

SOURCE CODE: [LabLFTC/Lab2/Lab2 at main · davidalexandru1370/LabLFTC \(github.com\)](https://github.com/davidalexandru1370/LabLFTC/blob/main/Lab2/Lab2)

## Documentation

- For the symbol table I've used a generic hash table with separate chaining as collision handler. The size parameter from the constructor means the size of the hashtable (number of cells). The symbol table stores the identifiers, strings and hashtables.
- The hash functions depend on the data type, for integers the hash function simply hash that number, for strings it sums up the ascii codes and hashes the sum. The hash number increases the efficiency of the hash function(mathematically, not asymptotically) if the number is prime.
- The hashtable uses an array of ArrayLists and it inserts the object at the index computed by hash function

Methods:

- Hashtable:

getSize() - returns the size of the hashtable

hash(key: Int) - hash the key by computing  $\text{key} \% \text{size}$

hash(key: String) - hash the sum of ascii codes of key computing  $\text{sum} \% \text{ascii}$

hash(key: Char) - hash the key by computing  $\text{key} \% \text{size}$

computeHashValue(key: T) - compute the hash of the key treating each known data type

insert(key: T) - insert a new value into the hashtable

contains(key: T) - returns true if the hash table contains the key, otherwise false

getPosition(key: T) - returns the position of the hashed key in the ArrayList representation

- Symbol table

addIntConstant(value: Int) - returns the a pair of integers first one representing the hash value and the second one the index in the array list

addIdentifier(value: String) - returns the a pair of integers first one representing the hash value and the second one the index in the array list

addStringConstant(value: String) - returns the a pair of integers first one representing the hash value and the second one the index in the array list

hasStringIdentifier(string: String) - returns true if the string exists in the string hashtable, otherwise false  
 hasIntIdentifier(int: Int) - returns true if the integer exists in the integers hashtable, otherwise false  
 hasIdentifier(identifier: String) - returns true if the integer exists in the identifiers hashtable, otherwise false  
 getIdentifierPosition(identifier: String) - returns a pair of integers, where the first integer is the hash and the second one the index in the array list  
 getIntIdentifierPosition(int: Int) - returns a pair of integers, where the first integer is the hash and the second one the index in the array list  
 getStringIdentifierPosition(string: String) - returns a pair of integers, where the first integer is the hash and the second one the index in the array list  
 getIntByPosition(position: Pair<Int, Int>) - returns the int by given position  
 getStringByPosition(position: Pair<Int, Int>) - returns the string by given position  
 getIdentifierByPosition(position: Pair<Int, Int>) - returns the identifier by given position

The screenshot shows the IntelliJ IDEA IDE with the following components:

- Project Files:** A sidebar on the left showing the project structure. The path is `Lab2 > src > main > kotlin > tests > SymbolTableTests`. Other files visible include `HashTable`, `Main.kt`, and `SymbolTable`.
- SymbolTableTests.kt:** The main editor window displays the following Kotlin code:
 

```

class SymbolTableTests {
    init {
        println("Start executing tests")
        testAddStringIdentifier_shouldReturnTheAddedString_returnsAddedString()
        testAddIntIdentifier_shouldReturnTheAddedInt_returnsAddedInt()
        testAddIdentifier_shouldThrowNotFoundException_throwNotFoundException()
        testHasStringIdentifier_ValidIdentifier_shouldReturnTrue()
        testHasStringIdentifier_InvalidIdentifier_shouldReturnFalse()
        testHasIntIdentifier_ValidIdentifier_shouldReturnTrue()
        testHasIntIdentifier_InvalidIdentifier_shouldReturnFalse()
        testHasIdentifier_ValidIdentifier_shouldReturnTrue()
        testHasIdentifier_InvalidIdentifier_shouldReturnFalse()
        println("Executed all tests successfully")
    }
}
      
```
- Run Console:** The bottom panel shows the execution output:
 

```

C:\Users\david\.jids\openjdk-21\bin\java.exe -ea "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.2\lib\idea_rt.jar=61873:C:\Program Files\JetBr
Start executing tests
Executed all tests successfully
Process finished with exit code 0
      
```
- Status Bar:** The bottom status bar indicates the file is at `Lab2 > src > main > kotlin > tests > SymbolTableTests`, with settings for `2:1` lines, `CRLF` line endings, `UTF-8` encoding, and `4 spaces` indentation.