

VB
(Visual Basic)

VBA
**(Visual Basic for
Application)**

VBScript



VBA это общее средство программирования для всего семейства **MSOFFICE** . Этот язык является произвольным от самостоятельной системы программирование **Visual Basic (VB)**. Приложения разрабатываемые на **VBA** могут выполняться только в среде , поддерживающей **VBA** , а **VB** ориентирован на разработку автономных приложений. Их синтаксис и интерфейс практически одинаков.

VBScript предназначен для работы в Internet



Общие сведения о VBA

VBA - это версия визуального средства для создания приложений. VBA является **объектно-ориентированным языком программирования**. Ключевой идеей объектно-ориентированного программирования является объединение данных и используемых для их обработки функций в один объект



Основные особенности VBA

- **Наличие объектов** (рабочие книги, рабочие листы, формы, и пр.). Можно сказать, что **объект - это то, чем мы управляем с помощью программы на VBA.** При конструировании объектов используются более мелкие объекты - элементы управления (кнопки, поля, надписи на форме и пр.). С другой стороны объекты объединяются в более крупные объекты - семейства (семейство полей таблицы, семейство таблиц БД и пр.). Таким образом **семейство может состоять из объектов, а объект может содержать семейство.**



- Все объекты имеют сохраняемый набор **СВОЙСТВ** (характеристик), изменяя которые можно управлять объектом. Ссылка на свойство объекта дается в программе в виде

**<имя объекта>.<свойство> =
значение свойства**



Примеры

Надпись на кнопке

```
CommandButton1.Caption = "Редактировать"
```

Переименовать рабочий лист

```
Worksheets(1).Name="Итоги"
```

Сделать лист невидимым

```
Worksheets(2).Visible=False
```

Изменить подпись приложения

```
Application.Caption="Пример"
```



- С каждым объектом связывается ряд **МЕТОДОВ** (действий или процедур над объектом). Один и тот же метод может использоваться для обработки различных объектов. Указание метода в программе дается в виде

-

<имя объекта>.<метод>



Примеры

Добавить рабочий лист

WorkSheets(10).Add

Удалить рабочий лист

Worksheet(3).Delete

Удалить строку

WorkSheets(1).Rows(3).Delete



- **Программа на VBA в отличие от традиционных систем программирования не требует запуска.** Ее выполнение инициируется наступлением какого-либо события (открытие формы) или действия пользователя (1с по кнопке).

- **В приложении нет управляющей программы, а есть набор процедур обработки событий. Порядок выполнения процедур зависит от последовательности возникновения событий. Событие действие, распознаваемое объектом**

-

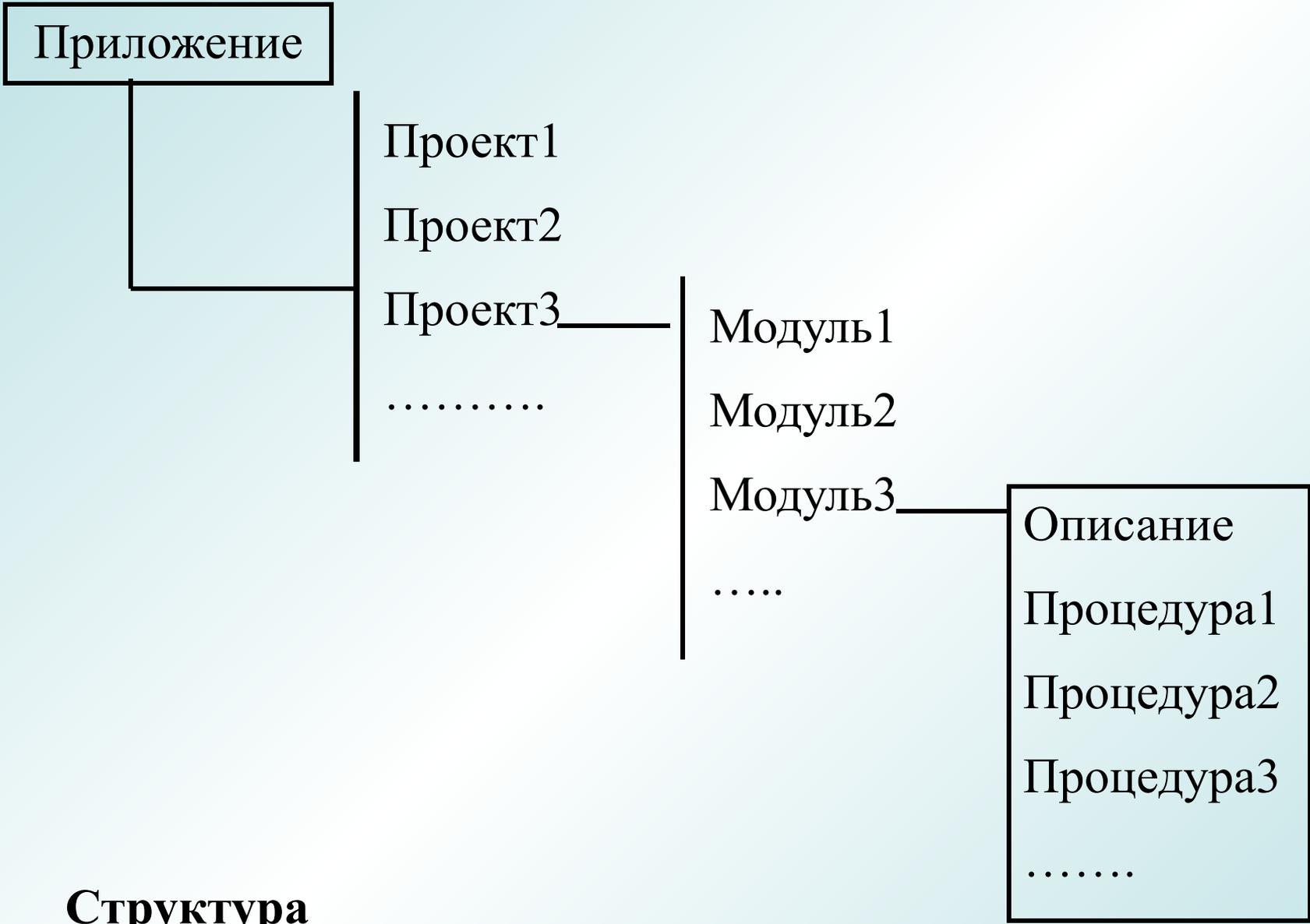


В простых приложениях применение программ VBA ограничивается созданием процедур обработки событий и простых процедур – функций, используемых в приложениях.



- Как правило программа на VBA состоит из **нескольких подпрограмм и функций**. Описание этих программных единиц хранится в модулях. Модули входят составными частями в проекты. Проекты хранятся в файлах документов приложений.





**Структура
программы**



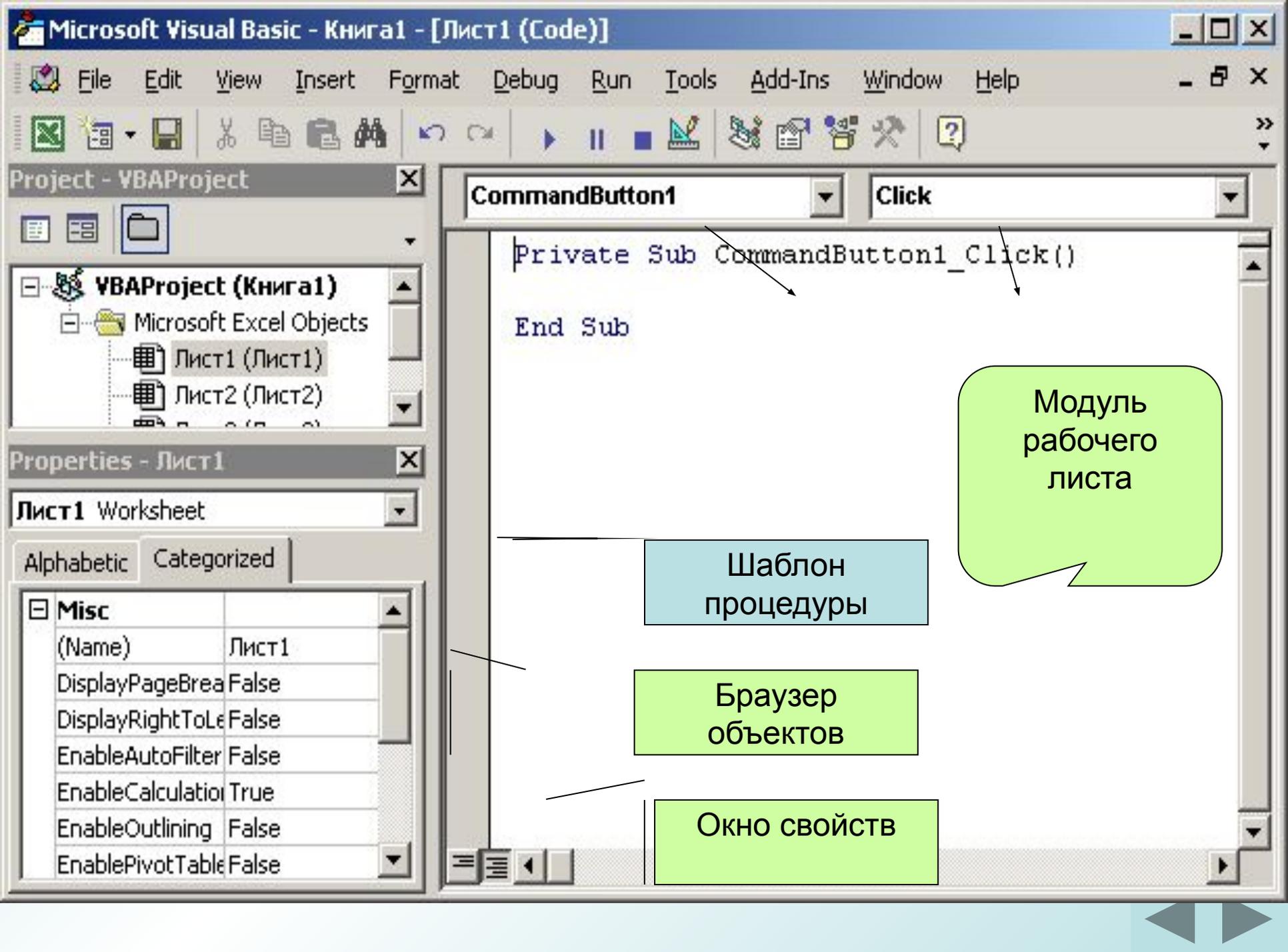
```
graph TD; A[Модуль] --- B[Стандартный]; A --- C[Класса или объекта];
```

Модуль

Стандартный

Класса или
объекта





Project - VBAProject

VBAProject (Книга1)

- Microsoft Excel Objects
 - Лист1 (Лист1)
 - Лист2 (Лист2)

Properties - Лист1

Лист1 Worksheet

Alphabetic Categorized

Misc	
(Name)	Лист1
DisplayPageBreak	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableOutlining	False
EnablePivotTable	False

CommandButton1 Click

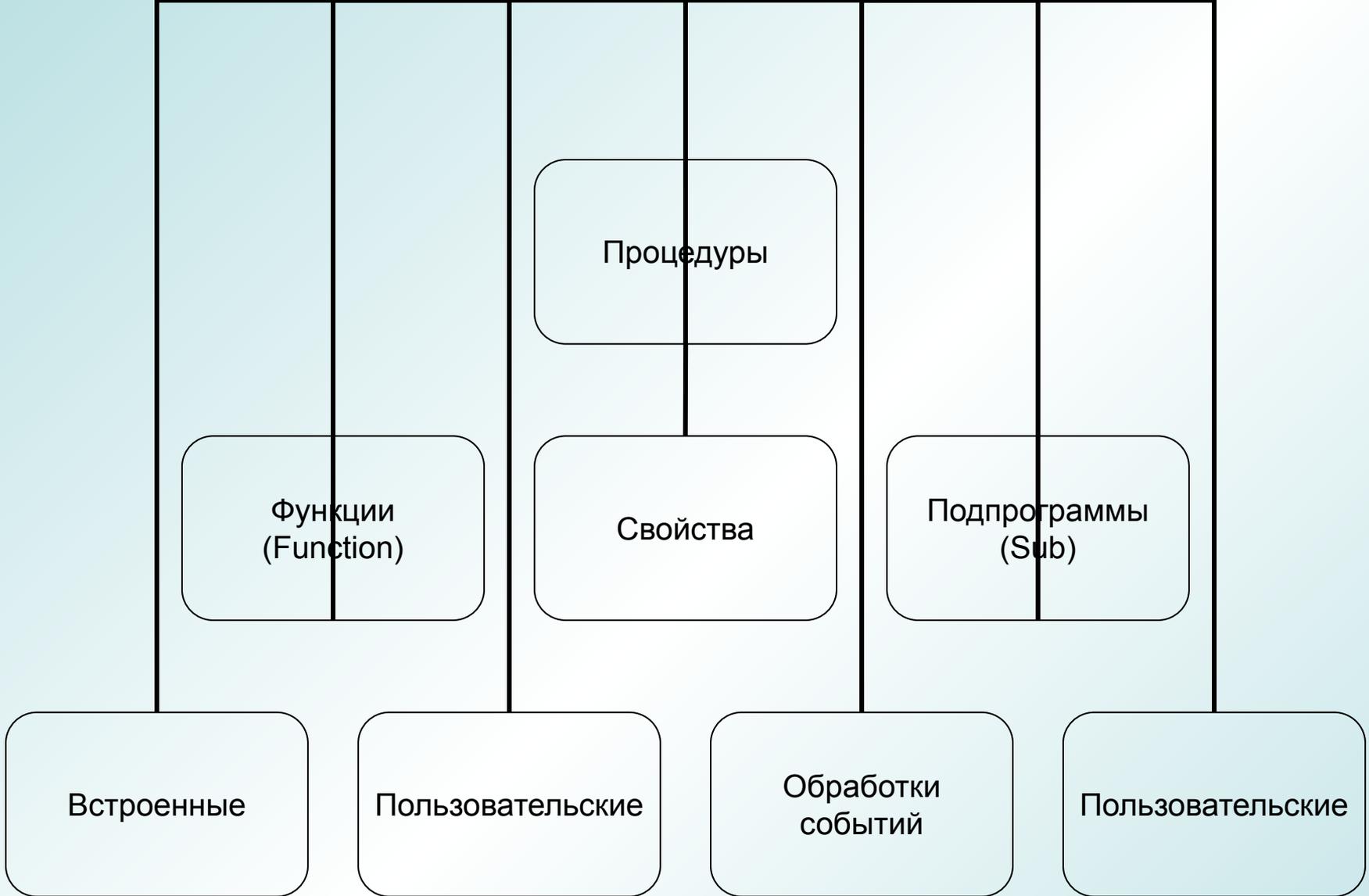
```
Private Sub CommandButton1_Click()  
  
End Sub
```

Модуль рабочего листа

Шаблон процедуры

Браузер объектов

Окно свойств



Структура процедуры

Sub(Function) Имя([параметры])

Описание переменных (Dim, Static)

<Оператор1>

<Оператор2>

.....

[exit Sub(Function)] досрочный выход

.....

<ОператорN>

End Sub(Function)



Описание переменных

Dim <имя переменной>As <тип данных>

Dim I As Integer, J As Integer

Dim Фамилия As String

Некорректное описание

Dim X,Y,Z As Integer



Область определения переменных

Public-общая

Private-только из своего модуля

Dim –внутри процедуры

Static – только внутри
процедуры, но при выходе из
процедуры значение
переменной сохраняется



Приложение

Public

Проект

Модуль

Private, Dim

Процедура

Dim

Static



Byte	Число от 0 до 255
Boolean	True или False
Integer	Число от -32768 до 32767
Long	Длинное целое: число от -2147483648 до 2147483647
Single	С плавающей точкой, одинарная точность:
Double	С плавающей точкой, двойная точность
Currency	Денежный
Date	Дата
String	Текстовый
Variant	

Для преобразования одних типов данных в другие можно использовать встроенные функции **CStr**, **Cint**, **CLng**, **CDate**



Константы

- Константа всегда имеет постоянное значение.
- При программировании на VBA используются следующие типы констант
 - Встроенные (**VbOk**, **VbNo**, **VbCancel**) (содержится в библиотеке типов)
 - Системные (True, False, Null)
 - Пользовательские

Пример описания пользовательской константы

```
Const MyCon As Integer=100
```



Примеры встроенных функций

Ввод

InputDialog(сообщение[,заголовок] [,значение по умолчанию])

Вывод

MsgBox(сообщение [,кнопки] [,заголовок])



Встроенные функции VBA

Стандартные диалоговые окна

Окно сообщения

MsgBox сообщение, атрибуты, заголовок

Пример

Msgbox “Good morning”



Окно сообщения

MsgBox сообщение, атрибуты заголовков

При вводе функции появляется подсказка,



позволяющая выбирать параметры из списка или вводить их вручную



Окно сообщения

MsgBox сообщение, атрибуты, заголовок

Второй параметр позволяет задать вид окна
Msgbox “Good morning”,vbInformation



Окно сообщения

MsgBox сообщение, атрибуты, заголовок

Второй параметр позволяет задать вид окна

Msgbox “Good morning”, vbExclamation



Окно сообщения

MsgBox сообщение, атрибуты, заголовок

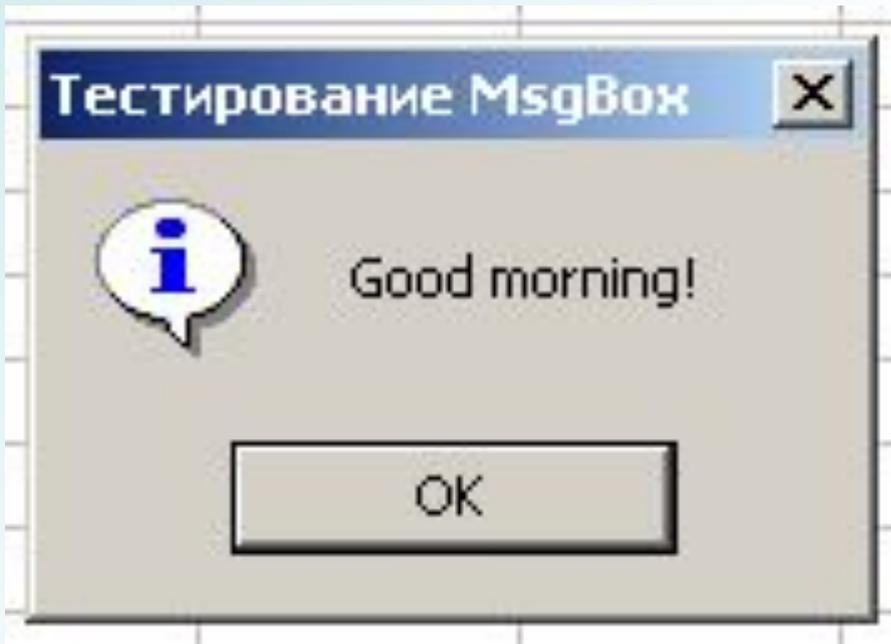
Второй параметр позволяет задать вид окна
Msgbox “Good morning”,vbCritical



Окно сообщения

MsgBox сообщение, атрибуты, **заголовок**

Третий параметр позволяет задать заголовок окна
Msgbox “Good morning”,vbInformation,”Тестирование MsgBox”



Окно сообщения может также вызываться при помощи **функции** – возвращаемое значение зависит от кнопки, нажатой пользователем. Синтаксис функции совпадает

Dim rc As Integer ‘ **код возврата**

Rc = MsgBox (сообщение, атрибуты, заголовок)

Обратите внимание на скобки.

Константа	Значение	Нажатая кнопка
VbOK	1	OK
VbCancel	2	Cancel
VbAbort	3	Abort
VbRetry	4	Retry
VbIgnore	5	Ignore
VbYes	6	Yes
VbNo	7	No



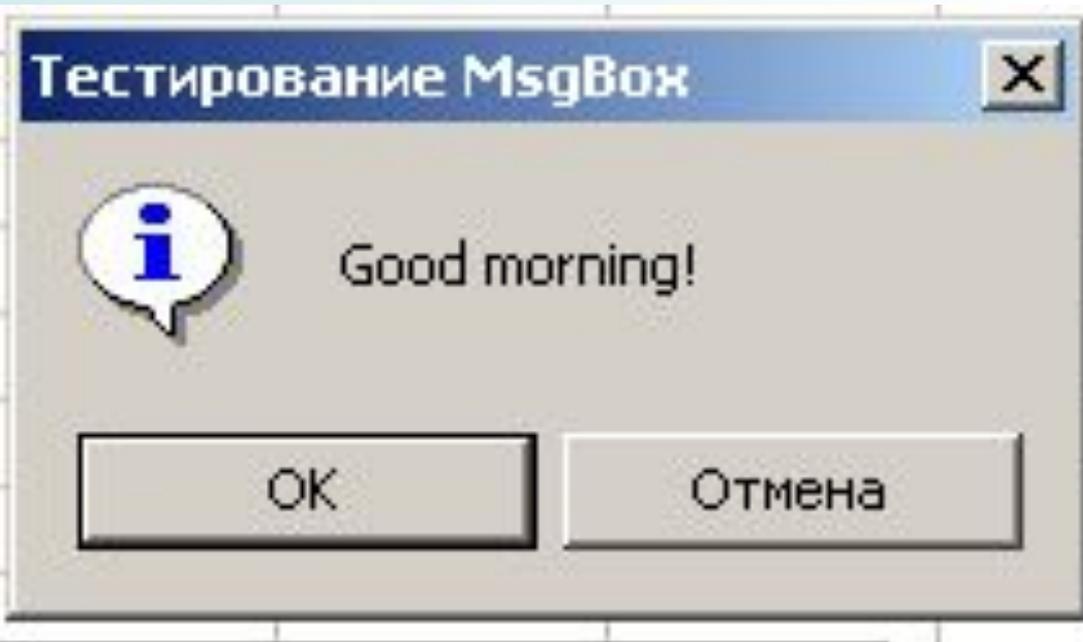
Функция

Rs =MsgBox (сообщение, атрибуты, заголовок)

**RC=(MsgBox "Good morning!", vbInformation + vbOKCancel, _
" Тестирование MsgBox")**

(Знак “_” используется для переноса на следующую строку)

Значения констант можно складывать для достижения требуемого результата



Пример программы

```
Private Sub CommandButton1_Click()  
Dim Rc As Integer  
Rc = MsgBox("Good morning!", vbInformation +  
    vbOKCancel, "Тестирование MsgBox")  
If Rc = vbOk Then  
    MsgBox "Отлично! Продолжим работу"  
Else  
    MsgBox "Увы! До скорой встречи"  
End If  
  
End Sub
```



Окно Ввода

```
Dim Rc As Integer
```

```
Rc = InputBox(Сообщение, заголовок,  
             значение по умолчанию)
```

Пример

```
Rc = InputBox("Введите имя")
```



Окно Ввода

`InputBox(Сообщение, заголовок, значение по умолчанию)`

Пример

```
Rs = InputBox("Введите имя", "Знакомство")
```



Окно Ввода

`InputBox(Сообщение, заголовок, значение по умолчанию)`

Пример

```
Rs = InputBox("Введите имя", "Знакомство", "Наташа")
```



Пример программы

```
Private Sub CommandButton1_Click()  
Dim NameS As String, Surname As String  
Dim Student As String  
Dim rc As Integer  
rc = MsgBox("Good morning!", vbInformation + vbOKCancel, "Testing")  
If rc = VbCancel Then  
    MsgBox "Увы! До скорой встречи"  
    Exit Sub  
End If  
MsgBox "Отлично! Продолжаем работу"  
NameS = InputBox("Введите Ваше Имя", "Знакомство", "Маша")  
Surname = InputBox("Введите Вашу фамилию", "Знакомство", "Петрова")  
Student = Surname & " " & NameS  
rc = MsgBox("Рады знакомству" & Student, vbExclamation + vbOKOnly,  
    "Testing")  
End Sub
```

Описание
переменных

Ветвление

Выход из
процедуры

Начало и конец процедуры



Пример программы

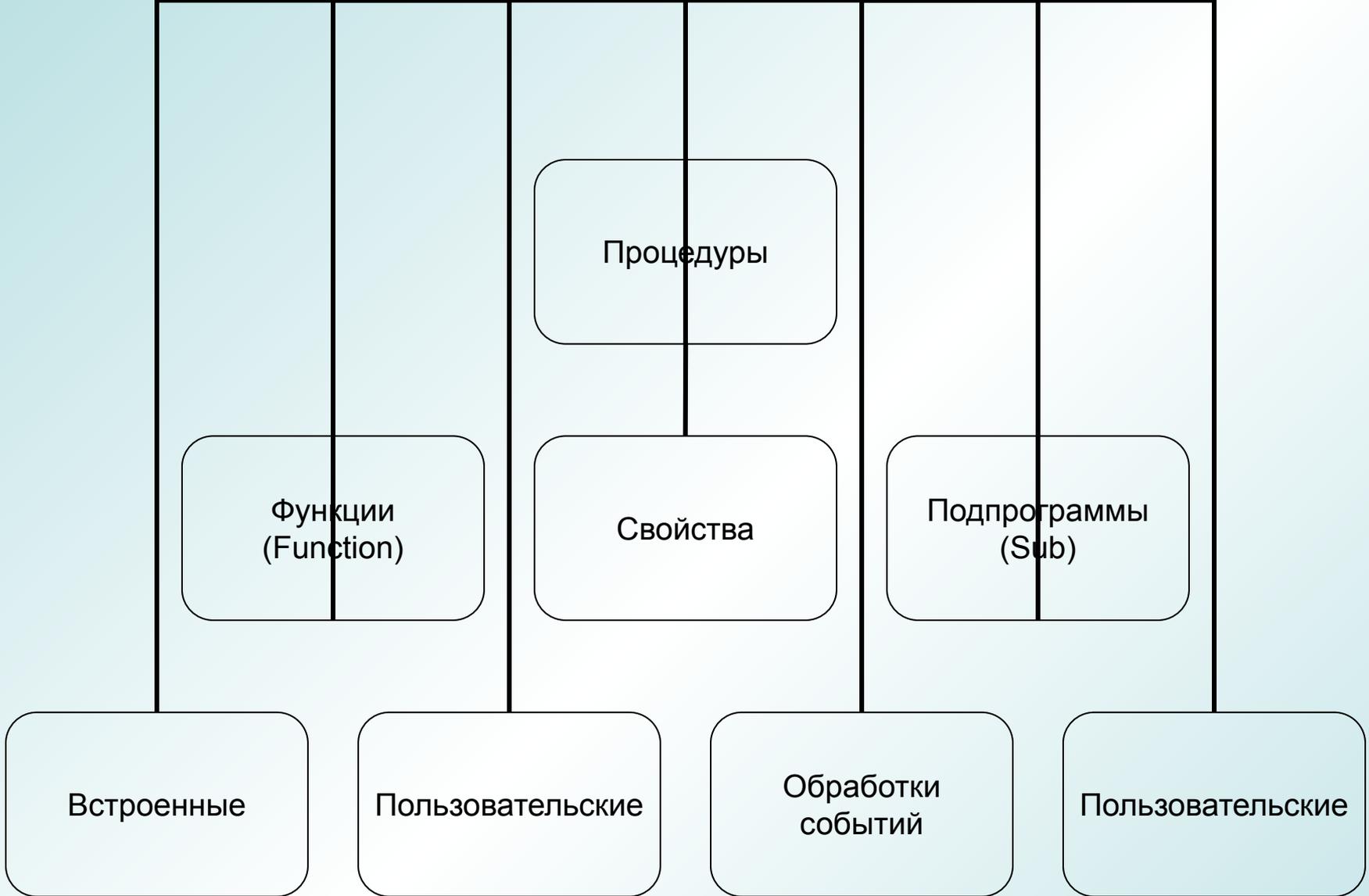
```
Private Sub CommandButton1_Click()  
Dim Wroom As Integer, HRoom As Integer  
Dim Sroom As Integer, stra As String  
stra = «вычисление площади комнаты»  
Wroom = InputBox(«введите длину", stra, 1)  
HRoom = InputBox(«введите ширину", stra, 1)  
Sroom = Wroom * HRoom  
MsgBox «площадь вашей комнаты = " & Sroom, vbInformation, stra  
End Sub
```

Использование переменной в
диалоговых окнах



В простых приложениях применение программ VBA ограничивается созданием процедур обработки событий и простых процедур – функций, используемых в приложениях.





Процедура программы

Могут вызываться из других процедур по имени. После выполнения подпрограммы управление передается в приложение или процедуру, которая ее вызывала. НЕ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ В ВЫРАЖЕНИИ



```
[Public| Private][Static] Sub <имя>[(список  
аргументов )]
```

```
<объявление переменных>
```

```
  [<инструкции>]
```

```
[Exit Sub]
```

```
  [<инструкции>]
```

```
End Sub
```



Public –общая, т.е. Доступная для всех других процедур во всех модулях (по умолчанию)

Private- доступна только в том модуле, в котором она описана

Static – значения локальных переменных сохраняются между вызовами этой процедуры

Exit Sub – немедленное завершение процедуры

Список аргументов – список переменных, которые передаются процедуре при вызове



Частный случай – процедура
обработки событий .

Это процедура типа **Sub**, связанная с конкретным элементом управления и конкретным событием.

Автоматически выполняется в ответ на событие



Функции

- При вызове функции имя функции выступает в роли переменной, содержащей вычисленное значение.
- Функции VBA как правило располагают списком параметров и не могут использоваться для обработки событий или в качестве программных макросов



Основная разница между функцией и процедурой- Функция может возвращать значение , а подпрограмма нет



Общее между функцией и процедурой

- Содержат операторы VBA, которые выполняют операции или вычисляют значения
- Могут принимать на вход аргументы
- Могут вызываться из других процедур



[Public| Private][Static] Function <имя
функции>[(список аргументов)] As <тип
данных>

<объявление переменных>

[<инструкции>]

<имя функции> =<выражение>

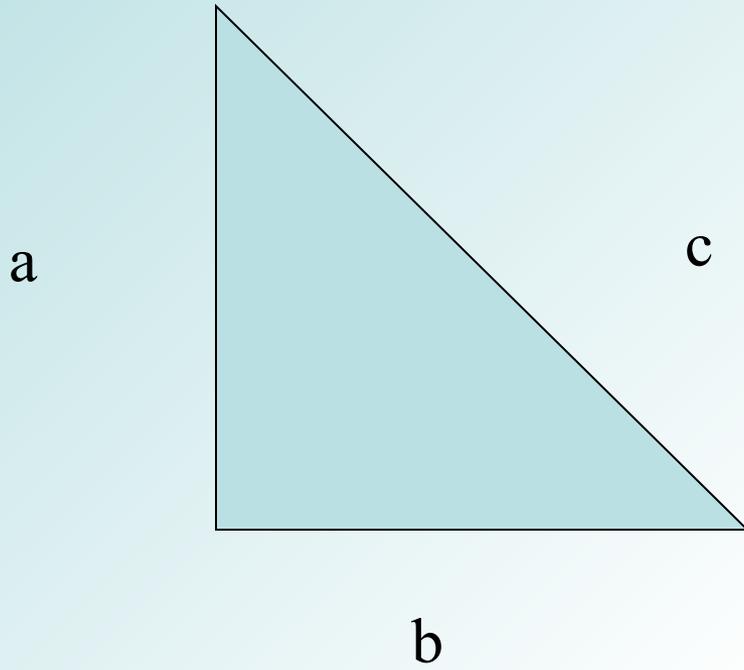
[Exit Function]

[<инструкции>]

<имя функции> =<выражение>

End Function





$$c = \sqrt{a^2 + b^2}$$



Подпрограмма

```
Sub PIFAGOR(X,Y,Z)
```

```
Z=Sqr(x*x+y*y)
```

```
End Sub
```

Функция

```
Function FPIFAGOR(x,y)
```

```
FPIFAGOR=Sqr(x*x+y*y)
```

```
End Function
```

ВЫЗОВ

```
Sub main( )
```

```
a= InputBox(“введите катет1 “,”вычисление гипотенузы “)
```

```
b= InputBox(“введите катет2 “,”вычисление гипотенузы “)
```

```
Call PIFAGOR(a, b, c) ‘ ВЫЗОВ ПОДПРОГРАММЫ
```

```
MsgBox(“гипотенуза= “ & c)
```

```
c=FPIFAGOR(a,b) ‘ ВЫЗОВ ФУНКЦИИ
```

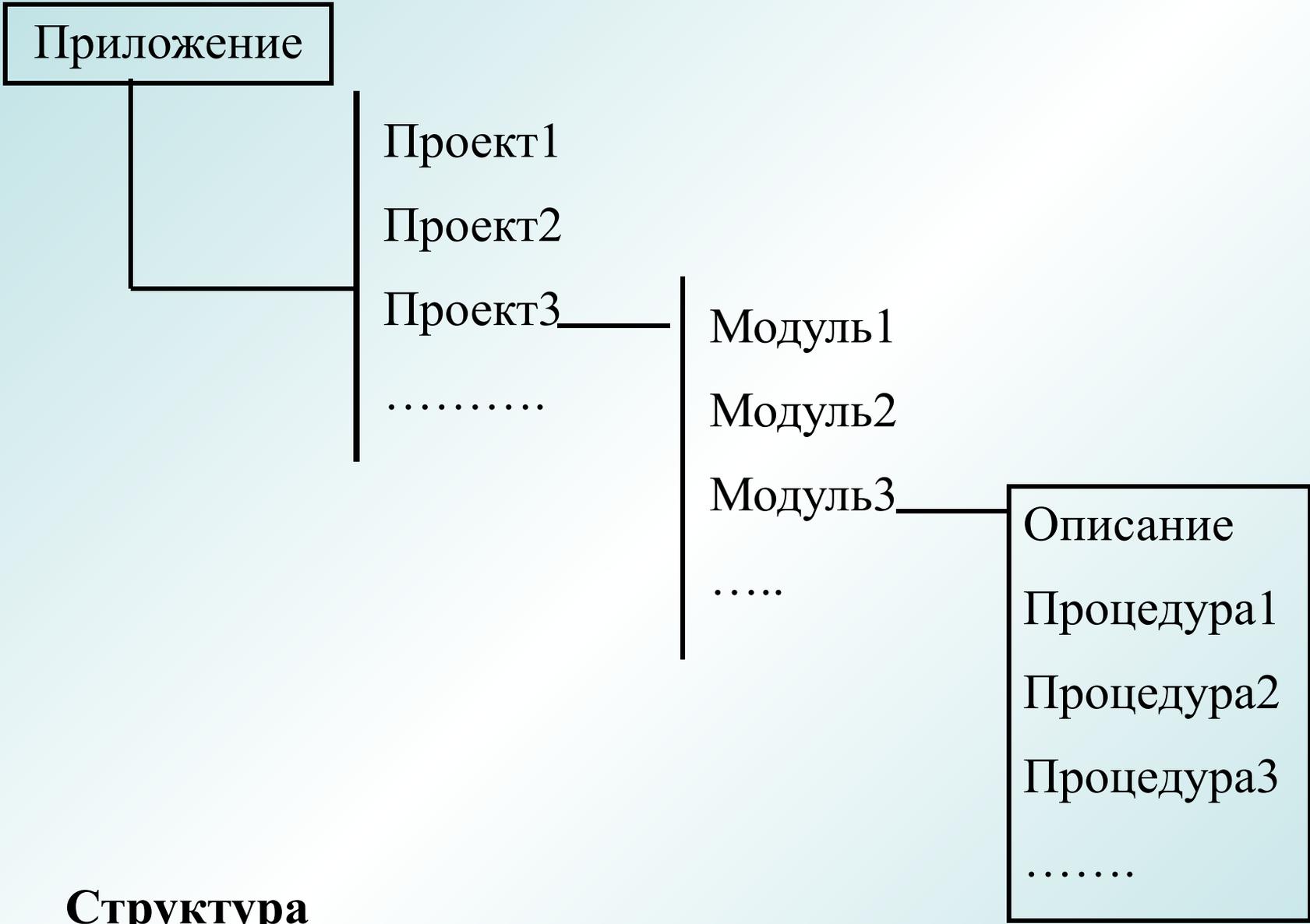
```
MsgBox(“гипотенуза= “ & FPIFAGOR(a,b) )
```

```
End Sub
```



- Как правило программа на VBA состоит из **нескольких подпрограмм и функций**. Описание этих программных единиц хранится в модулях. Модули входят составными частями в проекты. Проекты хранятся в файлах документов приложений.





Структура программы



Приложение

Public

Проект

Модуль

Private, Dim

Процедура

Dim

Static

