Documentation

link to git:

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

---

My implementation of the Symbolic Table is based on the hash table, designed to store and manage symbols (identifiers, integer and string constants). The hash table is using separate chaining (with linked lists) to handle collisions and supports both String and Integer keys. The operations are designed to store key-value pairs efficiently wit collision handling.

Hash Table

Operations:

- ***add(K key, V value)***:
    - Adds a key-value pair to the hash table.
    - If the key already exists, throws an exception.
    - **Returns**: A ***Pair<Integer, Integer>*** indicating the bucket index and the position within the list.
- ***contains(K key)***:
    - Checks if the key exists in the hash table.
    - **Returns**: ***true*** if the key is present, otherwise ***false***.
- ***getPosition(K key)***:
    - Retrieves the bucket index and position of the key in the hash table.
    - **Returns**: A ***Pair<Integer, Integer>*** indicating the position or ***(-1, -1)*** if not found.
- ***toString()***:
    - Converts the hash table into a string representation.

    - Lists all key-value pairs in the table and their respective buckets.

Hash Functions:

- **Integer Key Hashing**: Uses key % size.
- **String Key Hashing**: Sums the ASCII values of characters and applies sum % size.

Symbol Table

Key Features:

- Two instances of a hash table are used:
    - One for **identifiers**.
    - One for **constants** (both integers and strings).
- Identifiers and constants can be added to and retrieved from the table.

Operations:

- ***addIdentifier(String name, String type)***:
    - Adds a new identifier to the symbol table.
    - **Returns**: A ***Pair<Integer, Integer>*** with the identifier's position in the hash table.
- ***addIntegerConstant(int constant)***:
    - Adds an integer constant to the symbol table.
    - **Returns**: A ***Pair<Integer, Integer>*** with the constant's position.
- ***addStringConstant(String constant)***:
    - Adds a string constant to the symbol table.
    - **Returns**: A ***Pair<Integer, Integer>*** with the string's position.
- ***getPositionIdentifier(String name)***:
    - Retrieves the position of an identifier in the symbol table.

    - **Returns**: A ***Pair<Integer, Integer>*** with the identifier's position or ***(-1, -1)*** if not found.
- ***getPositionStringConstant(String constant):***
    - Retrieves the position of a string constant in the symbol table.
    - **Returns**: A ***Pair<Integer, Integer>*** with the string constant's position or ***(-1, -1)*** if not found.
- ***getPositionIntegerConstant(int constant):***
    - Retrieves the position of an integer constant in the symbol table.
    - **Returns**: A ***Pair<Integer, Integer>*** with the integer constant's position or ***(-1, -1)*** if not found.
- ***toString()***::
    - Provides a string representation of the entire symbol table, showing identifiers and constants stored.

I have also defined the Entry as a data structure. It is composed of a key and a value.