Go COM Automation Client

Steps

1. Download and install the latest version of Go (current 1.25.0) - Link
2. Create a directory named as `go-automation-client`
3. In that directory, create a file named as `GoAutomationClient.go`
4. Run the following commands in that directory to setup the project and dependencies

```
go mod init go-automation-client
```

```
go get github.com/go-ole/go-ole
```

5. Type the source code and replace **CLSID** with your **MyMath's CLSID**
6. Run the code using the command

```
go run Client.go
```

Documentation for go-ole package : go-ole

```go
package main

import (
    "fmt"

    ole "github.com/go-ole/go-ole"
)

func main() {

    // Variable Declarations
    var num1 int
    var num2 int

    // Code
    fmt.Print("\nEnter Number 1 = ")
    fmt.Scan(&num1)

    fmt.Print("\nEnter Number 2 = ")
    fmt.Scan(&num2)

    ole.CoInitialize(0)
    defer ole.CoUninitialize()

    CLSID_MyMath := "{69DEFCBC-00E1-4817-A8D1-C608EE5C20F3}"

    clsid_ptr, err := ole.CLSIDFromString(CLSID_MyMath)
    if err != nil {
        fmt.Println("Error Occurred When Retrieving Co-Class With Provided CLSID
!!!")
        return
    }

    iUnknown_ptr, err := ole.CreateInstance(clsid_ptr, ole.IID_IUnknown)
    if err != nil {
        fmt.Println("Failed To Obtain Implemented IID_IUnknown Interface !!!")
        return
    }
    defer iUnknown_ptr.Release()

    dispatch_ptr, err := iUnknown_ptr.QueryInterface(ole.IID_IDispatch)
    if err != nil {
        fmt.Println("Failed To Obtain Implemented IID_IDispatch Interface !!!")
        return
    }
    defer dispatch_ptr.Release()

    var dispid []int32
    dispid, err = dispatch_ptr.GetIDsOfName([]string{"SumOfTwoIntegers"})
    if err != nil {
        fmt.Println("Failed To Obtain ID For SumOfTwoIntegers() !!!")
        return
```

```
    }

    var result *ole.VARIANT
    result, err = dispatch_ptr.Invoke(dispid[0], ole.DISPATCH_METHOD, num1, num2)
    if err != nil {
        fmt.Println("Failed To Obtain Result For SumOfTwoIntegers() !!!")
        return
    }

    sum := result.Value()

    dispid, err = dispatch_ptr.GetIDsOfName([]string{"SubtractionOfTwoIntegers"})
    if err != nil {
        fmt.Println("Failed To Obtain ID For SubtractionOfTwoIntegers() !!!")
        return
    }

    result, err = dispatch_ptr.Invoke(dispid[0], ole.DISPATCH_METHOD, num1, num2)
    if err != nil {
        fmt.Println("Failed To Obtain Result For SubtractionOfTwoIntegers() !!!")
        return
    }

    subtract := result.Value()

    fmt.Printf("\nSum of %d and %d = %d\n", num1, num2, sum)
    fmt.Printf("\nSubtraction of %d and %d = %d\n", num1, num2, subtract)

}
```