



РЕАЛИЗАЦИЯ АЛГОРИТМОВ В СРЕДЕ ПРОГРАММИРОВАНИЯ

Часть 1. Интерфейс VBA

СИСТЕМЫ ПРОГРАММИРОВАНИЯ

Для создания компьютерных программ различного назначения используются системы программирования

Системы программирования – это комплекс инструментальных программных средств, предназначенных для работы с программами на одном из языков программирования

УРОВНИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Реализация компьютерного приложения для решения какой-либо задачи предполагает наличие четко сформулированных **команд** или **инструкций**

Создание необходимой для решения поставленной задачи **последовательности команд** и является **целью программирования**

Программа – это состоящий из **набора команд** или **инструкций алгоритм**, по которому компьютер обрабатывает **информацию** для **решения конкретной задачи**

УРОВНИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

В основе классификации языков программирования лежит **понятие уровня**.

Языки низкого уровня. Ядром любой вычислительной системы, как известно, является **центральный процессор**, который работает исключительно с **двоичными** данными. Таким образом для выполнения **процессором** любой команды необходимо, чтобы она была представлена в виде **двоичного** числа. Набор выполняемых процессором **двоичных кодов** называют **машинным кодом**, который является базовым языком процессора.

Разработку программ в машинных кодах называют программированием на языке **низкого уровня**. К языкам **низкого уровня** относятся **машинные языки**, **ассемблеры**, **макроассемблеры**.

ЯЗЫКИ ВЫСОКОГО УРОВНЯ

В настоящее время для разработки **прикладных приложений** используются языки **высокого уровня**. Это более совершенные языки программирования, которые позволяют не работать напрямую с машинными кодами. Они ориентированы на описание **алгоритмов** и называются соответственно, **алгоритмическими языками программирования**.

Преимуществом этих языков является **большая наглядность** и **независимость** от **конкретного** компьютера. К их недостаткам относятся дополнительные затраты времени и памяти для трансляции.

ЯЗЫКИ ВЫСОКОГО УРОВНЯ

Создание программы на алгоритмических языках содержит набор команд, написанных на ограниченном естественном языке. Команда на выполнение носит название оператора. При создании программы на алгоритмическом языке нужно тщательно придерживаться его синтаксиса.

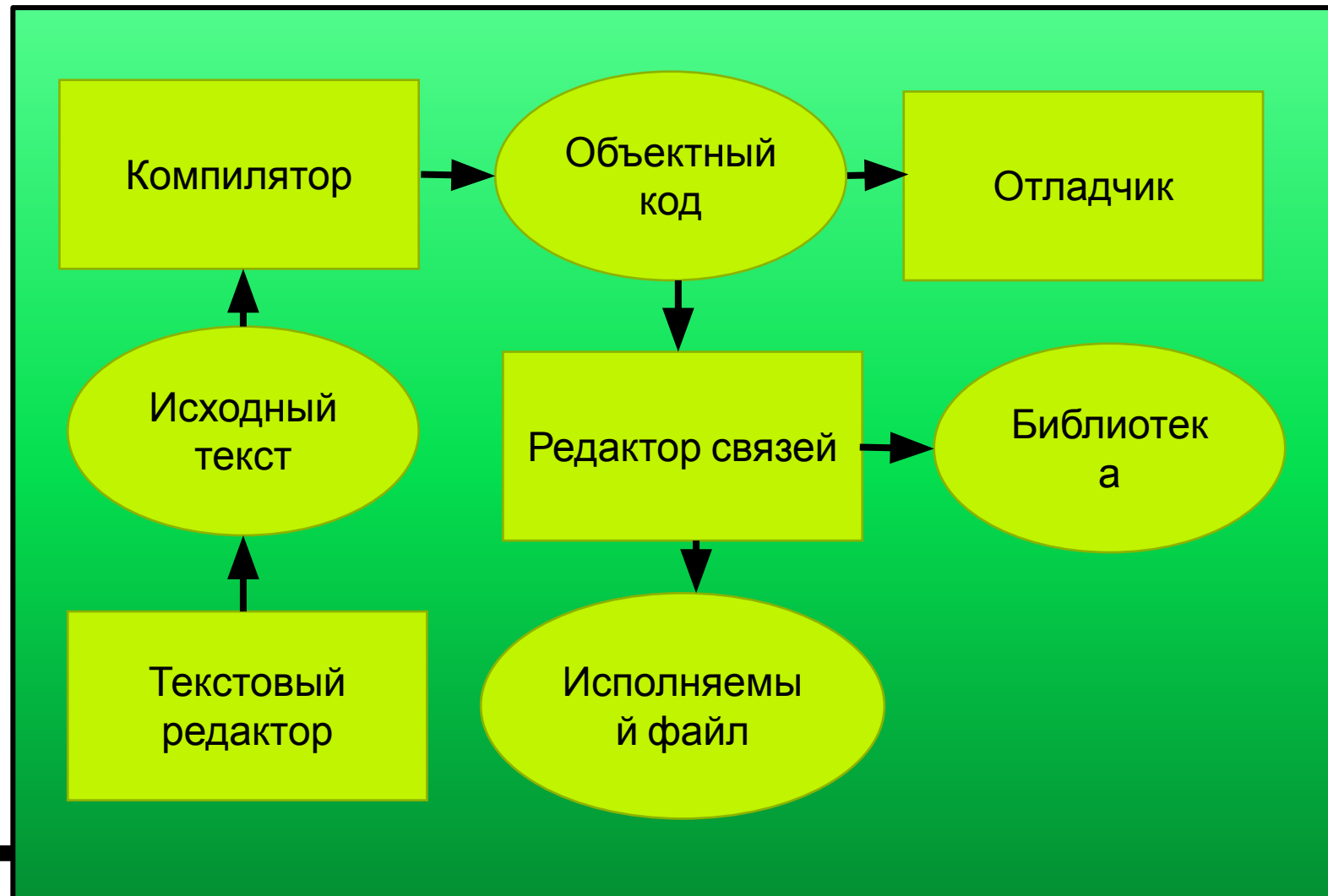
Общей тенденцией развития языков программирования, да и всего программирования в целом, является повышение их уровня.

Языками высокого уровня являются такие языки как Фортран, Visual Basic, Object Pascal, C/C++, C# и т.д.

СРЕДЫ РАЗРАБОТКИ ПРОГРАММ

Современные языки высокого уровня предоставляют разработчикам развитые среды разработки. Эти среды содержат:

- текстовые редакторы;
- средства преобразования программ к машинному виду, пригодному для выполнения на вычислительной системе (ВС);
- средства отладки





ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА VBA

ЯЗЫК VISUAL BASIC FOR APPLICATION (VBA)

- ✓ Язык VBA представляет собой набор средств программирования для создания собственных структурированных программ и расширений возможностей приложений пакета Microsoft Office под запросы пользователя.
- ✓ Фирма Microsoft создала язык VBA и обеспечила его поддержку во всех основных приложениях Microsoft Office: Word, Excel, Access и PowerPoint.
- ✓ Заметим, что если приложения могут взаимодействовать между собой, то это способно приносить пользу.

СОЗДАНИЕ ПРИЛОЖЕНИЙ НА VBA

- ✓ Создание приложений на VBA осуществляется с помощью специальной интегрированной среды разработки, встроенной в Microsoft Office.
- ✓ Для написания, тестирования и отладки модулей VBA, из которых состоит создаваемое приложение, используется редактор VBE. Он запускается как автоматически – при редактировании макросов, так и отдельно – для создания новых модулей, процедур и функций. Редактор обладает всеми необходимыми средствами для управления VBA-кодом.
- ✓ В общем случае VBE обладает одинаковыми возможностями для большинства приложений пакета Microsoft Office. Особенности VBE для каждого приложения MS Office заключается в специфике объектных моделей соответствующих приложений.



ЗАПУСК РЕДАКТОРА VBA

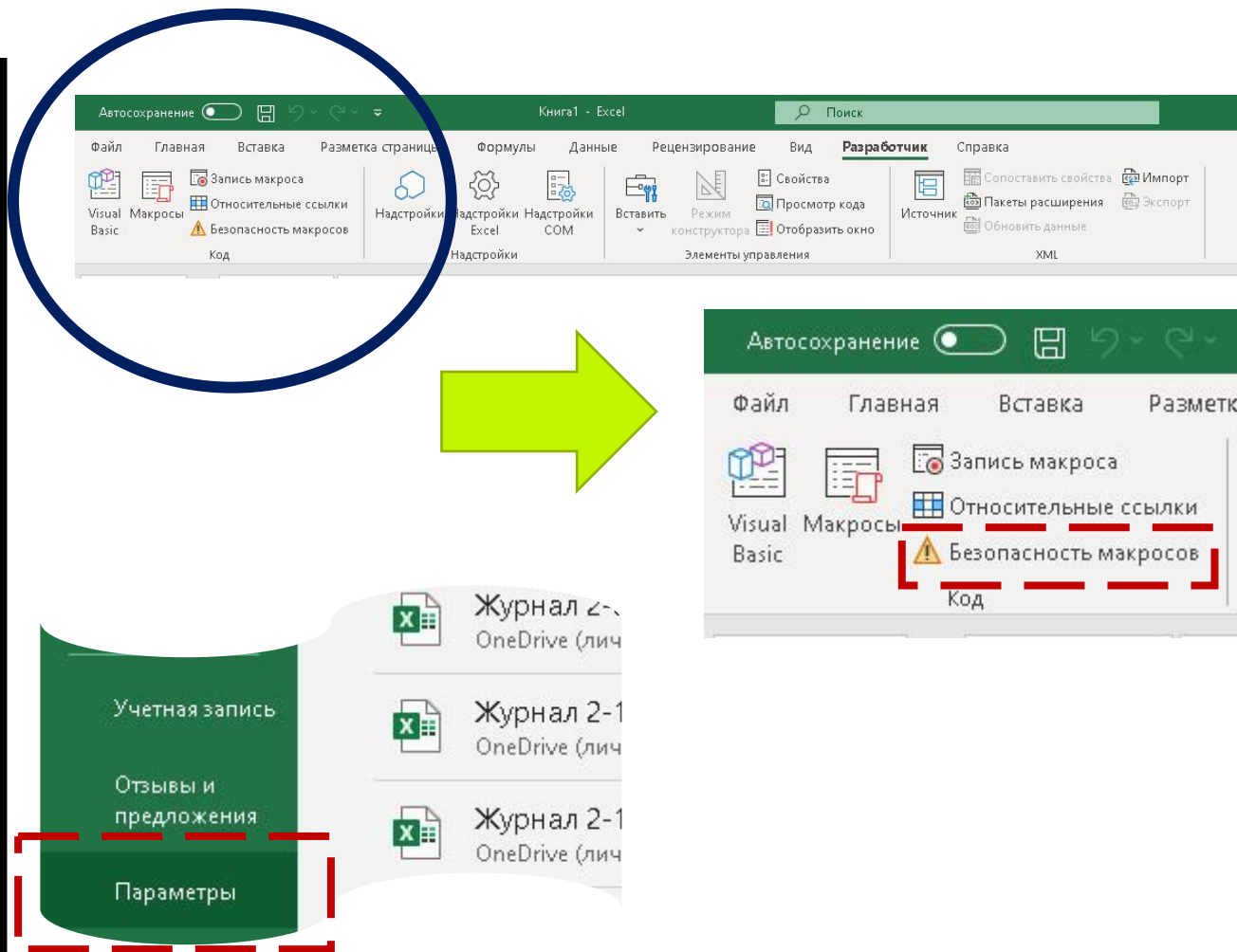
НАСТРОЙКА ПАРАМЕТРОВ БЕЗОПАСНОСТИ И ЗАПУСК РЕДАКТОРА VBA

- ✓ Для поддержки возможности выполнения разработанных на VBA программ в среде MS Excel прежде всего нужно осуществить настройку параметров безопасности макросов.
- ✓ Изменить параметры безопасности макросов можно в разделе центра управления безопасностью MS Excel. Получить доступ к параметрам безопасности макросов можно несколькими способами.

1-Й СПОСОБ. ИСПОЛЬЗУЕТСЯ ВКЛАДКА РАЗРАБОТЧИК

1. В группе **Код** на вкладке **Разработчик** нажмите кнопку **Безопасность макросов**

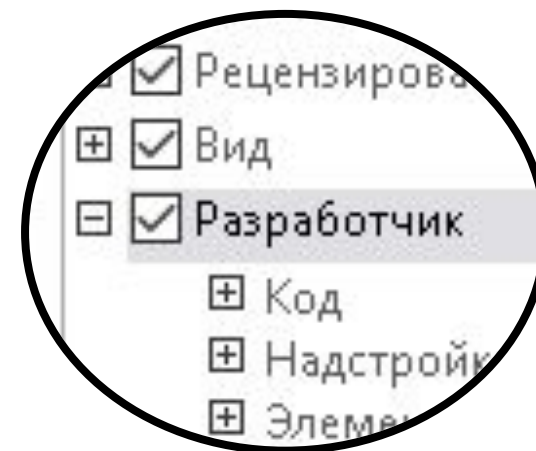
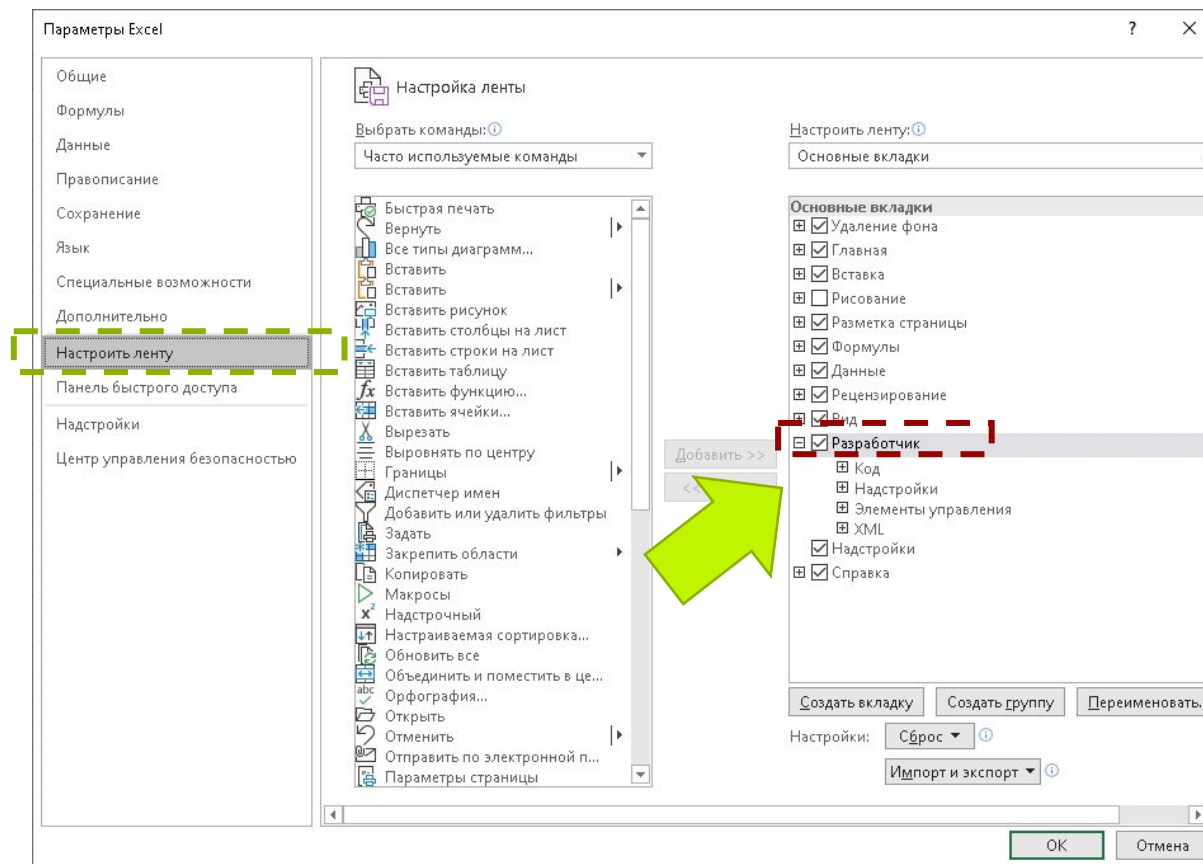
Замечание: если вкладка **Разработчик** не отображается, то на вкладке **Файл** выберите пункт **Параметры**.



1-Й СПОСОБ. ИСПОЛЬЗУЕТСЯ ВКЛАДКА РАЗРАБОТЧИК

Затем в открывшемся окне на левой панели выберите категорию **Настройка ленты**.

Наконец, на правой панели этого окна установите флажок **Разработчик** в разделе **Основные вкладки** и нажмите кнопку **ОК**

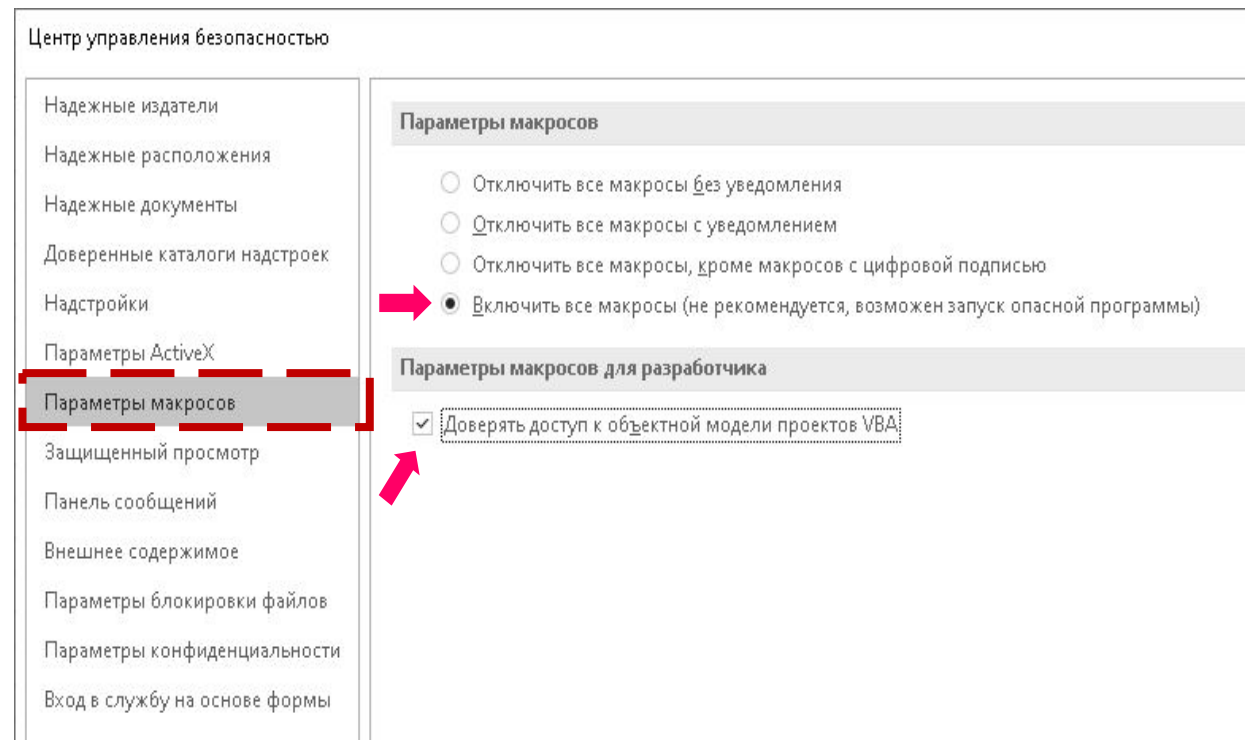


1-Й СПОСОБ. ИСПОЛЬЗУЕТСЯ ВКЛАДКА РАЗРАБОТЧИК

2. В категории **Параметры макросов** (разделы **Параметры макросов** и **Параметры макросов для разработчиков**) окна **Центр управления безопасностью** осуществите следующее.

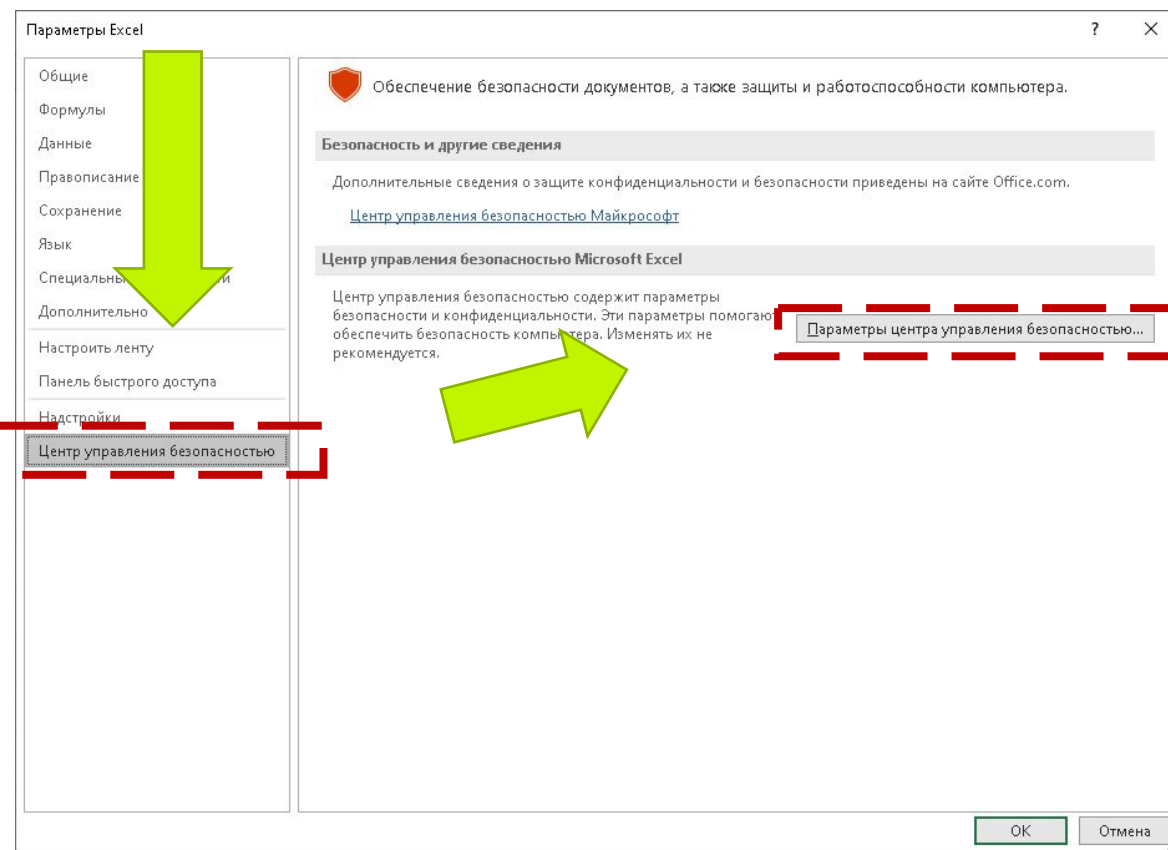
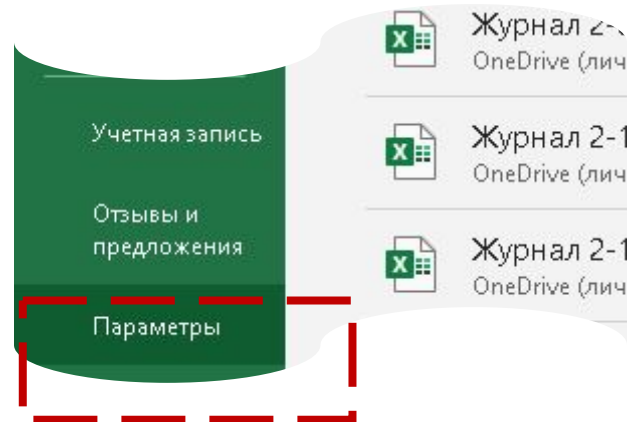
Выберите параметр **«Включить все макросы (не рекомендуется, возможен запуск опасной программы)»**. Следует заметить, что нужно соблюдать осторожность, так как в этом случае компьютер становится уязвимым для потенциально опасного кода

Установите флажок **«Доверять к объектной модели проектов VBA»**. Этот параметр предназначен для разработчиков и используется для запрещения или разрешения программного доступа к объектной модели VBA



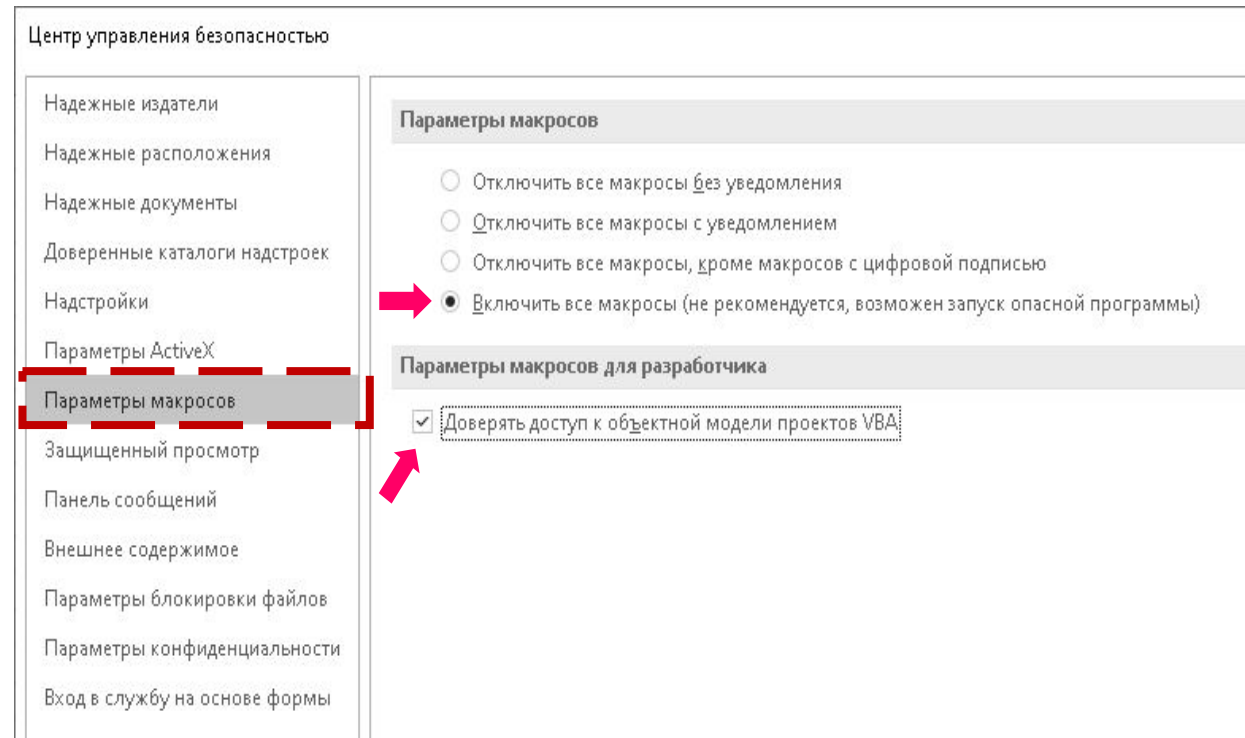
2-Й СПОСОБ. ИСПОЛЬЗУЕТСЯ ДИАЛОГОВОЕ ОКНО ПАРАМЕТРЫ EXCEL

1. На вкладке **Файл** выберите пункт **Параметры**. Затем в открывшемся окне на левой панели выберите категорию **Центр управления безопасностью**
2. В категории **Центр управления безопасностью** нажмите на кнопку **Параметры центра управления безопасностью**



2-Й СПОСОБ. ИСПОЛЬЗУЕТСЯ ДИАЛоговое ОКНО *ПАРАМЕТРЫ EXCEL*

3. В категории **Параметры макросов** в разделе **Параметры макросов** выберите нужную область безопасности, как это описано в п.2 первого способа

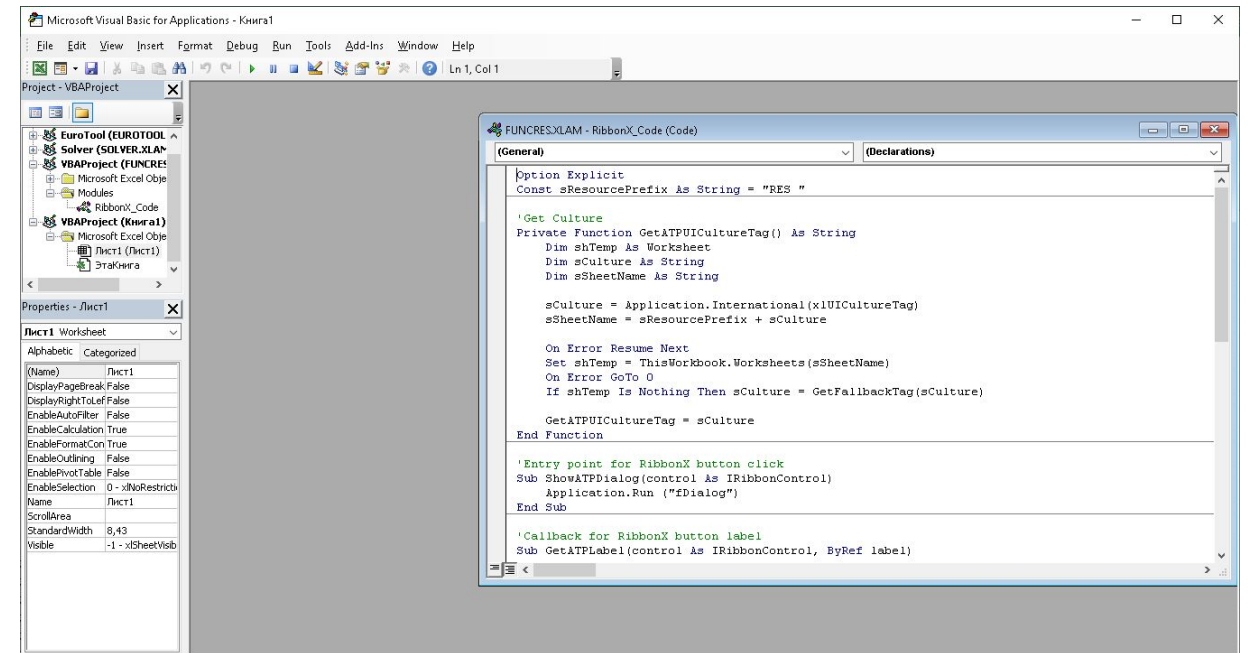


ЗАПУСК РЕДАКТОРА VBA


Переход в редактор VBE из приложения MS Excel осуществляется одним из способов:

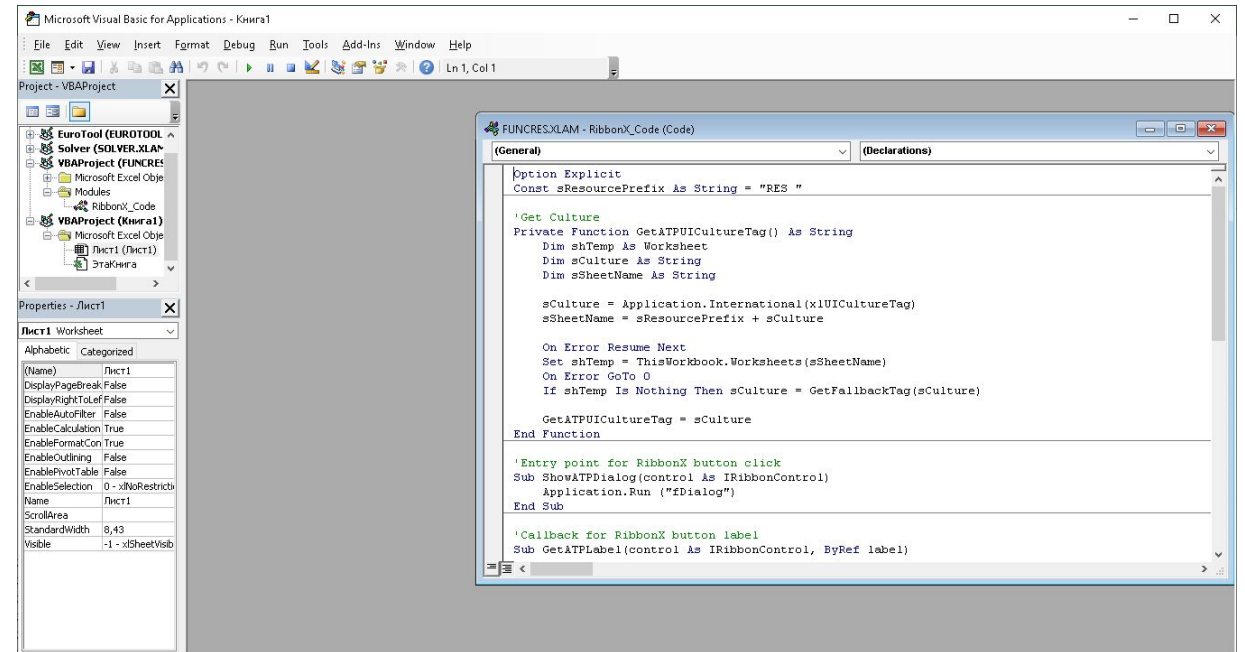
1. Щелкнуть на кнопке **Visual Basic** вкладки меню **Разработчик**;
2. нажать комбинацию клавиш **<Alt+F11>**

В любом случае запустится редактор VBE



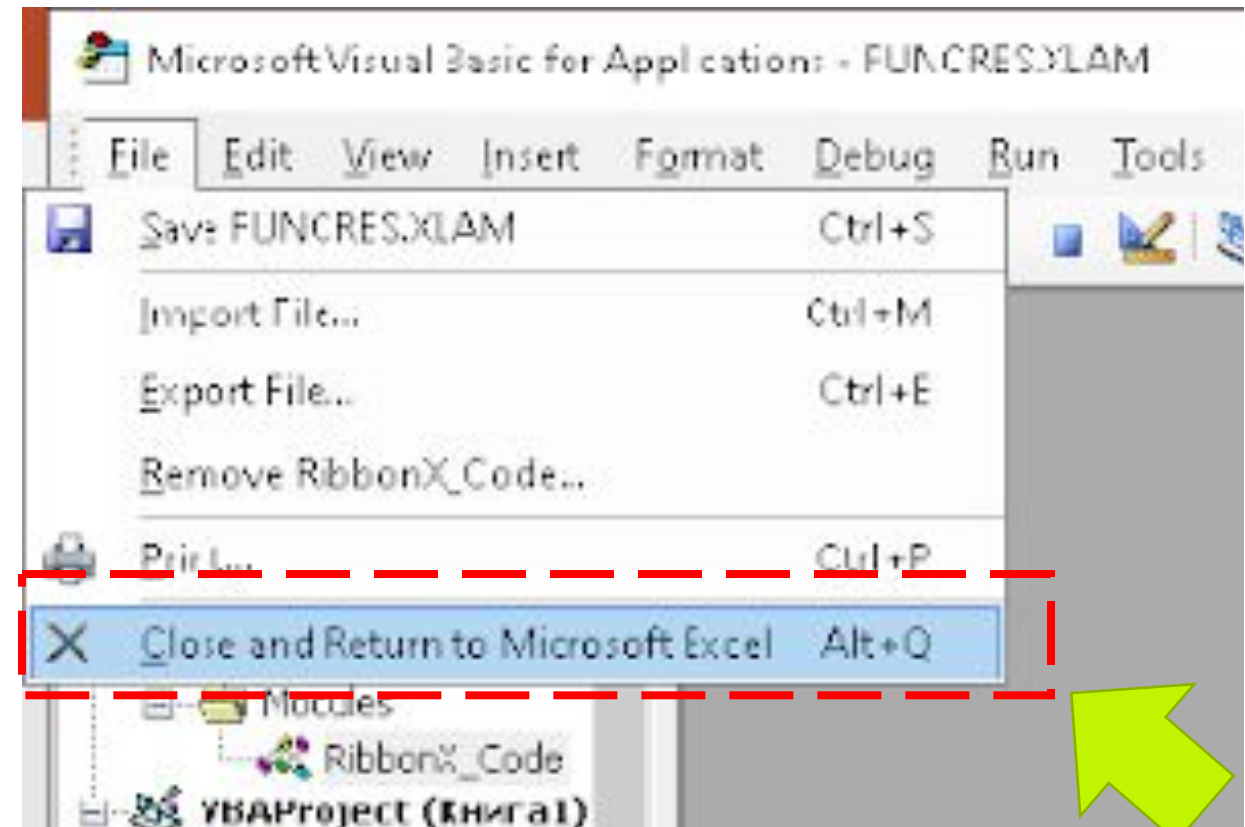
ОКНА VBA

Язык VBE состоит из ряда элементов. Это набор меню, панелей инструментов и окон, которые используются при создании программ. Для возврата в рабочую книгу, не закрывая редактора, можно использовать кнопку  или панель задач.



ОКНА VBA

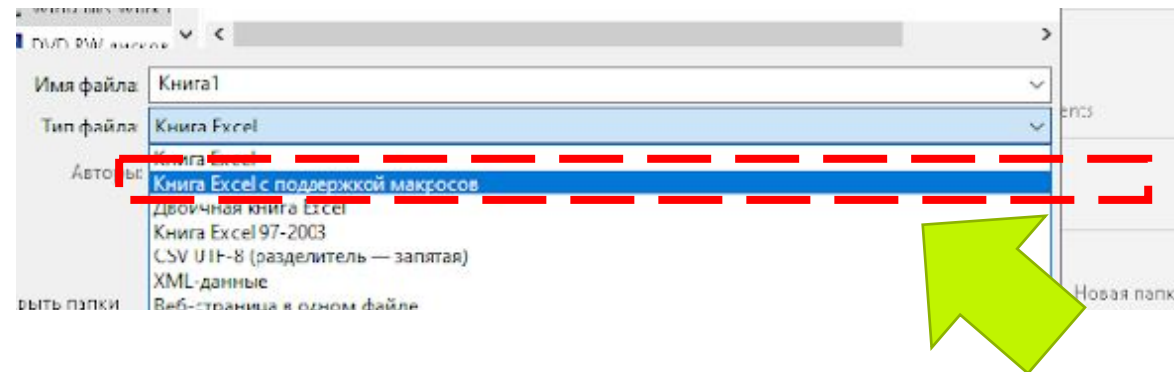
Главное окно обычно занимает весь экран. В окне имеются строка заголовка, меню и панель инструментов. В строке заголовка выводится имя текущей рабочей книги. В этом окне располагаются все другие окна. Для выхода из редактора VBA и возврата в рабочую книгу закройте главное окно или выберите команду меню **File** → **Close and Return to Microsoft Excel** (**Файл** → **Закреть и вернуться в Microsoft Excel**)



ОКНА VBA

Приложение, созданное на VBA, состоит из модулей. Модули бывают различных типов: модули форм, стандартные модули и модули класса. Каждый модуль представляет собой набор описаний и процедур. Совокупность модулей называется проектом. Модули чаще всего сохраняются в файле рабочей книги. При этом, для того чтобы проект сохранился вместе с разработанными на VBA модулями и формами, нужно в MS Excel, начиная с версии 2000, файл рабочей книги сохранять с поддержкой макросов (*.xlsm)

Добавить нижний колонтитул



ОКНА VBA

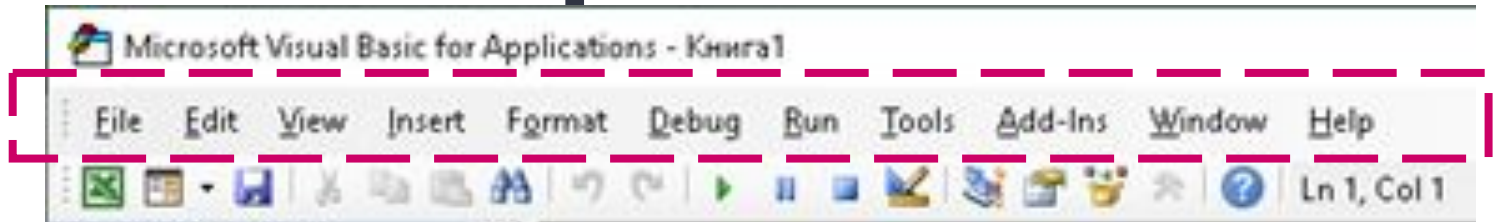
Существует также возможность сохранить модули в виде отдельных файлов и затем добавлять их в другие проекты. Для этого используются команды **File → Export File (Файл → Экспорт файла)** и **File → Import File (Файл → Импорт файла)**. При импорте в проект добавляется копия файла.

СТРОКА МЕНЮ

Строка меню (главное меню) содержит все команды, используемые в среде разработки VBA для создания приложений. Здесь есть пункты меню как стандартные, присущие большинству окон Windows (**File, Edit, View, Windows, Help**), так и специальные, предназначенные для разработки, запуска или отладки создаваемого приложения (Run, Debug). Кроме того, выполнение многих команд меню дублируется комбинацией клавиш.

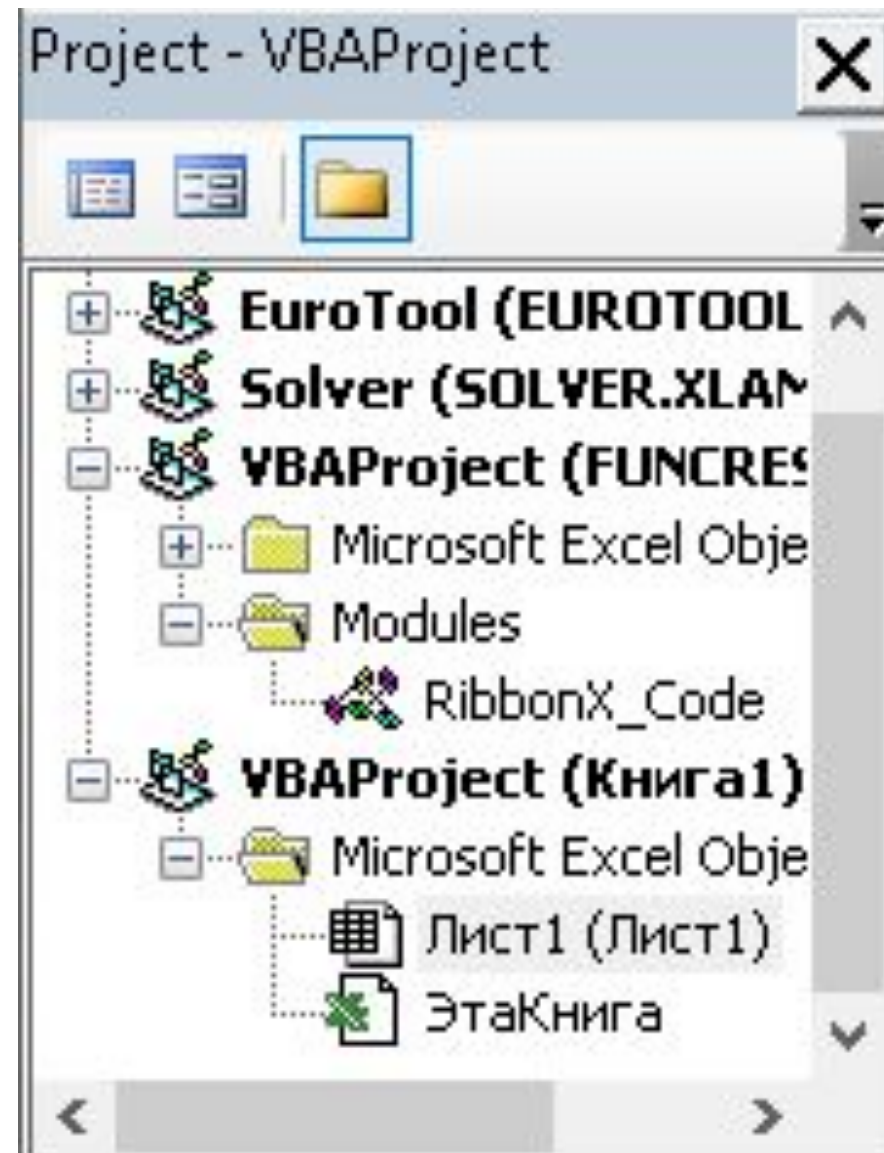
ПАНЕЛЬ ИНСТРУМЕНТ ОВ

Стандартная панель инструментов **Standard** содержит кнопки – аналоги часто используемых основных команд меню. С ее помощью можно выполнить разнообразный набор действий, которые реализованы в пунктах меню **File (Файл)**, **Project (Проект)**, **Run(Запуск)**, **Debug(Отладка)**. Данная панель, как правило, находится под строкой *Меню*.



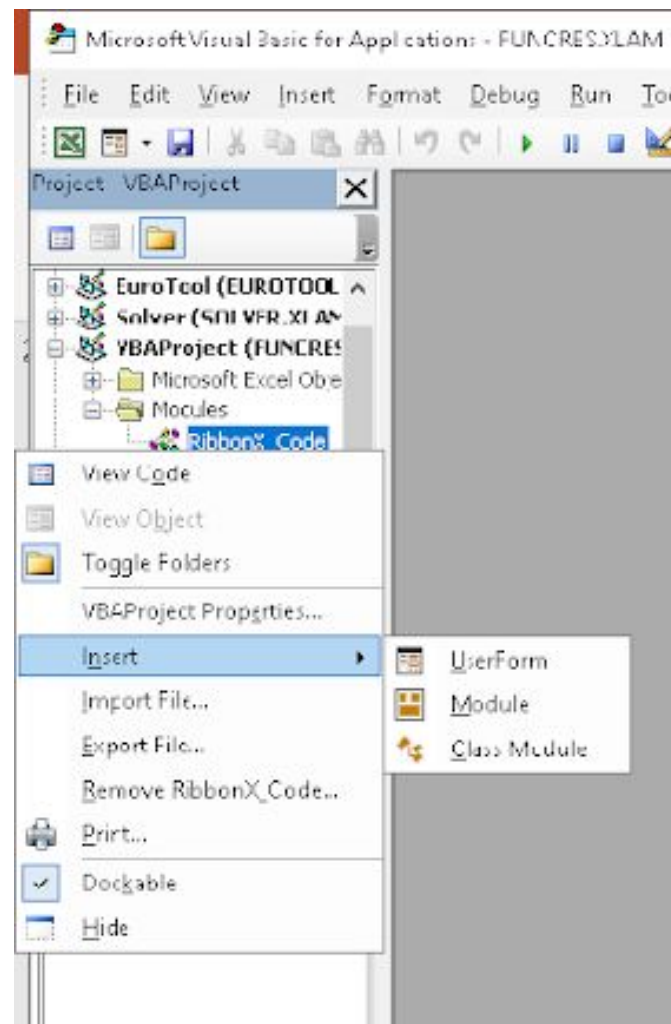
ОКНО ПРОЕКТА

В окне проекта (**Project Explorer**) отображаются иерархические структуры всех открытых в данный момент в *Excel* рабочих книг. Каждая рабочая книга является проектом. Открытие в данный момент проекты являются вершинами иерархических структур, а в качестве ветвей структур выступают составные части проектов (например, модули, формы).



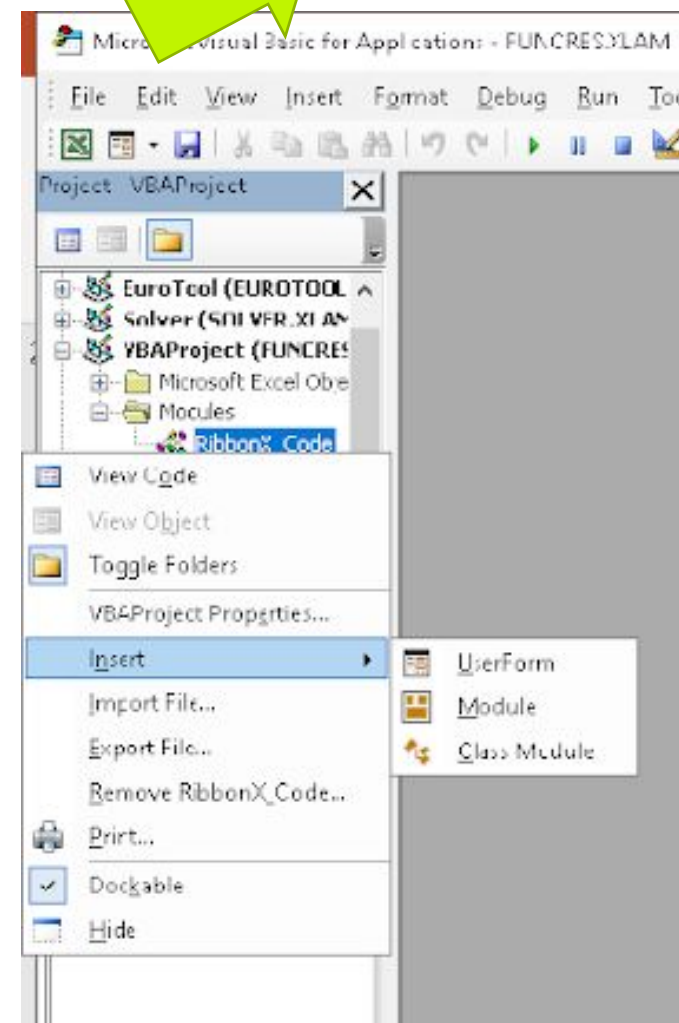
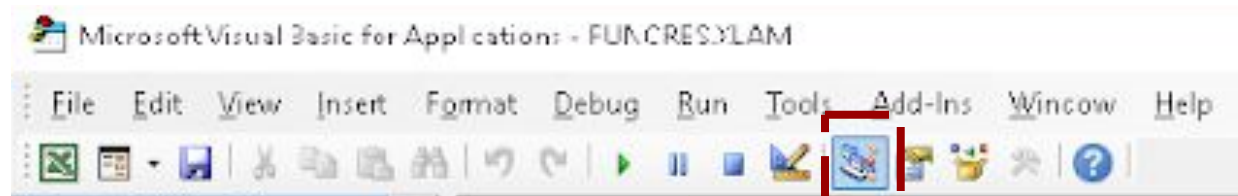
ОКНО ПРОЕКТА

В любом проекте автоматически создается модуль для каждого рабочего листа и рабочей книги в целом. При записи макроса к проекту добавляется стандартный модуль, содержащий текст макроса. Другие модули добавляет разработчик, используя команды меню **Insert (Вставка)**. Можно также применять контекстное меню.



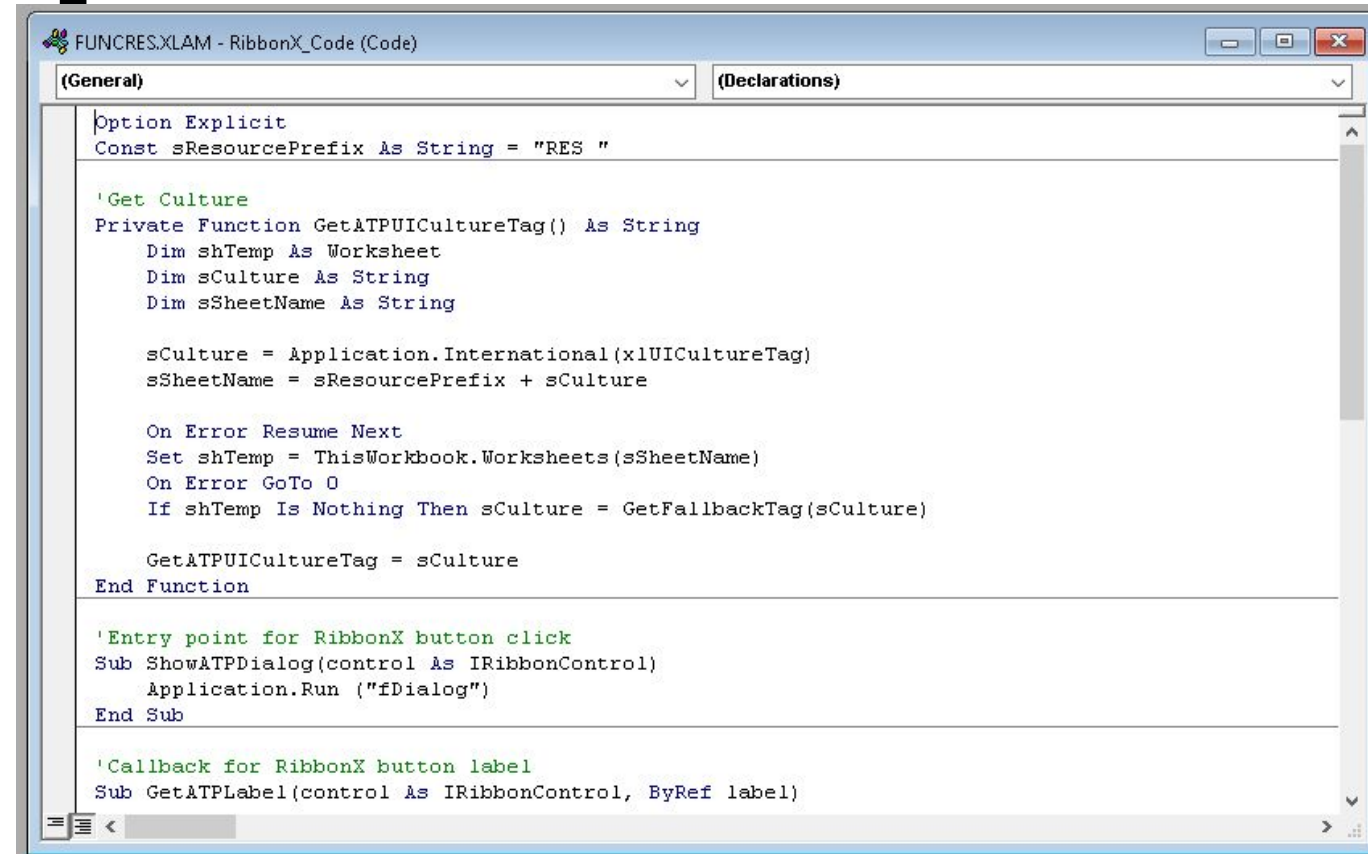
ОКНО ПРОЕКТА

Используя окно проекта, легко перейти от одного модуля к другому. Для этого достаточно выполнить двойной щелчок на имени нужного модуля. В верхней части окна имеются три кнопки: **View Code (Программа)**, **View Object (Объект)** и **Toggle Folders (Папки)**, предназначенные для выполнения следующих действий: просмотр и редактирование программы выбранного модуля, просмотр объектов выделенного модуля и переключение между режимами отображения компонентов проекта. Для просмотра, удаления, сохранения и добавления отдельных частей проекта удобно использовать контекстное меню выделенного элемента проекта. Открыть окно проекта можно с помощью команды **View → Project Explorer (Вид → Окно проекта)** или инструмента



ОКНО КОДА

Программный код, или просто код, - это список инструкций, написанных на языке программирования VBA (текст программы). Для создания, просмотра и редактирования текста программы используется окно кода. Это окно реализует возможности простейшего текстового редактора.



```
FUNCRES.XLAM - RibbonX_Code (Code)
(General) (Declarations)
Option Explicit
Const sResourcePrefix As String = "RES "

'Get Culture
Private Function GetATPUICultureTag() As String
    Dim shTemp As Worksheet
    Dim sCulture As String
    Dim sSheetName As String

    sCulture = Application.International(xlUICultureTag)
    sSheetName = sResourcePrefix + sCulture

    On Error Resume Next
    Set shTemp = ThisWorkbook.Worksheets(sSheetName)
    On Error GoTo 0
    If shTemp Is Nothing Then sCulture = GetFallbackTag(sCulture)

    GetATPUICultureTag = sCulture
End Function

'Entry point for RibbonX button click
Sub ShowATPDialog(control As IRibbonControl)
    Application.Run ("fDialog")
End Sub

'Callback for RibbonX button label
Sub GetATPLabel(control As IRibbonControl, ByRef label)
```


ОКНО КОДА

Вверху этого окна находятся два поля списка. Одно из них (левое) содержит набор объектов (например,, управляющие элементы, формы), а второе (правое) – набор событий для выбранного элемента. Первой строкой в списке объектов является **General** - раздел глобальных объявлений модуля.

```
Option Explicit
Const sResourcePrefix As String = "RES "

'Get Culture
Private Function GetATPUICultureTag() As String
    Dim shTemp As Worksheet
    Dim sCulture As String
    Dim sSheetName As String

    sCulture = Application.International(xlUICultureTag)
    sSheetName = sResourcePrefix + sCulture

    On Error Resume Next
    Set shTemp = ThisWorkbook.Worksheets(sSheetName)
    On Error GoTo 0
    If shTemp Is Nothing Then sCulture = GetFallbackTag(sCulture)

    GetATPUICultureTag = sCulture
End Function

'Entry point for RibbonX button click
Sub ShowATPDialog(control As IRibbonControl)
    Application.Run ("fDialog")
End Sub

'Callback for RibbonX button label
Sub GetATPLabel(control As IRibbonControl, ByRef label)
```

ОКНО КОДА

Для ускорения работы интерпретатора VBE каждую введенную строку (после нажатия клавиши Enter) преобразует в промежуточный код. При обнаружении же синтаксической ошибки сразу выдается диагностическое сообщение. Кроме того, при вводе многих свойств, методов, функций и процедур отображается синтаксическая подсказка, что облегчает создание программ

```
(General) GetATPUICultureTag
Option Explicit
Const sResourcePrefix As String = "RES "

'Get Culture
Private Function GetATPUICultureTag() As String
    Dim shTemp As Worksheet
    Dim sCulture As String
    Dim sSheetName As String
    Dim snTemp As

    sCulture = i
    sSheetName =
    On Error Res
    Set shTemp =
    On Error Go
    If shTemp Is


    GetATPUICultureTag = sCulture
End Function

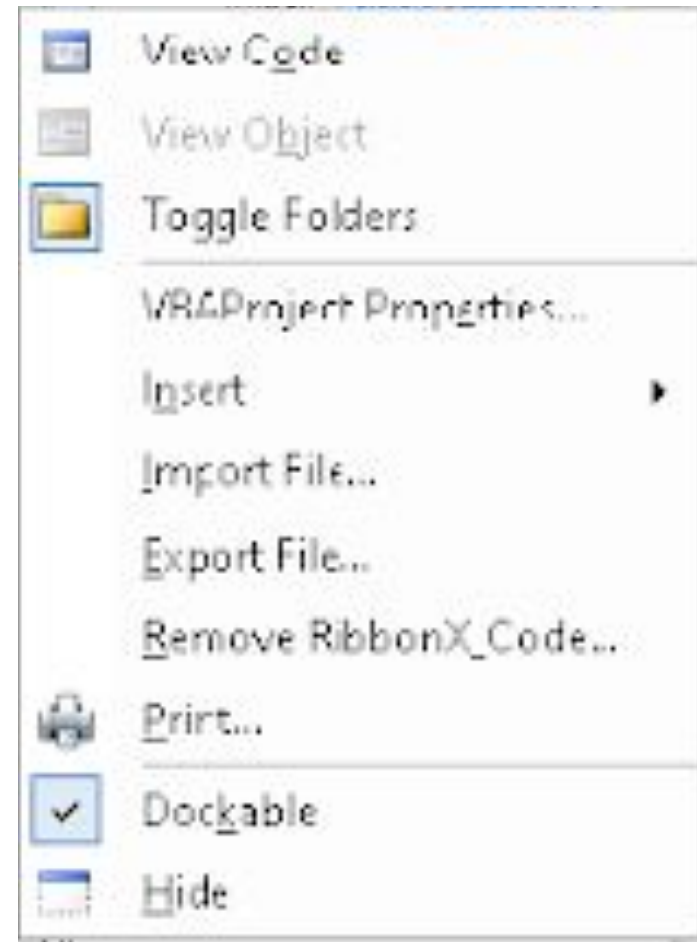
'Entry point for RibbonX button click
Sub ShowATPDialog(control As IRibbonControl)
    Application.Run ("Dialog")
End Sub

'Callback for RibbonX button label
Sub GetATPLabel(control As IRibbonControl, ByRef label)
    label = ThisWorkbook.Sheets(sResourcePrefix + GetATPUICultureTag()).Range("RibbonCommand").V
End Sub

'Callback for screentip
Public Sub GetATPScreenTip(control As IRibbonControl, ByRef label)
    label = ThisWorkbook.Sheets(sResourcePrefix + GetATPUICultureTag()).Range("ScreenTip").Value
End Sub
```

ОКНО КОДА

Для открытия окна кода в окне проекта выберите имя модуля и воспользуйтесь кнопкой  (**View Code**). Можно также выполнить команду меню **View → Code (Вид → Программа)** или команду контекстного меню **View Code**.



ОКНО КОДА

Для создания заготовок процедур и функций удобно использовать команду меню **Insert** → **Procedure** (**Вставка** → **Процедура**)



ОКНО КОДА

Переменные и типы данных обычно описываются в начале модуля(до процедур). В зависимости от настройки VBA может либо требовать, либо не требовать предварительного объявления переменных. Если поместить в первой строке оператор *Option Explicit*, то будет включен режим когда все используемые переменные должны быть предварительно объявлены.



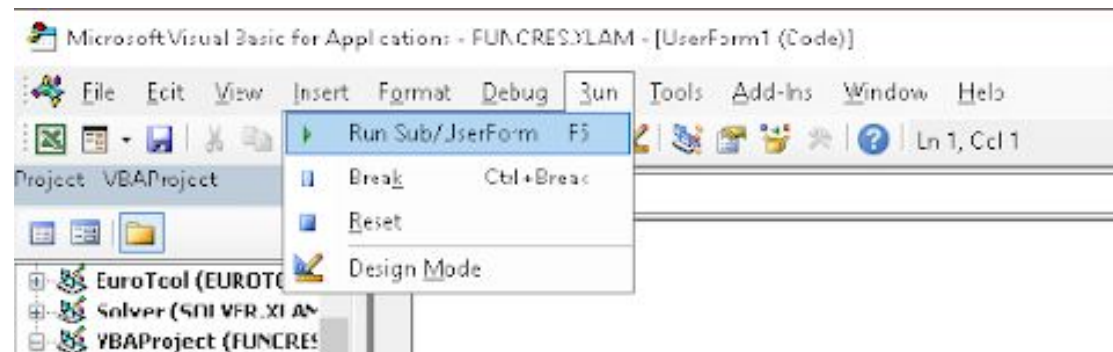
ОКНО КОДА

Если процедура, функция или тип данных могут потребоваться в другом проекте, то их лучше разместить не в модуле формы, а в отдельном модуле, не связанном с формой (стандартном модуле). Такие модули добавляются к проекту командой **Insert** → **Module** (**Вставка** → **Модуль**).



ОКНО КОДА

Существуют два режима работы с программой на VBA: режим проектирования и режим выполнения. В режиме проектирования создается или изменяется программный код. Режим выполнения нужен для тестирования программы или работы с ней в качестве обычного пользователя. Для перехода в режим выполнения необходимо в окне нужного модуля выполнить команду меню **Run → Run Sub (Запуск → Запуск программы)** или нажать клавишу F5. Для возврата в режим проектирования (останов программы) нужно выбрать команду **Run → Reset (Запуск → Сброс)**.



КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1. Каковы основные возможности Visual Basic Editor?
- 2. Как в MS Excel можно перейти в редактор VBE?
- 3. Как можно добавить модуль в проект?
- 4. Из каких элементов состоит редактор VBE?
- 5. Каковы режимы работы с программой в VBE и их назначение?
- 6. Что такое окно проекта?
- 7. Каково назначение и содержание стандартной панели инструментов?
- 8. Какие команды содержит строка меню?

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 9. Как включить режим работы обязательного предварительного объявления переменных?
- 10. Как открыть окно редактирования кода из окна проекта?
- 11. Как открыть окно проекта?
- 12. Из каких типов модулей может состоять созданное в VBA приложение?
- 13. Какие существуют способы изменения параметров безопасности?
- 14. Какие параметры безопасности нужно установить для работы с макросами?

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 15. С каким расширением рекомендуется сохранять файлы рабочей книги с разработанными на VBA модулями?