulo can improve the performance of the hashing function. The load factor is set to 0.75 by default, being a sweet spot for balancing computation times and the number of rebalancing the hash map when the capacity increases. The hash function used for this implementation takes the identifier or constant as string, sums all values of the characters and then takes the remainder of division by the current capacity. The capacity increases each time the load factor is exceeded computed by size divided by the current capacity.

Add

The add method of the symbol table takes the identifier or constant as a string and returns its position from the table. First, it checks if the token is already part of the table by calling the **find()** method. If the returned pair is not null, then it returns the found pair, otherwise it checks the load factor then increases the size, hashes the given string, stores the position that the given string will take, adds it to the table based on the hash key and returns the pair of hash key and position.

Find

The find method of the symbol table takes the identifier or constant as a string and returns its position from the table if it is found or null. The hash key is computed based on the given string and based on it, the list for the computed hash key is searched for the given token. In case it was not found, returns null, otherwise returns the pair of hash key and found position.

GrowIfLoadFactorExceeded

The method check whether the load factor has been exceeded and if the size divided by the capacity is greater or equal to the given load factor, a copy of the existing items is made, the capacity is doubled, the table's items are reinitialized with the new capacity and each pre existing token is added back.

GetTableOfCapacity

The method creates and returns a new array of current capacity, each position being an empty list.