

21.04.2020

ОБЪЕДИНЕНИЕ «ПРОГРАММИРОВАНИЕ»,

2-й год обучения

Тема:

«Процедуры и функции в Visual Basic»

Цели:

1. Ввести понятия процедуры (подпрограммы и функции) пользователя.
2. Научить учащихся использовать процедуры в проектах.
3. Развивать у учащихся способность к алгоритмическому мышлению
4. Развивать умение анализировать взаимосвязь между различными объектами проекта.
5. Формировать устойчивое внимание

Теоретические сведения

Работа с процедурами

Visual Basic — *процедурный программирования язык* Это означает, что в нем можно создавать блоки программного кода, на которые затем можно ссылаться по имени. После того как блок кода получит имя, он может быть *вызван* и выполнен. Это похоже на программу в программе. Маленькие программы, "живущие" в больших программах, называются *функциями*, если они возвращают какое-либо значение, и *подпрограммами*, если они значений не возвращают.

Подпрограммы и функции делают процесс программирования более простым и быстрым, а создаваемый код — более надежным. Создание собственных подпрограмм и функций — первый шаг к разработке *инкапсулированного* и повторно используемого кода. Под инкапсуляцией следует понимать сокрытие реализации свойств и методов объекта за его внешним интерфейсом.

Подпрограмма (Sub) - представляет собой процедуру, выполняющую программный код в пределах своего блока и не возвращающую значения.

Синтаксис простой подпрограммы таков:

(Private | Public) Sub SubMain ()

..строки кода End Sub

Где:

- [Private Public] —область видимости подпрограммы;
- Sub —тип процедуры (а именно — подпрограмма);
- *subMain* имя, присваиваемое подпрограмме;
- End Sub —окончание блока кода подпрограммы.

Функция (Function) - это процедура, которая выполняет строки своего кода и возвращает некоторое значение. Синтаксис простой функции таков:

[Private I Public] Function *FunctionName*() As *Datatype*

...строки кода

***FunctionName* = *ReturnValue* End Function**

где:

- [Private Public] —область видимости функции;
- Function — ключевое слово Visual Basic, указывающее на то, что это именно функция;
- *FunctionName* () — имя, присваиваемое функции;
- AS — ключевое слово Visual Basic, предваряющее назначение типа данных;
- *DataType* тип данных возвращаемого значения;
- *ReturnValue* значение, которое должно быть присвоено имени функции (это очень важный момент!);
- End Function —конец данного блока кода.

Практическое задание:

- I. Создание проекта для определения заработной платы менеджера фирмы по новой системе оплаты.**
 - а. Установить следующую форму:**

Бухгалтерия

**Расчет заработной платы
менеджера фирмы**

Количество отработанных часов	<input type="text"/>	
расценка одного часа	<input type="text"/>	
Стаж работы в фирме	<input type="text"/>	число
Премияльные доплаты	<input type="text"/>	%

Итого	<input type="text"/>	
-------	----------------------	--

Вычислить Перейти к другой форме

2. Набрать следующий программный код:

Public Oklad_itogo As Currency

```
Private Sub cmdА÷ёёёё_Click() 'Ïöäöäöä êñáíñé êñíèè Ä÷ёёёё
Kolic = CSng(Text1(0).Text)
Ctoim = CSng(Text2.Text)
Stag = CSng(Text3.Text)
Premium = CSng(Text4.Text)
Oklad_itogo = Baza(Kolic, Ctoim, Stag) + Baza(Kolic, Ctoim, Stag) *
Premium / 100
Text5.Text = FormatNumber(Oklad_itogo, 2)
End Sub
```

Private Function Baza(Kolic, Ctoim, Stag) As Currency

Select Case Stag

Case Is >= 1

```

    Baza = (Kolich * Ctoim) * 1.25
Case Is >= 5
    Baza = (Kolich * Ctoim) * 2.1
Case Is >= 7
    Baza = (Kolich * Ctoim) * 2.5
    Case Is >= 10
        Baza = (Kolich * Ctoim) * 3.5
Case Else
    Baza = Kolich * Ctoim
End Select
End Function

```

```

Private Sub Command2_Click()
    Form2.Show
    Unload Me
End Sub

```

```

Private Sub Form_Load()
End Sub

```

Для выполнения проекта щелкнуть по кнопке **Start** панели инструментов

V. Самостоятельная работа

1. Добавить вторую форму и создать программный код перехода на эту форму для кнопки **«переход на вторую форму»**.
2. Установить на второй форме кнопку перехода на первую форму.(пример программного кода):.

```

Private Sub Command2_Click()
    Form2.Show
    Unload Me
End Sub

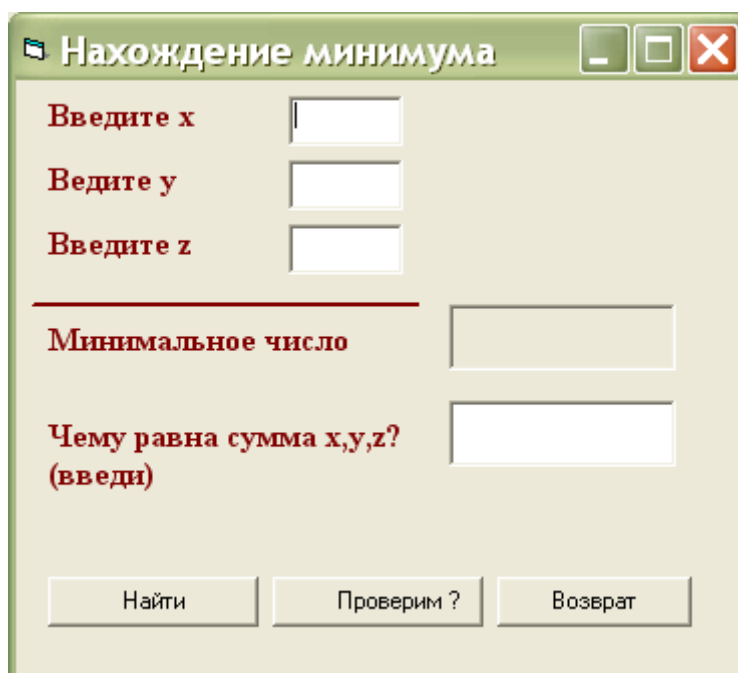
```

3. Установить на второй форме элементы для решения следующей задачи: Создать проект для проверки знаний устного сложения и нахождения минимального значения из трех введенных значений.

В программном коде должна быть использована подпрограмма нахождения минимального значения и подпрограмма проверки умения устного счета.

Значения чисел требуется вводить через текстовые поля TextBox. Вывод минимального числа в поле типа Label, а значение суммы в текстовое поле TextBox.

Установить элементы на форме, как показано на примере:



4. Предлагаемый программный код:

```
Private x As Single
```

```
Private y As Single
```

```
Private z As Single
```

```
Private min As Variant
```

```
Private Sub Command1_Click()
```

```
    Call Minimum
```

```
    Label5.Caption = min
```

```
End Sub
```

```
Public Sub Minimum()
```

```
    Data
```

```
    If x = y And x = z Then
```

```

    min = "Числа равны!"
ElseIf x <= y And x <= z Then
    min = x
ElseIf y <= x And y <= z Then
    min = y
ElseIf z <= x And z <= y Then
    min = z
End If
End Sub

Private Sub Command2_Click()
    Data
    Sum = x + y + z
    If CSng(Text4.Text) = Sum Then

        MsgBox "Вы гений! Получите премию!"
    Else

        MsgBox "А подумать!"
    End If

End Sub

```

```

Private Sub Data()
    x = CSng(Text1.Text)
    y = CSng(Text2.Text)
    z = CSng(Text3.Text)
End Sub

```