Formal Languages and Compiler Design

Lab 2 – Documentation

Ariadna Vilasó Escarp

Link to the repository:

https://github.com/UO270119/Formal-Languages-and-Compiler-Design

Problem statement:

Implement the Symbol Table (ST) as the specified data structure, with the corresponding operations

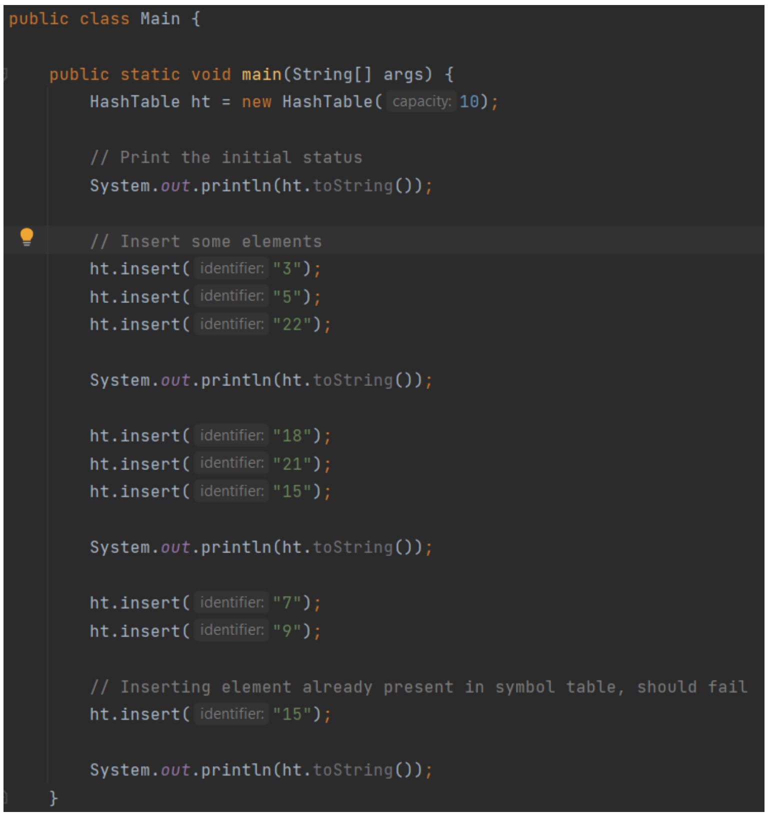
Deliverables: class ST(source code) + documentation.

HashTable: To represent the Symbol Table I chose to use the HashTable structure. The possible collisions will be solved using a linked list. The hash function we will use will be computed calculating the modulus of the ASCII code of the character from the corresponding token, divided by the capacity, which is the size of the token list.

Its methods are:

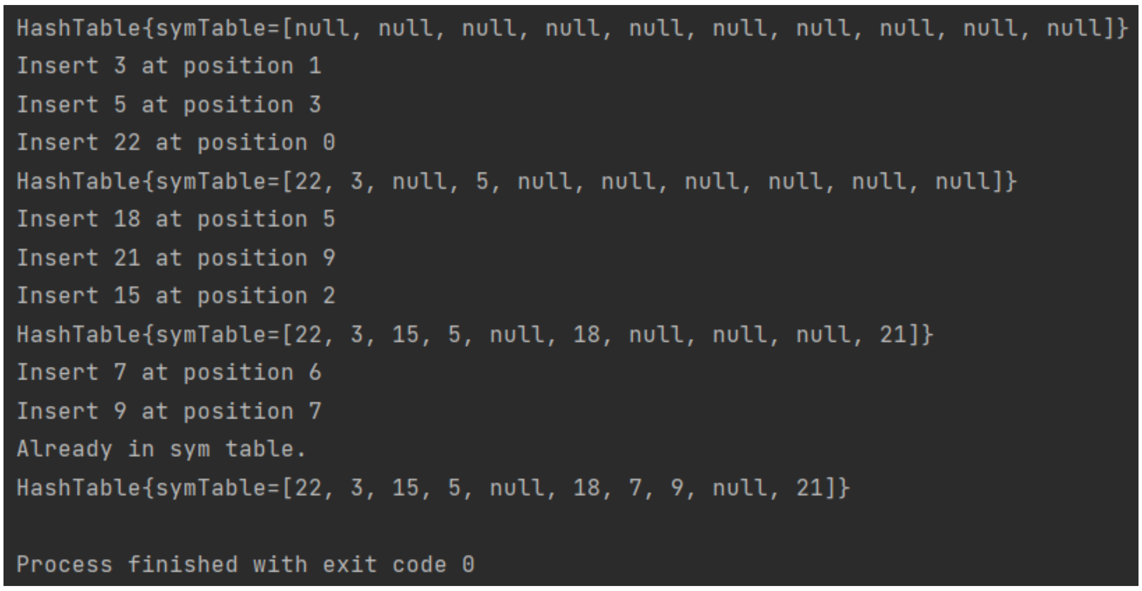
- *private int hashingFunction (String identifier):* Calculates the hashing function (explained above).
- *public boolean insert (String identifier):* Inserts a token in the symbol table while solving possible collisions.

In order to see if the functionality works, I wrote some tests:



```
public class Main {

    public static void main(String[] args) {
        HashTable ht = new HashTable( capacity: 10);

        // Print the initial status
        System.out.println(ht.toString());

        // Insert some elements
        ht.insert( identifier: "3");
        ht.insert( identifier: "5");
        ht.insert( identifier: "22");

        System.out.println(ht.toString());

        ht.insert( identifier: "18");
        ht.insert( identifier: "21");
        ht.insert( identifier: "15");

        System.out.println(ht.toString());

        ht.insert( identifier: "7");
        ht.insert( identifier: "9");

        // Inserting element already present in symbol table, should fail
        ht.insert( identifier: "15");

        System.out.println(ht.toString());
    }
```

And I obtained these results:



```
HashTable{symTable=[null, null, null, null, null, null, null, null, null, null]}
Insert 3 at position 1
Insert 5 at position 3
Insert 22 at position 0
HashTable{symTable=[22, 3, null, 5, null, null, null, null, null, null]}
Insert 18 at position 5
Insert 21 at position 9
Insert 15 at position 2
HashTable{symTable=[22, 3, 15, 5, null, 18, null, null, null, 21]}
Insert 7 at position 6
Insert 9 at position 7
Already in sym table.
HashTable{symTable=[22, 3, 15, 5, null, 18, 7, 9, null, 21]}

Process finished with exit code 0
```