Actividad Macros Unidad 2

1. VB:

Visual Basic (VB) es un lenguaje de programación desarrollado por Microsoft. Es el lenguaje de programación BASIC (Beginners' All-purpose Symbolic Instruction Code).

VB es conocido por su facilidad de uso y es utilizado por principiantes en programación. Permite a los desarrolladores crear interfaces gráficas de usuario (GUI) y se utiliza comúnmente para desarrollar aplicaciones de Windows.

VB admite una amplia gama de características, incluida la programación orientada a objetos, la programación orientada por eventos y la conectividad a bases de datos. También cuenta con un conjunto robusto de funciones y controles incorporados que facilitan la creación de aplicaciones interactivas.

VB se puede utilizar para crear una variedad de aplicaciones, como aplicaciones de escritorio, aplicaciones web y aplicaciones móviles. en conjunto con tecnologías como el marco .NET y SQL Server.

En general, Visual Basic es un lenguaje de programación versátil que es adecuado tanto para principiantes como para desarrolladores experimentados.

VBA:

VBA (Visual Basic for Applications) es una implementación de Visual Basic que se utiliza principalmente en aplicaciones de Microsoft Office, como Excel, Word y Access. Es un lenguaje de programación específico de la aplicación (ASP), lo que significa que está integrado en las aplicaciones para automatizar tareas y personalizar su funcionamiento.

VBA permite a los usuarios crear macros, que son secuencias de comandos que realizan acciones automatizadas en una aplicación. Estos macros pueden ser utilizados para realizar tareas repetitivas, manipular y analizar datos, generar informes y más.

VBA se basa en el lenguaje de programación Visual Basic y comparte muchas de sus características y estructuras de programación, como variables, bucles, condiciones y funciones. También tiene acceso a los objetos y métodos de la aplicación host, lo que permite interactuar con ellos y manipular su funcionamiento.

Una ventaja de VBA es que ofrece una interfaz de programación sencilla y familiar para aquellos que ya están familiarizados con Visual Basic. Además, al estar integrado en las aplicaciones de Office, los desarrolladores pueden automatizar tareas y realizar personalizaciones sin tener que utilizar un entorno de desarrollo separado.

En resumen, VBA es una poderosa herramienta de automatización y personalización que permite a los usuarios programar y automatizar acciones en las aplicaciones de Microsoft Office.

Diferencias entre VB y VBA:

Existen algunas diferencias clave entre VB (Visual Basic) y VBA (Visual Basic for Applications):

1. Aplicación y contexto de uso: VB es un lenguaje de programación independiente que se utiliza para desarrollar aplicaciones autónomas que no están específicamente integradas en otras aplicaciones. Por otro lado, VBA es una implementación de Visual Basic utilizada principalmente para automatizar y personalizar aplicaciones específicas, como Microsoft Office (Excel, Word, Access, etc.).
2. Entornos de desarrollo: VB viene con su propio entorno de desarrollo integrado (IDE), que proporciona herramientas y funciones para escribir, depurar y compilar código VB. VBA, por otro lado, se integra directamente en las aplicaciones de Office existentes, por lo que puede usar el Editor de Visual Basic, que está integrado en cada aplicación de Office.
3. Alcance de acceso a objetos y funciones: VB puede acceder a una amplia gama de bibliotecas de funciones y objetos, lo que le permite desarrollar aplicaciones independientes con múltiples funcionalidades. VBA, por otro lado, tiene acceso limitado a los objetos y funciones específicos de la aplicación host. Esto significa que puede interactuar con los objetos y funciones de la aplicación de Office, pero no tiene acceso a funciones más generales o a objetos y funciones de otros entornos.
4. Objetos y eventos específicos de la aplicación: VBA se enfoca en proporcionar acceso a los objetos y eventos específicos de la aplicación host. Por ejemplo, en Excel, puede usar VBA para manipular celdas, crear gráficos y automatizar la generación de informes en Excel. Estos objetos y eventos son específicos de Excel y no están disponibles en VB puro.

En resumen, VB es un lenguaje de programación independiente que se utiliza para desarrollar aplicaciones autónomas, mientras que VBA es una implementación específica de Visual Basic utilizada para automatizar y personalizar aplicaciones específicas, como las aplicaciones de Microsoft Office. VBA se encuentra estrechamente integrado en estas aplicaciones y ofrece acceso a objetos y eventos específicos de la aplicación host.

1. POO:

La Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en el concepto de "objetos". En POO, los objetos son entidades que encapsulan datos y funciones relacionadas, y se utilizan para representar elementos del mundo real o conceptos abstractos.

Las principales características de la POO son:

1. Abstracción: Permite enfocarse en las características y comportamientos esenciales de un objeto, ignorando los detalles irrelevantes.
2. Encapsulación: Consiste en agrupar datos y funciones en un solo objeto. Los datos están ocultos al mundo exterior y solo pueden ser accedidos y manipulados a través de métodos o funciones específicas.
3. Herencia: Permite que un objeto herede características y comportamientos de otro objeto existente (clase padre). Esto facilita la reutilización de código y la creación de jerarquías de objetos.
4. Polimorfismo: Se refiere a la capacidad de un objeto de comportarse de diferentes formas según el contexto. Mismo método puede tener diferentes implementaciones en diferentes clases.
5. Clases: Las clases son la base de la POO y se utilizan para crear objetos. Una clase es una plantilla o plano que define las características y comportamientos que tendrán los objetos de esa clase.

La POO ofrece varios beneficios, como la modularidad, la reutilización de código, la organización estructurada del código y la facilidad para mantener y ampliar los sistemas. Se utiliza en una amplia gama de lenguajes de programación, como Java, C++, C#, Python, entre otros.

otros.

Ejemplos de poo:

class Persona:

def \_\_init\_\_(self, nombre, edad, genero):

self.nombre = nombre

self.edad = edad

self.genero = genero

def saludar(self):

if self.genero == "M":

print(f"Hola, mi nombre es {self.nombre} y soy un hombre.")

elif self.genero == "F":

print(f"Hola, mi nombre es {self.nombre} y soy una mujer.")

else:

print(f"Hola, mi nombre es {self.nombre} y no identifico un género.")

def cumplir\_anios(self):

self.edad += 1

# Crear una instancia de la clase Persona

persona1 = Persona("Juan", 30, "M")

persona1.saludar() # Output: Hola, mi nombre es Juan y soy un hombre.

print(persona1.edad) # Output: 30

# Modificar atributos de la instancia

persona1.edad = 31

print(persona1.edad) # Output: 31

# Llamar a métodos de la instancia

persona1.cumplir\_anios()

print(persona1.edad) # Output: 32

En este ejemplo, la clase Persona tiene tres atributos: nombre, edad y genero. También tiene dos métodos: saludar(), que muestra un saludo personalizado según el género de la persona, y cumplir\_anios(), que incrementa en uno la edad de la persona.

Luego se crea una instancia de la clase Persona llamada persona1, se modifican sus atributos y se llaman a sus métodos.



|  |  |  |  |
| --- | --- | --- | --- |
| Objeto | Propiedad | Valor | Significado |
| Textbox | Name | txtnombre | Nombre del objeto |
|  | Font | Elegida por el usuario |  |
|  | ForeColor | Elegida por el usuario | Color de fuente |
|  | Backcolor | Elegida por el usuario | Color de fondo |
|  | Text |  | Contenido |
|  |  |  |  |
| Command Button | Name | Btnmostrar |  |
|  | Caption | Mostrar mensaje |  |
|  | Font | Elegido por el usuario |  |
|  |  |  |  |
| Label | Name | Etimensaje |  |
|  | Alingment | Centrado | Alineación |
|  | Font | Elegido por el usuario |  |
|  | ForeColor | Elegido por el usuario |  |
|  | BackColor | Elegido por el usuario |  |
|  | Caption |  |  |
|  |  |  |  |
| Form | Name | Frmprimer |  |
|  | Caption | Primer programa |  |
|  | BackColor | Elegido por el usuario |  |

1. Los tipos de objetos que se emplean comúnmente en VBA son:

Controles: Etiqueta, cuadro de texto, botón de comando, etc.

Contenedores: Formularios, marcos, página múltiple, etc.

Los controles en VBA (Visual Basic for Applications) se utilizan para interactuar con los usuarios a través de formularios y cuadros de diálogo en aplicaciones de Microsoft Office, como Excel, Word, PowerPoint, etc. Estos controles te permiten agregar botones, cuadros de texto, listas desplegables y muchos otros elementos interactivos a tus aplicaciones VBA.

Aquí tienes algunos ejemplos de controles comunes en VBA:

1. Botón de comando (Command Button):

Private Sub CommandButton1\_Click()

MsgBox "¡Hola, mundo!"

End Sub

1. Cuadro de texto (Text Box):

Private Sub TextBox1\_Change()

Dim texto As String

texto = TextBox1.Text

MsgBox texto

End Sub

1. Lista desplegable (Combo Box):

Private Sub ComboBox1\_Change()

Dim seleccion As String

seleccion = ComboBox1.Value

MsgBox seleccion

End Sub

1. Casilla de verificación (Check Box):

Private Sub CheckBox1\_Click()

If CheckBox1.Value = True Then

MsgBox "Casilla marcada"

Else

MsgBox "Casilla desmarcada"

End If

End Sub

Estos son solo algunos ejemplos de controles que puedes utilizar en VBA. Puedes adaptarlos y personalizarlos según tus necesidades. Los controles son muy útiles para crear interfaces de usuario interactivas y mejorar la experiencia del usuario en tus aplicaciones VBA.

En VBA (Visual Basic for Applications), los contenedores son objetos que permiten agrupar otros objetos dentro de ellos. Estos contenedores pueden organizar y controlar la disposición de los objetos dentro de un formulario o una hoja de cálculo, por ejemplo. Algunos ejemplos comunes de contenedores en VBA son:

1. Cuadro de grupo (Frame): El cuadro de grupo es un contenedor visual que permite agrupar objetos relacionados en un formulario. Ayuda a organizar y separar visualmente diferentes secciones de la interfaz del usuario. Puedes agregar objetos como botones, cuadros de texto, etiquetas, etc., dentro del cuadro de grupo.
2. Panel (Panel): El panel es un contenedor liviano que se utiliza para agrupar varios controles en una hoja de cálculo o un formulario. Puedes agregar objetos como botones, cuadros de texto, etiquetas, etc., dentro del panel.
3. Propiedad MultiPage: El control MultiPage es un contenedor que puede contener varias páginas, donde cada página puede tener sus propios controles. Permite crear pestañas o pestañas que contienen diferentes conjuntos de controles relacionados. Es útil para organizar y presentar información de manera estructurada.
4. Cuadro de lista (ListBox): Aunque no es un contenedor en sí mismo, el cuadro de lista se puede utilizar como un contenedor para almacenar y mostrar una lista de elementos. Puedes agregar y eliminar elementos en el cuadro de lista, y los usuarios pueden seleccionar elementos de la lista.

Estos son algunos ejemplos comunes de contenedores en VBA que te ayudarán a organizar y controlar la presentación de elementos dentro de formularios, hojas de cálculo u otros objetos en tus aplicaciones VBA.