Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2023.

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 2
PROYECTO 2

201709149 – Percy Juventino Duarte Gálvez

Resumen

Se ha analizado los pasos necesarios que deba seguir una cierta máquina seleccionada para poder crear un compuesto detallando el tiempo que le tomará si cada acción representa un segundo, se ha buscado el tiempo óptimo en segundo y cada paso que debe dar cada uno de los pines de la máquina para poder lograr crear un compuesto empleando estructuras TDA que simulan el arreglo rectangular de N filas y M posiciones de elementos que posee una máquina para la creación de compuestos mientras le sea posible con los elementos que posea. Los elementos almacenados se disponen para mostrarse ordenados por número atómico y los compuestos acompañados de su fórmula.

Palabras clave

Coordenada: sistema de referencia para ubicación de posición.
Matriz: arreglo bidimensional de números o datos.
Camino: recorrido unidireccional con posición inicial y final entre organismo.

TDA: Tipo de Dato Abstracto (modelo de datos en memoria dinámica).
XML: Xtensible Markup Language (Lenguaje de marcado extensible).

Abstract

The necessary steps that a certain selected machine must follow in order to create a compound have been analyzed, detailing the time it will take if each action represents one second, the optimal time in seconds has been sought and each step that each of the pins must take of the machine to be able to create a compound using TDA structures that simulate the rectangular arrangement of N rows and M positions of elements that a machine has for the creation of compounds while it is possible with the elements it has. The stored elements are arranged to be displayed ordered by atomic number and the compounds accompanied by their formula.

Keywords

ADT: *Abstracta Data Type (dynamic memory data model).*
Coordinate: *reference system for position location.*
Matrix: *bidimensional array of numbers or data.*
Orthogonal: *unidirectional path with initial and final position between organisms.*

XML: *Xtensible Markup Language.*

Introducción

Dada una serie de elementos químicos éstos han sido ordenados según el número atómico que les corresponde para listarles y almacenarles con un definido ordenamiento. Se obtienen de los archivos de entrada los datos para la simulación de máquinas compuestas por una serie de pines con ciertas posiciones donde se almacenan elementos. Las máquinas tienen como propósito la creación de compuestos al realizar acciones con los pines independientes consumiendo un segundo de tiempo por cada acción las cuales pueden ser avanzar o retroceder entre pines para ubicar el elemento correcto y fusionarlo en caso de estar en la ubicación adecuada. Una máquina hace un compuesto basada en la fórmula química de dicho compuesto de manera secuencial y ordenada.

a) Estructura del programa

El algoritmo principal desarrollado para la solución consta del recorrido independiente entre pines al cual se le asigna una tarea a cada uno dependiendo de los elementos en cola, si el siguiente elemento en cola se encuentra en un pin, dicho pin tiene la tarea de avanzar hasta encontrarlo, una vez lo encuentra éste es fusionado por lo que se extrae de la cola. Si un pin tiene un elemento, pero no es el primero en la cola éste deberá esperar hasta que sea el turno de dicho elemento, siguiendo ésta secuencia se eliminan de la cola los elementos uno por uno hasta tener vacía la cola. Cada acción es registrada según el pin que la realizó y el segundo en que ocurrió. Finalmente, con la información obtenida se grafican los pasos detallando el pin, segundo y tipo de acción realizada. Para todo análisis se verifica previamente que la máquina sea capaz de realizar un compuesto, esto quiere decir que obtenga dentro de sus pines a todos los elementos que contiene la fórmula del compuesto.

b) Clases

b.1) Clases UI:

Las clases UI (Main_UI, Second_UI, Third_UI) se encargan de mostrar la interfaz gráfica de usuario, así como el manejo inicial de datos y operaciones (tales como agregar, analizar y graficar) que pueda hacer el usuario comunicando el resto de clases.

Figura 1. Interfaz de usuario.
Fuente: elaboración propia.

b.1.1) Inicialización

Resetea todos los datos almacenados a su estado inicial en el que las estructuras no poseen datos definidos.

b.1.2) Cargar XML

Permite al usuario eleccionar el archivo de entrada. El archivo de entrada debe poseer la extensión .xml para ser válido.

b.1.3) Generar XML

Escribe un archivo de salida en formato .xml que contiene la información de todos los compuestos que hayan sido analizados hasta el punto de la solicitud. El formato de archivo cuenta con la información del compuesto tales como su tiempo óptimo de elaboración, máquina que lo elabora y acciones de los pines de la máquina para hacerlo.

b.1.4) Listado de elementos

Muestra todos los elementos que existen almacenados en memoria. Los elementos se muestran en una gráfica de Graphviz y son ordenados por número atómico previo a realizar su gráfica.

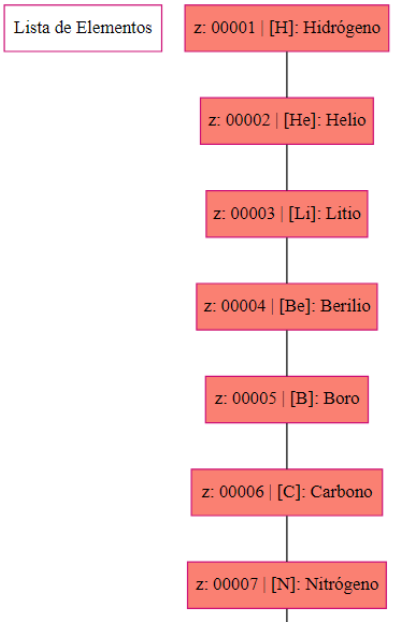


Figura 2. Gráfica de elementos.
Fuente: elaboración propia.

b.1.5) Agregar elemento

Permite al usuario introducir un número atómico, símbolo y nombre manualmente para almacenarlo en memoria. El elemento puede ser guardado siempre y cuando los datos previamente mencionados no coincidan con ninguno de los guardados anteriormente.

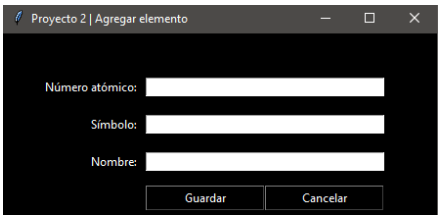


Figura 3. Agregación de elementos
Fuente: elaboración propia.

b.1.6) Listado de compuestos

Permite visualizar en gráfica los compuestos almacenados con su nombre y fórmula de elaboración.

b.1.7) Realizar análisis

Permite al usuario seleccionar el compuesto y máquina para realizar un análisis de los pasos y tiempos necesarios para elaborarlo. Permite también la elaboración de un análisis múltiple con todas las máquinas de un solo compuesto.

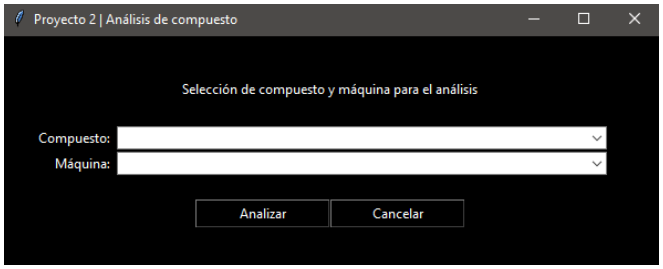


Figura 4. Análisis de compuesto
Fuente: elaboración propia.

b.2) Element

Nodo de la clase ElementsList que guarda la información de elementos, tales como su nombre, símbolo y número atómico.Los datos son únicos entre demás nodos elementos. Éstas listas son de tipo doblemente enlazadas para poder hacer uso de éstas propiedades en el manejo de pines los cuales se componen también de nodos Elemento.

b.3) Compound

Nodo de la clase CompoundsList que guarda la información de los compuestos, tales como su nombre y fórmula química. Adicionalmente tras realizados los análisis almacena información de tiempo de elaboración y máquina que le elabora de un determinado análisis realizado.

b.5) Clase Machine

Nodo de la clase MachinesList que guarda la información de máquinas, tales como su nombre, y dimensiones (cantidad de pintes y posiciones por pin). Ésta clase almacena a el algoritmo principal del análisis de creación de compuestos.

b.6) Clase Pin

Lista simplemente enlazada cuyo primer nodo es un nodo vacío que representa la varilla. El resto de nodos son elementos de la clase Element que actúan como las posiciones de la varilla. Los pines en conjunto conforman la estructura rectangular de una máquina.

b.7) Clase Escritor

Se encarga de la escritura del código en lenguaje DOT necesario para graficar a los elementos, máquinas y compuestos. Se encarga también de escribir la salida en XML y realizar la escritura en lenguaje DOT para todas las gráficas que el usuario puede solicitar.

b.8) Clase Lector:

Se encarga de la lectura y procesamiento de la información proporcionada en el archivo de entrada, extrae los datos de elementos, máquinas y compuestos según las correspondientes etiquetas en el archivo xml leído.

b.9) Clase Configuration:

Clase que funciona como almacenamiento principal de la información de entrada. Se encarga de almacenar las listas simplemente enlazadas de elementos, máquinas y compuestos durante toda la ejecución del programa.

b.10) Clase Time

Nodo de lista TimesList que almacena un valor numérico de tiempo de correspondencia, y un listado con todas las acciones que realizan ciertos pines en dado segundo durante la creación de un cierto compuesto para una máquina.

b.11) Clase Action

Clase que almacena una acción textual (String) y un código de acción, así como el número de pin que la realiza. Éstas acciones son propias de un compuesto quien se encarga de almacenar el nombre de la máquina quien le compone y sobre la cuál son realizadas dichas acciones.

Anexos:

Figura 5. Ejemplo de gráfica de máquinas.
Fuente: Elaboración propia.

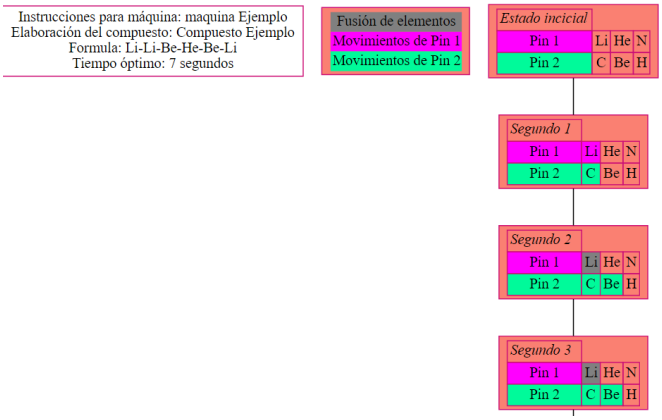


Figura 6. Ejemplo gráfica de instrucciones.
Fuente: Elaboración propia.

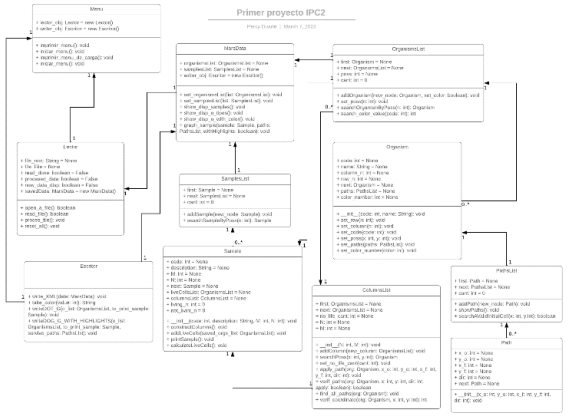
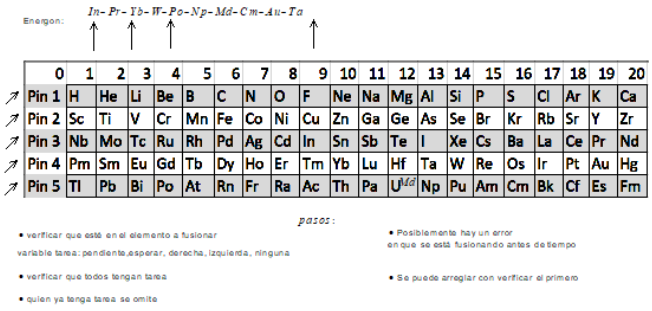


Figura 7. Diagrama de clases UML del programa.
Fuente: Elaboración propia.



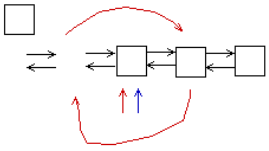


Figura 8. Análisis de la solución.
Fuente: Elaboración propia.

Referencias bibliográficas

Graphviz. *Documentation*, 2010.
Disponible en: graphviz.org

Serrano. M. *Estructura de datos*. Universidad de Valladolid. Segovia.
Disponible en: www.infor.uva.es