Universidad del Valle de Guatemala
Programación Orientada a Objetos

Gian Luca Rivera 18049
Francisco Rosal 18676

Proyecto

“Jarvis IDE”

Descripción del proyecto:

1. Párrafo introductorio:
   El proyecto trata sobre un IDE en el que por medio de reconocimiento de voz se pueden crear algoritmos. El IDE contendrá el algoritmo descrito por la persona y una sección en donde podrá ver las funciones que ha creado. Los programas son creados en código python. El proyecto se construirá con Java y estará limitado a las instrucciones de crear variables, if, while, prints, inputs, funciones, y acciones libres.
   a. Listado de features:
      i. Ver algoritmo creado: el algoritmo se creará al momento que el usuario termine de decir la instrucción vocal.
      ii. Ver funciones: el usuario podrá ver las funciones en el panel superior izquierdo del algoritmo que ha creado.
      iii. Guardar diagrama: el archivo se guardará en el lugar deseado del usuario en su computadora.
      iv. Asignación de parámetros: el usuario podrá ver en el panel inferior izquierdo los parámetros que le va a asignar a su instrucción.
   b. Herramientas que se utilizará:
      i. APIs: se hará uso de apis por las cuales se pueda complir la funcion de reconocimiento de voz, se harán varias pruebas para decidir cual usaremos en el proyecto final:
         1. Dialog Flow: https://dialogflow.com/ es un tipo de chatbot que reconoce voz para convertirlo a texto.
         2. Google Cloud: https://developers.google.com/api-client-library/java/apis/language/v1 epecificamente seria cloud speech to text api que es un api muy completo de google que reconoce hasta 120 idiomas para reconocer la voz y pasarlo a texto.
         3. wit.ai: https://wit.ai/getting-started es un tipo de api con cierto nivel de inteligencia artificial en donde aprende el lenguaje humano.
      ii. Librerías: se utilizarán librerías para poder dibujar las figuras del diagrama de flujo:
         1. canvas: https://docs.oracle.com/javase/7/docs/api/java/awt/Canvas.html representa un rectangulo blanco en la pantalla con el cual se puede dibujar

2. Graphics: https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html es una libreria de java que permite dibujar sobre la plataforma que se haya creado

2. Diseño de la experiencia de usuario: (Mockups)
   - https://marvelapp.com/9e7afh1

3. Catálogo de clases:
   a. Algoritmo
      i. Clase principal, contiene la secuencia de las acciones.
      ii. Atributos:
         1. secuencia:
            a. Lista de acciones, contiene la secuencia de acciones
            b. Tipo: lista.
      iii. Métodos:
         1. Constructor
         2. addAccion:
            a. Agrega una acción a la secuencia
            b. Firma: public void addAccion(accion)
         3. removeAccion:
            a. Elimina una acción de la secuencia
            b. Firma: public void removeAccion(accion)
         4. guardar:
            a. Guarda el algoritmo creado
            b. Firma: public
   b. Módulo
      i. Clase secundaria, para contener todas las funciones.
      ii. Atributos:
         1. funciones:
            a. Lista de funciones que tiene el programa
            b. Tipo: Lista
      iii. Métodos:
         1. Constructor
         2. addFuncion:
            a. Agrega una función nueva al modulo
            b. Firma: public void addFuncion(Funcion)
         3. removeFuncion:
            a. Elimina una función del modulo
            b. Firma: public void removeFuncion(Funcion)
   c. Input
      i. Para crear inputs
      ii. Atributos:
         1. texto: instrucción, comando de la instrucción
            a. Tipo: String
         2. valorIngresado: captura el texto que el usuario ingreso

            a. Tipo String
        iii. Métodos:
            1. Constructor
            2. setTexto: le asigna el valorIngresado al atributo texto.
                a. Firma: public void setTexto()
d. Funcion
    i. Clase para crear funciones
    ii. Atributos:
        1. nombre: es el nombre de la función.
            a. Tipo String
        2. parametros: lista de parametros que tendrá la función
            a. Tipo List<String>
        3. acciones: lista de acciones de la función
            a. Tipo Lista<Accion>
        4. retorno: retorno de la función:
            a. Tipo String
    iii. Métodos:
        1. Constructor
        2. addAccion: agrega una acción a la función
            a. Firma: public void addAccion(Accion)
        3. removeAccion: elimina una acción de la función
            a. Firma: public void removeAccion(Accion)
e. Var
    i. Clase para crear variables
    ii. Atributos:
        1. nombre: es el nombre que se le va a asignar a la variable
            a. Tipo String
        2. valor: lo que contendrá la variable
            a. Tipo String
    iii. Métodos:
        1. Constructor
        2. setValor: le asigna el valor a la variable creada
            a. firma: public void setValor(valor)
        3. getValor: captura el valor de la variable
            a. firma: public void getValor()
f. If
    i. Clase que crea verificaciones if con una condición
    ii. Atributos:
        1. condicion: condicion con la cual trabajara el if
            a. Tipo objeto Condición
        2. accionesTrue: el bloque de instrucciones que hará el if si la condición es verdadera
            a. Tipo list
        3. accionesFalse: el bloque de instrucciones que hará el if al entrar al else o cuando la condición sea falsa
            a. Tipo list

iii. Métodos:
    1. Constructor
    2. addAccion: agrega las instrucciones o acciones que van dentro del if
        a. firma: public void addAccion()
    3. removeAccion: quita una acción que ya existe dentro del if
        a. firma: public void removeAccion()

g. Condición:
    i. Clase para crear condiciones para los if y while
    ii. Atributos:
        1. variable1: Variable para comparar
            a. Tipo Var
        2. operadorLogico: Operador lógico de la condición
            a. Tipo String
        3. variable2: Variable para comparar
            a. Tipo Var
    iii. Métodos:
        1. Constructor
        2. setVariable1: asigna la variable 1
            a. Firma: public void setVariable1(Var)
        3. setOperadorLogico: asigna el operador lógico
            a. Firma: public void setOperadorLogico(String)
        4. setVariable2: asigna la variable 2
            a. Firma: public void setVariable2(Var)

h. While
    i. Clase para crear ciclos while que reciban parámetros
    ii. Atributos:
        1. condición: se le asigna la condición con la cual es ciclo podrá funcionar (solo una condición).
            a. Tipo objeto Condición
        2. acciones: las instrucciones que tendrá adentro el while
            a. Tipo List
    iii. Métodos:
        1. Constructor
        2. addAccion: le agrega una acción, instrucción al ciclo while
            a. firma: public void addAccion(accion)
        3. removeAccion: quita una acción que ya existe dentro el ciclo while
            a. firma: public void removeAccion(accion)

i. Print
    i. Para realizar salidas de texto, impresiones
    ii. Atributos
        1. texto: recibe la cadena de texto que se imprimirá
            a. Tipo String
    iii. Métodos:
        1. Constructor

2. setTexto: le asigna la cadena de texto que desea el usuario a la variable texto
    a. Firma: public void setTexto(texto)

j. Acción
    i. Servirá para poner acciones dentro del programa como sumar numeros, etc… Aún no sabemos cómo modelar esta clase.

4. Diagrama UML: