IMPLEMENTACIÓN DE ESTRUCTURAS DE LA JAVA COLLECTION FRAMEWORK

Lisp es una familia de lenguajes de tipo multiparadigma, cuya estructura base está dada bajo la organización y utilización intensiva de listas. Dados estos argumentos y la realización recurrente del proyecto 1 bajo el lenguaje Java, se ha de recurrir a recursos que puedan sostener este comportamiento en forma eficiente y conveniente en cuanto a materia de desarrollo e interpretación. Por lo que se ha recurrido a la ***Java Collection Framework***, la cual almacena recursos excepcionales en ámbitos de listas y adyacentes, recurso que será prioritario al momento de desarrollar dicha herramienta interlingüística.

Dada la vastedad de recursos disponibles, se ha de limitar a los plenamente utilizados a lo largo de la misma implementación, los recursos descritos a continuación planean destacar el objetivo de su utilización y así mismo referenciar donde fueron utilizados. Todos los recursos parten de la ***Java Collection Framework***.

Los recursos utilizados son descritos a continuación:

ArrayList
El recurso denominado ArrayList fue utilizado con el fin de manejar un tipo de lista dinámica, este tipo de recurso nos permitiría almacenar un número indefinido como dinámico de recursos entrantes. El dinamismo del ArrayList nos permitiría captar flexibilidad en el programa, permitiendo que este se ejecute así reciba grandes o pequeños montos de entradas. Al desconocer ya sea el número de condiciones, parámetros o instrucciones que el usuario ingresaría, se optó por su utilización qué es lo más conveniente para mantener el flujo operativo del programa.

Especificación
Ingreso y salida de datos, dinamismo de tamaño, fácil extracción, eliminación y obtención de datos.

Área de referencia
- Almacenamiento de variables (Context class)
- Ingreso de valores a funciones aritméticas (IOperaciones & OpAritmeticas)
- Almacenamiento de parámetros (Runtime class)
- Instrucciones actuales (Runtime class)
- Comando Separado (Runtime class)
- etc.

List
El recurso List pretende delimitar el número de entradas almacenadas dentro de la misma, es utilizado precisamente para delimitar el dinamismo de la lista a costa de aumentar el desempeño dado su comportamiento límite. El recurso planea ser utilizado en casos donde se conoce el límite de entradas, evitando así los problemas en relación al index. Se optó por este recurso con el fin de evitar de recursos dinámicos en situaciones innecesarias.

Especificación
Ingreso y salida de datos, número limitado de espacios, posibles problemas de index.

Stack
El recurso Stack mucho más allá de enfocar su funcionamiento al ingreso de entradas, focaliza su desempeño en gran medida a la salida de estas. Su dinamismo y su forma de acceso a los datos permiten llevar a cabo operaciones aritméticas, de comparación y de condición bajo la forma LIFO, que coincide plenamente con la característica sintaxis de notación polaca propia de LISP. El dinamismo como el acceso LIFO de los datos fue fundamental para el buen desarrollo del intérprete.

Especificación
Ingreso y salida de datos, último en entrar primero en salir, dinámico en cuanto a tamaño.

Área de referencia
- Contador Operaciones Aritméticas (Runtime class)
- Contador Instrucciones (Runtime class)
- Contador Funciones (Runtime class)
- etc.

HashMap (Maps)
El recurso HashMap pretende presentar una opción dinámica a la relación de valores bajo la forma key-value, permite identificar valores de entrada con su respectivo valor almacenado. Este recurso permite relacionar básicamente valores presentes en parámetros, condiciones e incluso funciones.

Especificación
Ingreso y salida de datos, dinamismo de tamaño, fácil extracción, eliminación y obtención de datos y almacenamiento par de recursos.

Área de referencia
- Almacenamiento de la variable junto a su valor (Context class)

REFERENCIAS

- Reddy, Abhishek (22 de agosto de 2008). «Features of Common Lisp». Archivado desde el original el 26 de diciembre de 2009.
- Java Point (2019). “Collections in Java”. Retrieved from: https://www.javatpoint.com/collections-in-java
- "Lesson: Introduction to Collections". Oracle Corporation. Retrieved 2010-12-22.
- "Java Collections Framework" (PDF). IBM. Archived from the original (PDF) on 2011-08-07. Retrieved 2011-01-01.