

DÍA 4 (3) CLASE LUNES 04-MAYO-2021

INICIO DE CLASE-

Realizamos repaso de la clase anterior,

1) Pensar en dos problemáticas las cuales podamos resolver aplicando tecnología (software), luego analizar y definir tres requisitos funcionales y determinar las variables y constantes que pueden existir dentro de estos requerimientos para luego desarrollar el algoritmo para cada uno de estos. (Requerimientos funcionales son acciones que realice el software)

1) Problemática resuelta:

Software que genera ficha médica, presupuesto y receta mediante voz (pensado en odontólogos)

Requerimientos funcionales:

- a) Transcribir la información dada por voz, a escrito (constante)
- b) Generar valores según procedimiento a realizar (variable)
- c) Calcular el total (variable)
- d) Generar ficha con presupuesto o receta (variable)

¿O las constantes y variables van en mi entrada?

Entrada:

Dispositivo móvil

Internet

Proceso:

- Abrir el Software
- Iniciar sesión
- Hablar (ejemplo, realizar presupuesto)
- Software inicia reconocimiento de voz
- Procede a iniciar la plantilla de presupuesto
- Hablar según lo que se requiera (ejemplo, tapadura simple pieza #3)
- Software reconoce
- Hace match de voz con procedimiento integrado

- Procede a integrar el match a la plantilla, junto con precio (que estará integrado)
- Repite procedimiento hasta que sea necesario

Salida: Generar ficha

2) Problemática resuelta:

Software visual que busca ayudar con la búsqueda de vivienda según requerimientos de cada usuario

Requerimientos funcionales:

- a) Filtrar información
- b) Búsqueda visual (maps)
- c) Guardar información filtrada (favoritos)

Entrada:

Datos usuario (variable) (chart)

Lista de requisitos “duros” (constante)

Lista de requisitos “blandos” (constante)

Buscador (constante)

Mapa interactivo (tipo maps)

Herramienta de comparación (constante)

Herramienta para guardar (constante)

Proceso:

- Crear un perfil de usuario (o más de uno si se necesita)
- Ingresar los requisitos “duros o principales” (ej: subsidio o no)
- Ingresar los requisitos “blandos o secundarios” (ej: cercano a parques, cercano a estación de metro)
- Realizar la búsqueda

- Mostrar la búsqueda como mapa interactivo (tipo maps)
- Elegir alguna de las viviendas o proyectos de interés
- Navegar a través de la vivienda, lugares cercanos, otros
- Guardar o no si se gusta
- Salir de la navegación y volver al maps principal mostrado anteriormente
- Repetir los pasos anteriores hasta que usuario determine que es suficiente
- Ir a la sección “Favoritos o Guardados”
- Poder visualizar una comparación entre ellos
- Elegir la de interés
- Presionar una tecla que te pondrá en contacto con la inmobiliaria
- Salir de la aplicación

Salida: Búsqueda de vivienda según requerimientos exitosa

Pseudocódigo: Lo que hemos estado realizando. No es realmente un lenguaje de programación, si no que ayuda al programador a razonar el programa antes de programarlo.

Como pseudoprogramar cuando estás en pañales ¿?



Estructura secuencial: Una acción (instrucción) sigue a otra en secuencia. Como fichas de dominó cayendo



Asignación: Paso de valores o resultados a una zona de memoria. Dicha zona será reconocida con el nombre de valor a una variable. Ejemplo, asignar colores a un grupo de personas.



Variable contador: Consiste en usarla como un verificador del número de veces que se realiza un proceso. Ejemplo: $a = a + 1$ ¿?

Variable acumulador: Consiste en usarla como un sumador en un proceso, acumula operaciones. Ejemplo: $a = a + b$ ¿?

Estructura algorítmica

Suma +	División /	Mayor o igual que >=	Igual que ==
Resta -	Mayor que >	Menor o igual que <=	Modulo %
Multiplicación *	Menor que <	Distinto <>	

Modulo devuelve el resto de la operación de división, ejemplo: $4\%3=1$ o $4\%2=0$
(se usa para ver si el número es par o no)

Proposición lógica: Una expresión que se puede comprobar si es verdadera o falsa. Ejemplo:
El último mes del año es Diciembre (verdadero)
Está lloviendo (falso)

Todo lo que se puede comprobar en el momento.

Sería como plantear una hipótesis, que se debe aprobar o rechazar, pero de manera inmediata o rápida (¿?) ¿Puede mi proposición (o hipótesis) demorarse más en ser comprobada? Como meses o años ¿?

Lo que no es proposición, es algo subjetivo, ejemplo: hace frío, el cielo es azul, todas las preguntas que no se puedan almacenar en verdaderos o falsos.

Operadores lógicos:

Operador lógico Y: Permite **comparar** proposiciones simples, **transformándose en una compuesta**, dicha proposición **es siempre verdadera si y sólo si todas las proposiciones lo son**, de lo contrario es falsa

El año tiene 10 meses **y** la semana 4 días, *FALSE*

El año tiene 12 meses **y** la semana 7 días, *TRUE*

Operador lógico O: Permite **comparar** proposiciones simples, **transformándose en una compuesta**, dicha proposición **es siempre verdadera si al menos se cumple alguna de las proposiciones**, y falso sólo si todas lo son.

El año tiene 10 meses **o** la semana 4 días, *FALSE*

El año tiene 10 meses **o** la semana 7 días, *TRUE*

Estructura algorítmica (funciones fundamentales)

Lectura: Consiste en recibir desde un dispositivo de entrada (teclado). Esta operación se representa en un pseudocódigo como: leer. Ejemplo Leer el resultado

Escritura: Consiste en mandar por un dispositivo de salida (pantalla) un resultado o mensaje. Este proceso se representa en un pseudocódigo como: escribir o imprimir. Ejemplo escribe o Imprime "El resultado es:", R

Inicio

Print "ingresar un número"

Leer número

Print "el número ingresado es", R

Fin

Donde "El resultado es:" es un mensaje que se desea aparezca y

R es una variable que contiene un valor.

2) Escribir algoritmo en pseudocódigo que solicite al usuario ingresar dos números

Inicio

Print "dime un número"

Leer un número

Print "el número ingresado es", A

Print "dime un segundo número2

Leer segundo número

Print "el número ingresado es", A

Print "los números son", A y B

FIN DE CLASE-

POST CLASE-

Matemática discreta

Teoría de conjunto

Lógica proposicional

Tablas de verdad

SAP

BAP

C++

CLIPPER

COBOL

FORTRAM

Pseudocódigo: Lenguaje artificial e informal que ayuda a los programadores a desarrollar algoritmo. Similar al lenguaje cotidiano, cómodo, amable. No es un lenguaje de programación. No se ejecuta en computadora. Ayuda al programador a razonar el programa antes de programarlo.

Yo: lo que hemos estado realizando

Estructura secuencial: Una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin de un proceso. Una estructura secuencial se representa: Inicio, acción1, acción2, acción3,... acciónN, fin.

Yo: como en el baile, cuando un paso te lleva a otro de manera fluida.

Se asigna un espacio de memoria a una variable

Asignación: Paso de valores o resultados a una zona de memoria. Dicha zona será reconocida con el nombre de valor a una variable.

Yo: sería como si tuviera un elenco de actores y a cada uno le asigno un rol ¿? como...

actor1: el bufón

actor2: la princesa

actor3: el rey

Asignación simple: Pasar un valor a una variable. $A=2$

Yo: lo mismo de arriba

Variable contador: Consiste en usarla como un verificador del número de veces que se realiza un proceso. Ejemplo: $a=a+1$

Variable acumulador: Consiste en usarla como un sumador en un proceso, ejemplo: $a=a+b$

Estructura algorítmica:

Operador lógico Y: Compara proposiciones

Yo: es verdadero si todas las sentencias lo son

Operador lógico O: Compara proposiciones

Yo: es verdadero si al menos una de las sentencias lo es

¡Pasamos un montónnn de materia hoy!

