
PROYECTO 2

202010406 – Dennis Mauricio Corado Muñoz

Resumen

El proyecto desarrollado consiste en un algoritmo capaz de simular el proceso de un robot de ensamblaje de productos, este algoritmo fue desarrollado en el lenguaje de programación Python haciendo uso de tipos de datos abstractos, así como la librería Tkinter para brindar una interfaz grafica y la herramienta GraphViz para poder mostrar de manera grafico la secuencia de los pasos. Durante el desarrollo del proyecto se hizo una implementación de código bastante segmentada en diferentes archivos y clases buscando un mejor entendimiento, así como uso de programación orientada a objetos. La mayor parte del proyecto se desarrollo utilizando listas simplemente enlazadas y lista de listas, esto como finalidad de unir la información dispersa y poder darle un uso más óptimo.

Palabras clave

TDA = Tipo de dato abstracto

Lista Simple = simulación de un arreglo

Abstract

The developed project consists of an algorithm capable of simulating the process of a product assembly robot, this algorithm was developed in the Python programming language using abstract data types, as well as the Tkinter library to provide a graphical interface and the GraphViz tool to be able to graphically show the sequence of the steps. During the development of the project, a fairly segmented code implementation was made in different files and classes looking for a better understanding, as well as the use of object-oriented programming. Most of the project was developed using simply linked lists and lists of lists, this in order to unite the dispersed information and to be able to give it a more optimal use.

Keywords

ADT = Abstract data type

Simple List = array simulation

Introducción

El uso de TDA'S para brindar solución a un problema planteado puede ampliar la capacidad del código, debido que al implementar estructuras propias se pueden agregar funcionalidades a conveniencia. En el desarrollo del proyecto se hace uso de muchas estructuras cada una con diferente finalidad buscando brindar una solución diferente dependiendo el entorno donde fueron utilizadas y así al ser utilizadas posteriormente poseer acceso a esos métodos únicos de la clase. El uso de una herramienta grafica para mostrar la información relevante del programa es necesaria, en el desarrollo de este proyecto se utilizó la librería Tkinter.

Desarrollo del tema

Para el desarrollo del programa se hizo uso de diferentes clases con la finalidad de segmentar el código entre las clases principales tenemos:

Clase main: dentro de la clase main, como su nombre lo indico podemos encontrar los métodos principales del programa, así con la interfaz grafica que sirve para darle funciones al programa y acceder a otras clases haciendo uso de botones, etc.

Clase Administrador: Esta clase fue creada con el fin de no saturar el main y así poder tener una mejor visión de este, dentro de esta clase podemos encontrar una gran cantidad de métodos que conectan con las demás clases del programa, se podría decir que desde este archivo ocurre toda la funcionalidad del programa. Al ser instanciada la clase se instancian a su vez las demás clases, como la clase *Reader*, *Analizador*, *EscribirArchivo*, *Reporte*, se cuenta con el método 'setRoot', este método tiene como

finalidad abstraer de la clase main la ruta principal donde ejecuta la interfaz gráfica para poder agregar widgets mientras se ejecuta el programa, método 'configMaquina', este método hace uso de las clases posteriormente instanciadas, específicamente de la clase 'Reader', dentro de este método se despliega una ventana que permite seleccionar archivos y si el archivo seleccionado es valido se procede a obtener la información correspondiente, se generan avisos para mostrar si se tiene éxito o no. Método 'genSimulacion', este método al igual que el anterior se encarga de desplegar una ventana para poder seleccionar archivos, en este caso un archivo de simulación que servirá para simular el proceso del robot. Método 'datos', este método muestra los datos del estudiante que desarrollo el programa. Método 'showValues', este método despliega un mensaje que contiene los productos que puede procesar la máquina para así escribir uno en el textbox. Metodo 'ensamblarPr', este metodo sirve para poder ensamblar productos individuales, se obtiene el nombre del textbox, se verifica que exista por medio de un método que contiene la lista, y si existe se procede a procesar dicho producto. Metodo 'reporte' se encarga de generar por medio de otra clase un reporte en formato html y desplegarlo.

Clase Reader: Dentro de esta clase se analiza la información tanto del archivo que configura la maquina como del archivo que genera la simulación se extraen los datos de los archivos por medio de la librería 'ElementTree'.

Clase Analizar: dentro de esta clase se analizan las instrucciones de cada producto, se hace uso de la teoría de autómatas para poder procesar la cadena sin hacer uso de estructuras nativas de Python, en el método 'evaluarCadena' se procesa la cadena con un autómata finito determinista y al tener la información separada se envía a una lista especial que contendrá la línea y el componente en su nodo.

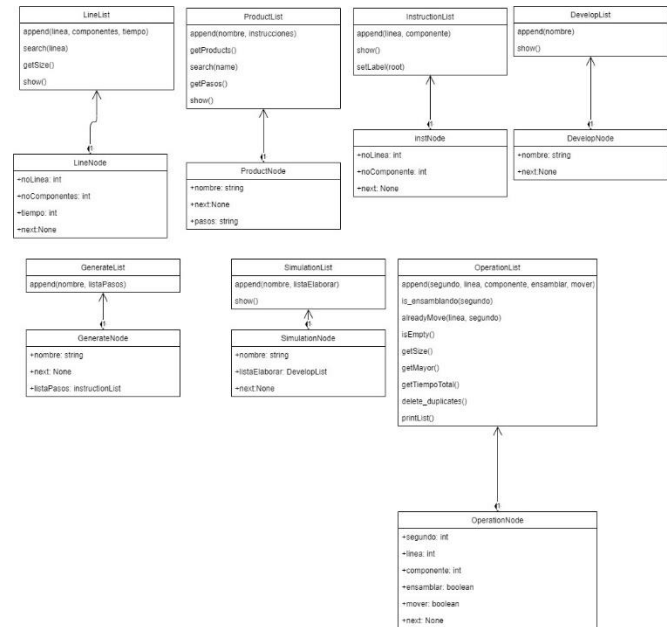
Clase Escribir Archivo: Dentro de esta clase como su nombre lo indica se escribe un archivo específicamente un archivo xml que contiene la resolución del algoritmo que brinda los pasos para poder ensamblar un producto, cuenta con el método ‘movimientos’ aca prácticamente es el algoritmo que analiza que pasos deben hacerse y en que tiempo debido a que hace paso por paso desde el tiempo 1 al n pueden existir pasos repetidos, aun con estos pasos repetidos se llena una lista

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.

Extensión: de cuatro a siete páginas como máximo



Fuente: Elaboracion Propia 2021