

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Техническая эксплуатация автомобилей»

# ИНФОРМАТИКА

*Методические рекомендации к самостоятельной работе  
для студентов специальности 1-37 01 07 «Автосервис»  
заочной формы обучения*



Могилев 2021

УДК 004.41/42  
ББК 32.973-018  
И146/1

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Техническая эксплуатация автомобилей»  
«15» января 2021 г., протокол № 9

Составители: ст. преподаватель О. А. Пономарева;  
ассистент Г. С. Мигурский;  
ассистент С. Ю. Билык

Рецензент С. К. Крутолевич

Изложены рекомендации к самостоятельной работе по дисциплине  
«Информатика» для студентов специальности 1-37 01 07 «Автосервис» заочной  
сокращенной формы обучения.

Учебно-методическое издание

## ИНФОРМАТИКА

Ответственный за выпуск	О. В. Билык
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2021

## Содержание

Введение.....	4
1 Краткие теоретические сведения.....	5
2 Пример выполнения задания .....	17
3 Задания для самостоятельного выполнения.....	21
Список литературы .....	23

## **Введение**

Методические рекомендации включают в себя краткие теоретические сведения, задания и пример выполнения заданий, на основании которых выполняется аудиторная контрольная работа по дисциплине «Информатика».

Целью самостоятельной работы студентов является изучение основ программирования, методов математического моделирования, общих вопросов алгоритмизации и приобретение навыков решения задач с применением средств вычислительной техники.

На выполнение аудиторной контрольной работы согласно учебной программы, студентам отводится 2 часа.

## 1 Краткие теоретические сведения

Среда *Visual Basic for Application* (VBA) является интегрированной, т. е. объединяет в себе несколько составляющих: компилятор, редактор текстов, средства отладки, визуальный конструктор форм и т. д.

Запуск редактора VBA можно осуществить одним из следующих способов: либо на вкладке **Разработчик** в группе **Код** нажмите кнопку *Visual Basic*, либо используйте сочетание клавиш **ALT + F11**.

В результате вы попадаете в интегрированную среду разработки приложений редактора Visual Basic (рисунок 1).

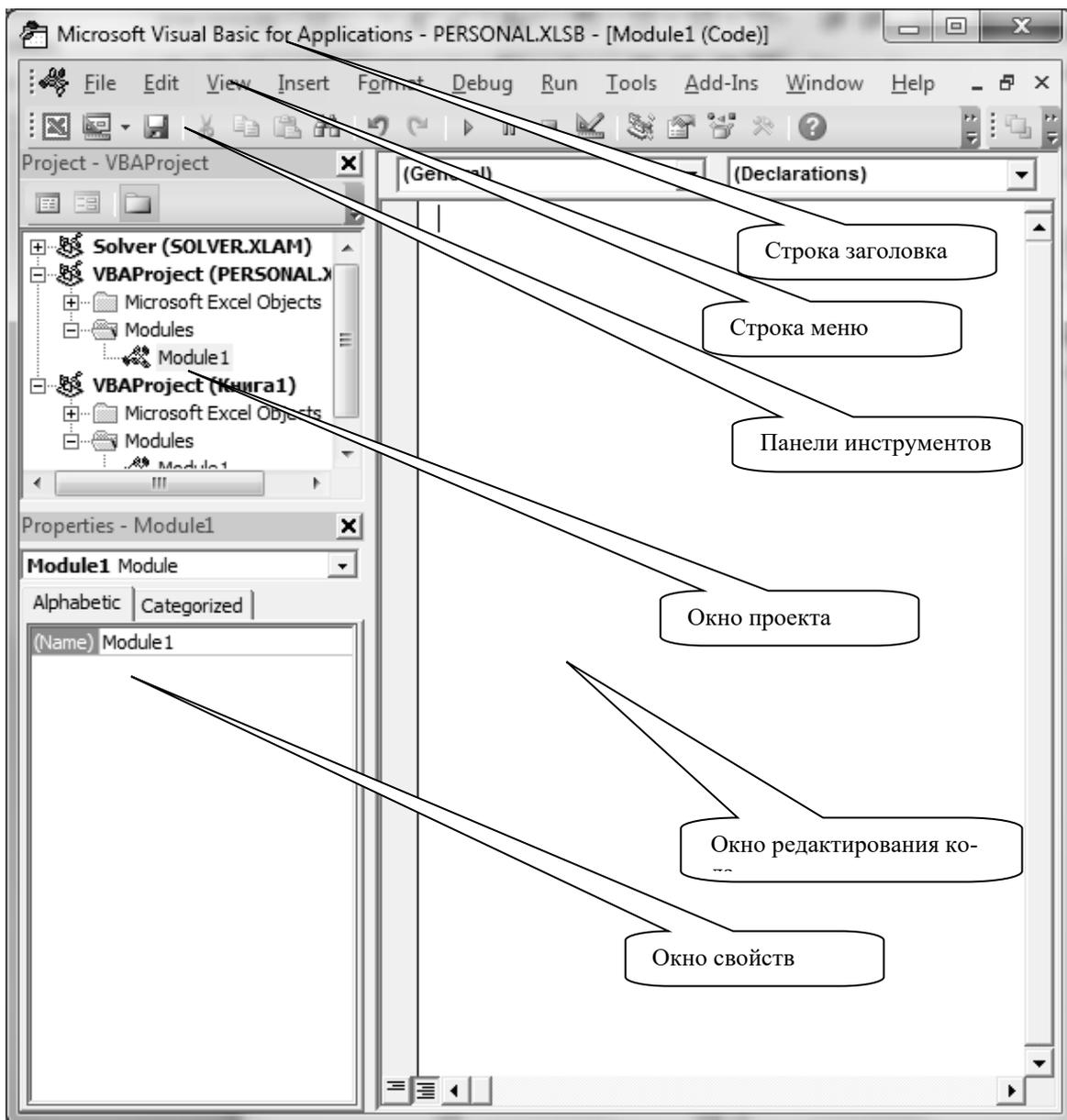


Рисунок 1 – Редактор Visual Basic

Возвратиться из редактора VBA в рабочую книгу Microsoft Excel можно нажатием кнопки  на панели управления VBA.

Окно редактора VBA состоит из следующих основных компонентов (см. рисунок 1):

- строки заголовка;
- строки меню;
- панели инструментов;
- окна проекта;
- окна свойств;
- окна редактирования кода;
- окна форм.

**Строка меню**, как и во всех приложениях Windows, представляет собой линейку раскрывающихся меню. Для выполнения многих команд меню можно использовать комбинации клавиш. Рассмотрим назначение пунктов меню редактора VBA (рисунок 2).

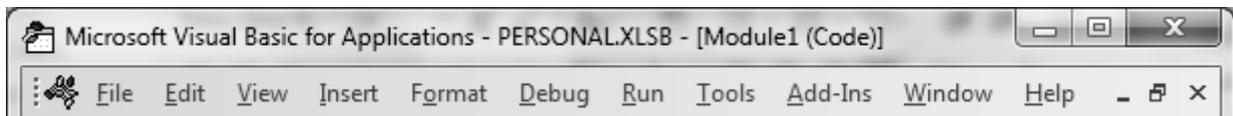


Рисунок 2 – Строка меню

Меню **File (Файл)** содержит команды, необходимые для сохранения изменений в проекте VBA и вывода на экран или на печать исходного кода макросов.

Меню **Edit (Правка)** содержит команды, предназначенные для управления исходным кодом макроса в окне редактирования кода (Code), а также объектами в формах.

Меню **View (Вид)** содержит команды, позволяющие выводить или убирать с экрана различные окна редактора VBA.

Команды меню **Insert (Вставка)** позволяют добавлять в проект различные объекты – процедуры, модули, формы, классы и пр.

Меню **Format (Формат)** содержит команды, используемые при создании пользовательских диалоговых окон. Команды этого меню позволяют выравнивать объекты в форме по отношению друг к другу, настраивать размеры и внешний вид элементов управления, а также выполнять другие операции.

Меню **Debug (Отладка)** содержит команды, предназначенные для тестирования и отладки программ пользователя. Команды этого меню позволяют запускать макрос с заданной точки, отслеживать выполнение макроса по шагам и останавливать выполняемую программу в любой момент его выполнения.

Меню **Run (Запуск)** содержит команды, предназначенные для запуска макроса на выполнение прерывания или возобновления его работы, а также для возврата прерванного макроса в начальное его состояние.

Меню **Tools (Сервис)** содержит команды, позволяющие выбрать макрос для выполнения или получить доступ к внешним библиотекам макросов.

Меню **Add-Ins** содержит всего одну команду – Add-In Manager, при выборе которой на экране отображается диалоговое окно Add-In Manager, в котором

можно загружать или выгружать, регистрировать и определять поведение программ-дополнений (надстроек).

Меню **Windows (Окно)** позволяет выбирать активное окно, разбивать текущее, размещать окна вертикально, горизонтально и в виде каскада.

Команды меню **Help (Помощь)** аналогичны командам меню Help в Word, Excel и других приложениях Windows, справочная информация представлена на английском языке.

В VBA имеется возможность использовать контекстное меню. Щелкнув правой кнопкой мыши на любом элементе окна VBA или нажав **Shift + F10**, пользователь увидит меню, содержащее список команд, которые он может применить к данному объекту.

**Панели инструментов.** Наиболее часто используемые команды меню изображены в виде кнопок со значками на панели инструментов. По умолчанию под строкой меню отображается панель инструментов **Standard (Стандартная)** (рисунок 3). Если эта панель отсутствует в главном окне программы, для ее отображения выполните команду *View (Вид) → Toolbars (Панели инструментов) → Standart (Стандартная)*.



Рисунок 3 – Панель инструментов Стандартная

**Панель Debug (Отладка).** Кнопки этой панели позволяют запустить программу на выполнение, проследить за ходом ее работы, а также обнаружить различные ошибки в отлаживаемых программах (рисунок 4).



Рисунок 4 – Панель инструментов Отладка

**Панель Edit (Правка).** Кнопки этой панели инструментов позволяют редактировать текст в окне *Code* (окне программного кода). Они дублируют команды меню *Edit* (рисунок 5).



Рисунок 5 – Панель инструментов Правка

**Панель UserForm (Формы пользователя).** Эта панель используется при проектировании форм. Многие ее кнопки дублируют команды меню *Format* (рисунок 6).



Рисунок 6 – Панель инструментов Формы пользователя

Окно проекта **Project Explorer** показывает иерархическую структуру открытых в данный момент проектов и предоставляет пользователю средства для быстрого доступа к окнам программного кода, пользовательских форм и других объектов (рисунок 7). Окно проекта вызывается с помощью команды *View (Вид) → Project Window (Окно проекта)* или щелчком на кнопке  на панели инструментов *Стандартная* или воспользоваться специальной комбинацией клавиш **Ctrl + R**.

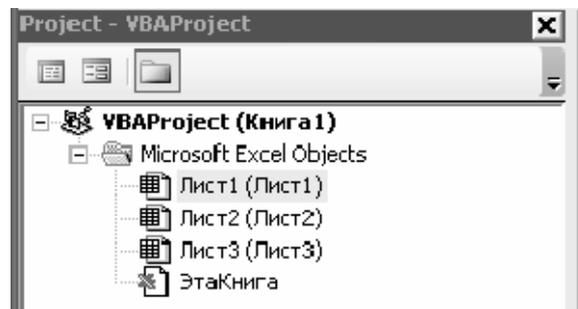


Рисунок 7 – Окно проекта

В проекте автоматически создается модуль для каждого рабочего листа и для всей книги. Кроме того, модули создаются для каждой пользовательской формы, макросов и классов. По своему предназначению модули делятся на два типа: модули объектов и стандартные. К стандартным модулям относятся те, которые содержат макросы. Такие модули добавляются в проект командой *Вставка → Модуль (Insert → Module)*. К модулям объектов относятся модули, связанные с рабочей книгой, рабочими листами, формами, и модули класса.

Окно проекта Project – VBAProject – предназначено для быстрого получения информации о различных составляющих проекта.

Под проектом в редакторе VBA понимается набор всех программных модулей, связанных с документами MS Office (в нашем случае MS Excel).

Проект может включать модули с процедурами обработки событий объектов приложения, стандартные модули с пользовательскими процедурами, модули классов и ссылки на другие проекты, позволяющие использовать их процедуры и данные.

В окне проекта выводится проект всех открытых рабочих книг. Это позволяет легко копировать формы и коды из одного проекта в другой, что убыстряет процесс создания новых приложений.

Окно свойств **Properties Window** используется для просмотра и изменения свойств любого активного в данный момент объекта (проекта, модуля, формы, элемента управления) (рисунок 8).

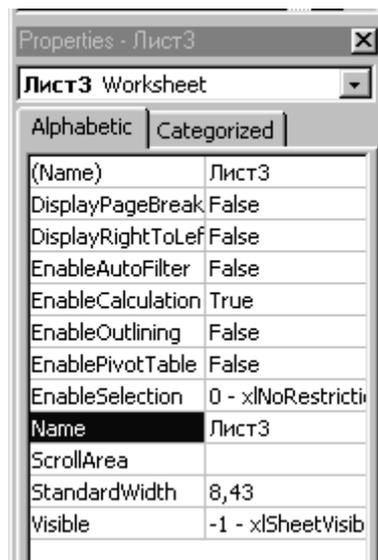


Рисунок 8 – Окно свойств

Открыть окно свойств можно следующим образом:

- с помощью команды *View (Вид) → Properties Window (Окно свойств)*;
- нажатием клавиши **F4**;
- кнопкой  на панели инструментов *Стандартная*.

Свойства можно отображать как в алфавитном порядке, так и по категориям, посредством выбора соответствующих вкладок *Alphabetic* или *Categorized*.

Для изменения свойства выделенного объекта необходимо вызвать *Properties Window*, в появившемся окне в левой колонке выбрать свойство, а в правой колонке изменить его значение. Изменяются значения свойств одним из следующих способов: вводом с клавиатуры значения свойства в соответствующее поле либо значения большинства свойств можно выбрать из раскрывающегося списка. Раскрывающийся список активизируется щелчком в соответствующем поле окна свойств.

*Окно редактирования кода Code* выполняет функции текстового редактора для ввода и изменения процедур и функций проекта (рисунок 9). Каждому объекту в проекте соответствует свое окно кода.

Открыть окно редактирования кода можно следующим образом: сделать двойной щелчок на выбранном элементе управления на форме либо выполнить команду меню *View (Вид) → Code (Программа)* либо нажать клавишу **F7**.

Для выбора конкретной процедуры, которую в данный момент нужно просмотреть или отредактировать в окне программного кода, можно использовать раскрывающиеся списки *Object* и *Procedure*, расположенные в верхней части окна *Code* (см. рисунок 9).

Список *Object* предназначен для выбора из общего списка помещенных в форму объектов конкретного элемента управления или самой формы. В результате в окне программного кода будет отображен текст процедуры для данного объекта.

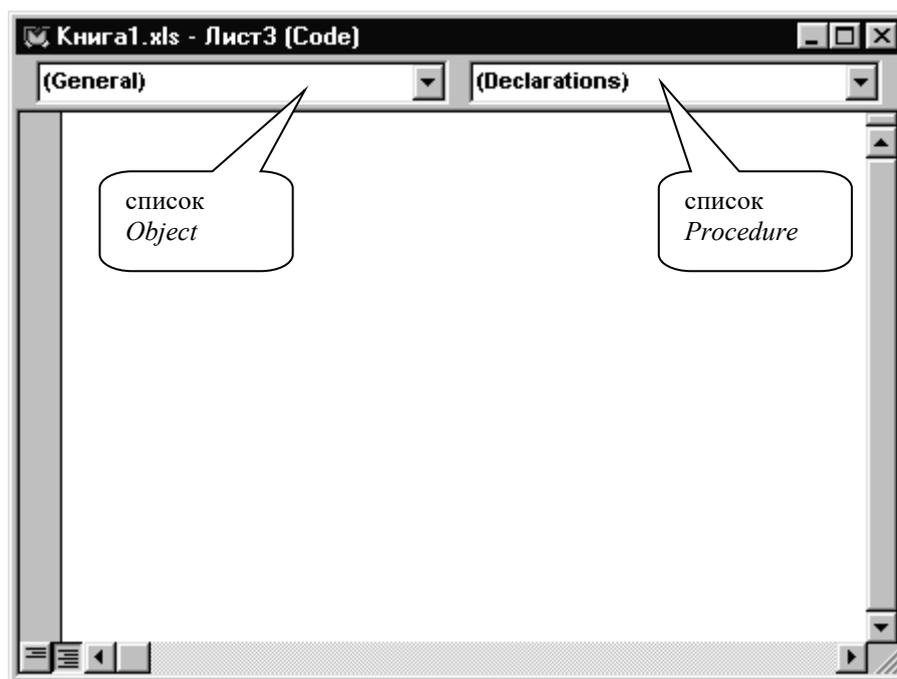


Рисунок 9 – Окно редактирования кода

Список *Procedure* содержит перечень методов обработки событий, допустимых для выбранного в левом списке объекта. При выборе в правом списке некоторого события в окне программного кода появится текст процедуры обработки этого события. Если данная процедура еще не описана, то будет выведена соответствующая заготовка.

**Окна редактирования формы и элементов управления.** В VBA можно использовать пользовательские (настраиваемые) диалоговые окна в создаваемых программах при помощи добавления в проект объекта *UserForm*. Пользовательская форма *UserForm* представляет собой пустое диалоговое окно (рисунок 10), на которое в зависимости от решаемой задачи размещаются нужные элементы управления. Окно элементов управления *Toolbox* открывается при создании новой формы *UserForm*. Окно *Toolbox* можно отобразить с помощью команды меню *View (Вид) → Toolbox (Элементы управления)*.

Используя панель инструментов *Toolbox* из незаполненной формы, можно сконструировать любое требуемое для приложения диалоговое окно. Размещение нового управляющего элемента в форме осуществляется следующей последовательностью действий.

1 Щелкните левой клавишей мыши (ЛКМ) на значке того элемента, который вы собираетесь разместить на форме.

2 Поместите указатель мыши на то место, где будет располагаться выбранный элемент.

3 Нажмите левую кнопку мыши и, не отпуская ее, растяните появившийся прямоугольник до требуемых размеров.

4 Отпустите кнопку мыши.

Размеры формы и расположенных на ней элементов управления можно изменять. Для этого необходимо выделить изменяемый элемент, разместить

указатель мыши на одном из размерных маркеров и протащить его при нажатой левой кнопки мыши так, чтобы объект принял требуемые размеры.

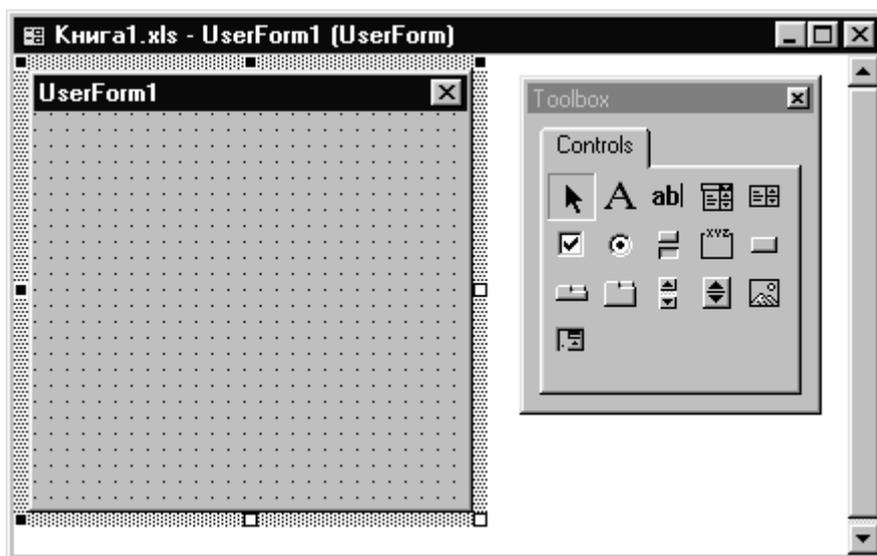


Рисунок 10 – Окно редактирования форм и элементов управления

Окно редактирования форм поддерживает операции буфера обмена. Таким образом, можно копировать, вырезать и вставлять элементы управления, расположенные на поверхности формы. Для облегчения размещения и выравнивания элементов управления используется сетка. Кроме того, команды меню **Формат (Format)** автоматизируют и облегчают процесс выравнивания элементов управления как по их взаимному местоположению, так и по размерам.

#### **Интеллектуальные возможности редактора кода.**

1 Написание программ существенно облегчается за счет способности редактора кода автоматически завершать написание операторов, свойств и параметров. При написании кода редактор сам предлагает пользователю список компонентов, логически завершающих вводимую пользователем инструкцию (рисунок 11). Двойной щелчок на выбранном элементе из этого списка или нажатие клавиши <Tab> вставляет выбранное имя в код программы.

Автоматическое отображение списка компонентов происходит только при установленном флажке **Auto List Members** вкладки **Editor** диалогового окна **Options**, отображаемого на экране выбором команды **Tools → Options**. Список компонентов можно выводить на экран нажатием клавиш **Ctrl + J**.

2 Автоматическое отображение на экране сведений о процедурах, функциях, свойствах и методах после набора их имени (рисунок 12).

Автоматическое отображение на экране сведений о процедурах, функциях, свойствах и методах после ввода их имени происходит только при установленном флажке **Краткие сведения (Auto Quick Info)** вкладки **Редактор (Editor)** диалогового окна **Параметры (Options)**, отображаемого на экране после выбора команды **Сервис → Параметры (Tools → Options)**.

3 Редактор кода также производит автоматическую проверку синтаксиса набранной строки кода сразу после нажатия клавиши <Enter>.

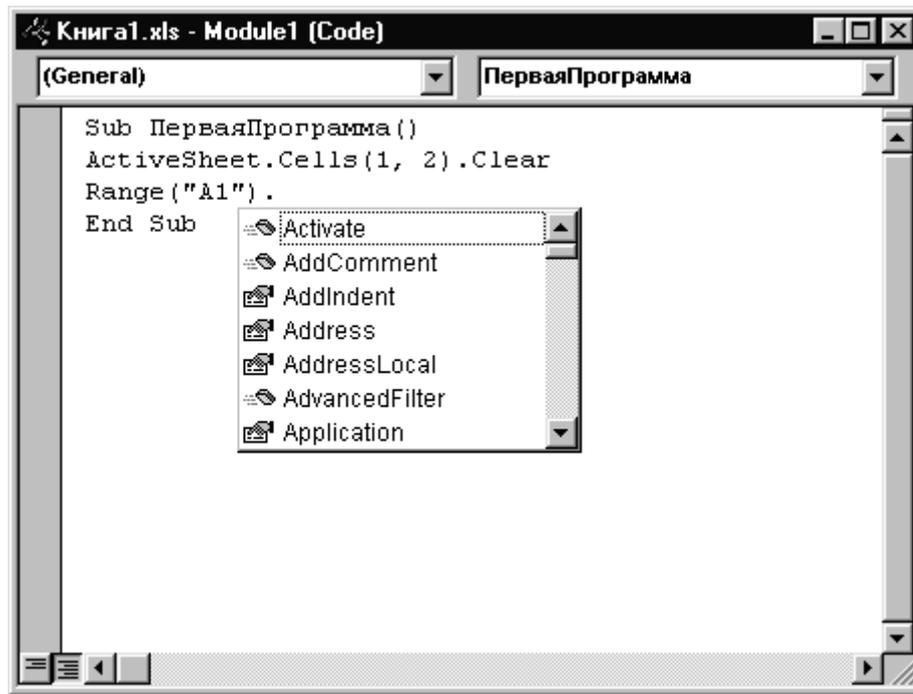


Рисунок 11 – Автоматическое завершение написания свойства

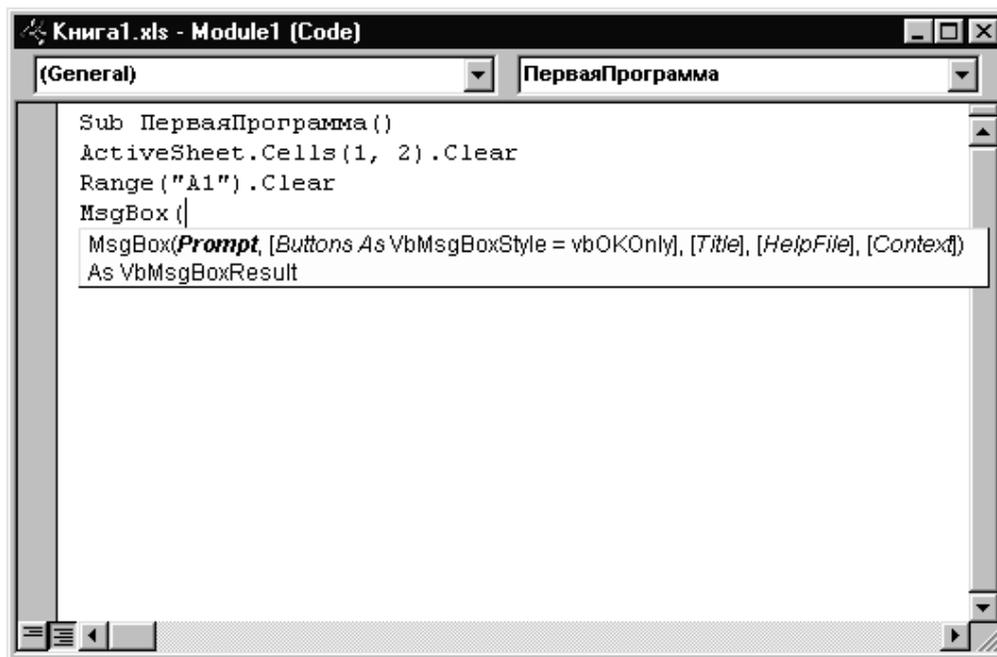


Рисунок 12 – Автоматическое отображение сведений о процедуре

Если после набора строки и нажатия клавиши **<Enter>** строка выделяется красным цветом, то это указывает на наличие синтаксической ошибки в набранной строке.

4 Если курсор расположить на ключевом слове языка VBA, имени процедуры, функции, свойства или метода и нажать клавишу **F1**, то на экране появится окно со справочной информацией об этой функции. Обычно в справке имеется пример использования кода, что позволяет быстрее разобраться в ситуации, которая при написании программы озадачила вас.

**Объекты VBA.** По своей сути форма (или пользовательская форма) представляет собой диалоговое окно, в котором можно размещать различные элементы управления. В приложении может быть как одна, так и несколько форм. Новая форма добавляется в проект выбором команды *Вставка (Insert) → UserForm*.

Одним из основных понятий VBA является объект. В VBA имеется более 100 встроенных объектов (рабочие книги (WorkBook), рабочие листы (WorkSheet), рабочие ячейки (Cell), формы (UserForm), элементы управления (TextBox, CommandButton, Label и др.), диалоги и т. д.).

В VBA имеется обширный набор встроенных элементов управления. Используя этот набор и редактор форм, нетрудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows. Элементы управления создаются при помощи Панели элементов, которая отображается на экране либо выбором команды *Вид (View) → Панель элементов (Toolbox)*, либо нажатием кнопки  панели инструментов *Standard*. На этой панели представлены кнопки, позволяющие конструировать элементы управления. Для создания элементов управления служат все кнопки панели инструментов, за исключением кнопки *Выбор объекта* . Щелкнув по кнопке *Выбор объекта*, можно выбрать уже созданный в форме элемент управления для последующего его редактирования (изменения размеров или редактирования).

Для размещения элемента управления на лист или в форму необходимо нажать соответствующую кнопку на панели элементов и с помощью мыши перетащить рамку элемента управления в нужное место. После этого элемент управления можно перемещать, изменять его размеры, копировать в буфер обмена, вставлять из буфера обмена и удалять из формы.

Список основных элементов управления и соответствующих кнопок панели элементов приведен в таблице 1.

Таблица 1 – Элементы управления

Элемент управления	Имя элемента управления	Создающая его кнопка
Поле	<i>TextBox</i>	
Надпись	<i>Label</i>	
Кнопка	<i>CommandButton</i>	
Список	<i>ListBox</i>	
Поле со списком	<i>ComboBox</i>	

Окончание таблицы 1

Элемент управления	Имя элемента управления	Создающая его кнопка
Полоса прокрутки	<i>ScrolBar</i>	
Счетчик	<i>SpinButton</i>	
Переключатель	<i>OptionButton</i>	
Флажок	<i>CheckBox</i>	
Выключатель	<i>ToggleButton</i>	
Рамка	<i>Frame</i>	
Рисунок	<i>Image</i>	
Набор страниц	<i>MultiPage</i>	

Элементы управления являются объектами. Как любые объекты, они обладают свойствами, методами и событиями.

**Свойства объектов.** Каждый объект обладает некоторыми свойствами. Изменяя свойства, можно менять характеристики объекта. Таким образом, *свойство* представляет собой атрибут объекта, определяющий его характеристики, такие, как размер, цвет, положение на экране и состояние объекта, например, доступность или видимость.

Синтаксис применения свойства: *Объект.Свойство*.

Основные общие свойства элементов управления приведены в таблице 2.

Таблица 2 – Общие свойства элементов управления

Свойство	Описание
Caption	Надпись, отображаемая при элементе управления
AutoSize	Допустимые значения: True (устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption) и False (в противном случае)
Visible	Допустимые значения: True (элемент управления отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь вручную может управлять элементом управления) и False (в противном случае)
Height и Width	Устанавливают геометрические размеры объекта (высоту и ширину)

## Окончание таблицы 2

Свойство	Описание
Left и Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме
ControlTipText	Устанавливает текст в окне всплывающей подсказки, связанной с элементом управления. Например, элементу управления <code>CommandButton</code> назначен текст, всплывающей подсказки "Это кнопка": <code>CommandButton1.ControlTipText = "Это кнопка"</code>
BackColor, ForeColor, BorderColor	Устанавливают цвет заднего и переднего плана элемента управления, также его границы
BackStyle	Устанавливает тип заднего фона
BorderStyle	Устанавливает тип границы. Допустимые значения: <code>fmBorderStyleSingle</code> (граница в виде контура); <code>fmBorderStyleNone</code> (граница невидима)

Описание назначения элементов управления и наиболее часто используемые свойства приведены в таблице 3.

Таблица 3 – Элементы управления и их свойства

Свойство	Описание
<b>TextBox</b> (поле)	используется для ввода текста пользователем или для вывода в него результатов расчетов программ
Text	Возвращает текст, содержащийся в поле
Multiline	Допустимые значения: <code>True</code> (устанавливает многострочный режим ввода текста в поле) и <code>False</code> (однорядочный режим)
WordWrap	Допустимые значения: <code>True</code> (устанавливает режим автоматического переноса) и <code>False</code> (в противном случае)
<b>Label</b> (надпись)	используется для отображения надписей, например, заголовков элементов управления, не имеющих свойства <code>Caption</code>
Caption	Возвращает текст, отображаемый в надписи
Multiline	Допустимые значения: <code>True</code> (устанавливает многострочный режим ввода) и <code>False</code> (однорядочный режим)
WordWrap	Допустимые значения: <code>True</code> (устанавливает режим автоматического переноса) и <code>False</code> (в противном случае)
<b>CommandButton</b> (кнопка)	используется для инициирования выполнения некоторых действий, вызываемых нажатием кнопки, например запуск программы или остановка ее выполнения, печать и т. д.
Caption	Возвращает текст, отображаемый на кнопке
Cancel	Допустимые значения: <code>True</code> (устанавливаются отменяющие функции для кнопки, т. е. нажатие клавиши <code>&lt;Esc&gt;</code> приводит к тем же результатам, что и нажатие кнопки) и <code>False</code> (в противном случае)
Default	Задаёт кнопку по умолчанию, т. е. кнопку, для которой действия, связанные с ней, будут выполняться при нажатии клавиши <code>&lt;Enter&gt;</code>

## Окончание таблицы 3

Свойство	Описание
<b>Frame</b> (рамка)	используется для визуальной группировки элементов управления
Capture	Отображает надпись рамки
<b>OptionButton</b> (переключатель)	позволяет выбрать один из нескольких взаимоисключающих параметров. Переключатели обычно отображаются группами, обеспечивая возможность выбора альтернативного варианта
Capture	Надпись, отображаемая рядом с переключателем
Value	Возвращает True, если переключатель выбран и False в противном случае

**Метод.** Объект содержит также список методов, которые к нему могут быть применены. Например, показать форму на экране или убрать его можно с помощью методов Show и Hide соответственно. Таким образом, метод представляет собой действие, выполняемое над объектом.

Синтаксис применения метода: *Объект.Метод*.

Таким образом, объект – это программный элемент, который имеет свое отображение на экране, содержит некоторые переменные, определяющие его свойства, и некоторые методы для управления объектом. В таблице 4 перечислены основные общие методы элементов управления.

Таблица 4 – Основные общие методы элементов управления

Метод	Описание
Add	Позволяет добавить элемент управления во время выполнения программы
Move	Перемещает элемент управления
SetFocus	Устанавливает фокус на вызвавшем этот метод элементе управления
Zorder	Помещает объект до или после всех пересекающихся с ним объектов

**Событие** представляет собой действие, распознаваемое объектом (например, щелчок мышью или нажатие клавиши), для которого можно запрограммировать отклик. События возникают в результате действий пользователя программы, или же они могут быть вызваны системой.

Процедуры обработки событий имеют следующий синтаксис:

*Sub UserForm\_Событие()*

*Последовательность инструкций*

*End Sub*

В таблице 5 приведены события элементов управления, для которых можно создать процедуры обработки событий. Каждый элемент управления, который вы добавите в свою форму, будет иметь доступ к этим событиям.

Таблица 5 – События элементов управления

Событие	Описание
Click	Происходит, когда пользователь выбирает элемент управления с помощью одинарного щелчка кнопкой мыши
DblClick	Происходит, когда пользователь выбирает элемент управления с помощью двойного щелчка кнопкой мыши
Change	Происходит при изменении значения элемента управления
GotFocus	Происходит, когда элемент управления получает фокус
LostFocus	Происходит, когда элемент управления теряет фокус
Error	Используется при уведомлении об ошибке
Exit	Происходит, когда с элемента управления снимается выделение

Для связывания объекта с кодом выполнения данной операции необходимо дважды щелкнуть ЛКМ по элементу управления в форме. Появляется окно модуля для выбранного объекта в котором необходимо выбрать событие для которого требуется создать процедуру обработки, в списке, расположенном в верхнем правом углу окна модуля и ввести текст процедуры.

## 2 Примеры выполнения задания

**Пример 1** – Найти сумму  $a + b = c$ .

**Порядок выполнения работы.**

- 1 Запустить редактор Visual Basic (сочетание клавиш **ALT + F11**).
- 2 Выполнить команду *Вставка (Insert)* → *UserForm*.
- 3 Поместить на форму элементы, требуемые для решения задачи, с панели элементов, и расположить их нужным образом, как показано на рисунке 13.

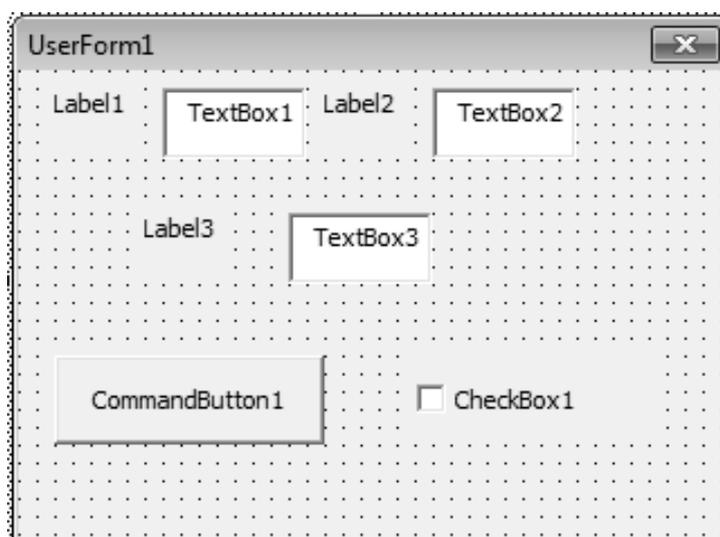


Рисунок 13 – Вид пользовательской формы

4 Изменить свойства объектов (таблица 6) на форме с помощью окна свойств. Вид пользовательской формы после назначения свойств, представлен на рисунке 14.

Таблица 6 – Свойства объектов

Свойство	Значение
Label1.Caption	<i>a</i>
Label2.Caption	<i>b</i>
Label3.Caption	<i>c</i>
CommandButton1.Caption	<i>“результат”</i>
CheckBox1.Caption	<i>“очистка окон”</i>
Для всех объектов свойство .BackColor	Выбрать цвет из палитры цветов
Для Label1, Label2 ,Label3 свойство .Font	В ДО «Шрифт», которое появится после щелчка по <i>кнопке с изображением трех точек</i> , расположенной напротив свойства Font в окне свойств, выбрать размер 16

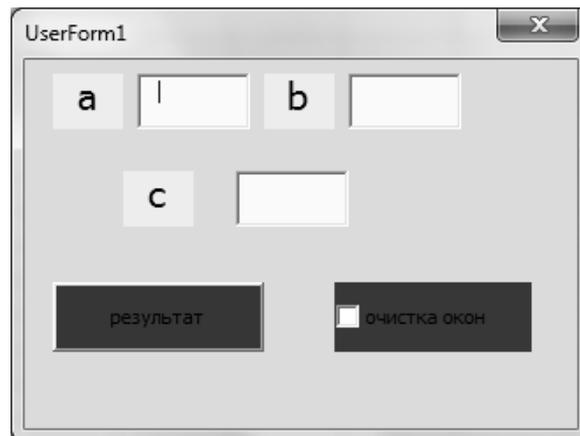


Рисунок 14 – Вид пользовательской формы после назначения свойств

5 Написать программный код. Для этого рекомендуется выполнить двойной щелчок по кнопке *«результат»* и перейти в окно программы, где набрать текст процедуры обработки события Click() для кнопки CommandButton1:

```
Private Sub CommandButton1_Click()
    Dim a As Integer, b As Integer
    Dim c As Integer
    a = CInt(TextBox1.Text)
    b = CInt(TextBox2.Text)
    c = a + b
    MsgBox "результат смотри в TextBox3"
    TextBox3.Visible = True
End Sub
```

```

TextBox3.Text = c
End Sub

```

6 Двойной щелчок по элементу управления CheckBox1 вызовет процедуру обработки события Click(), где необходимо написать программный код для очистки полей TextBox:

```

Private Sub CheckBox1_Click()
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox3.Visible = False
    TextBox1.SetFocus
    CheckBox1.Value = False
End Sub

```

**Пример 2** – Разработать программу выполнения одной из двух арифметических операций над двумя числами по выбору пользователя. Исполняемая операция устанавливается за счет выбора соответствующего переключателя.

**Порядок выполнения работы.**

- 1 Запустите редакторы Visual Basic (сочетание клавиш ALT + F11).
- 2 Выполнить команду *Вставка (Insert) → UserForm*.
- 3 Поместить на форму элементы, требуемые для решения задачи, с панели элементов, и расположить их нужным образом, как показано на рисунке 15.

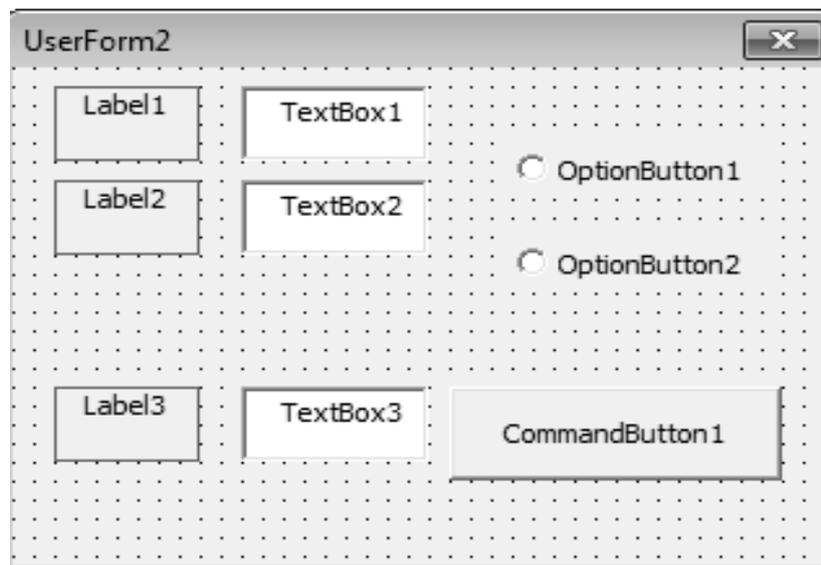


Рисунок 15 – Вид пользовательской формы

4 Изменить свойства объектов (таблица 7) на форме с помощью окна свойств. Вид пользовательской формы после назначения свойств, представлен на рисунке 16.

Таблица 7 – Свойства объектов

Свойство	Значение
Label1.Caption	<i>a</i>
Label2.Caption	<i>b</i>
Label3.Caption	<i>c</i>
CommandButton1.Caption	“ Расчет”
OptioButton1.Caption	“ сложение”
OptioButton2.Caption	“ вычитание”
Для всех объектов свойство .BackColor	Выбрать цвет из палитры цветов
Для Label1, Label2 ,Label3 свойство .Font	В диалоговом окне «Шрифт», которое появится после щелчка по <i>Кнопке с изображением трех маленьких точек</i> , расположенной напротив свойства Font в окне свойств, выбрать размер 16

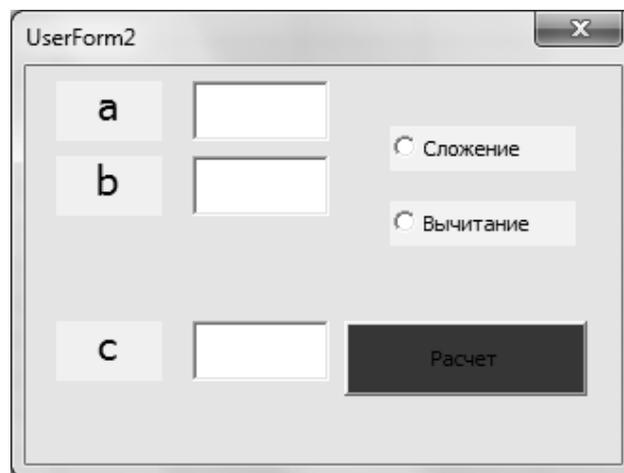


Рисунок 16 – Вид пользовательской формы после назначения свойств

5 Написать программный код. Для этого рекомендуется выполнить двойной щелчок по кнопке *Расчет* и перейти в окно программы, где набрать текст процедуры обработки события Click() для кнопки CommandButton1:

```
Private Sub CommandButton1_Click()
Dim a As Integer, b As Integer, c As Integer
a = TextBox1.Value
b = TextBox2.Value
If OptionButton1.Value = True Then c = a + b
If OptionButton2.Value = True Then c = a - b
TextBox3.Value = c
End Sub
```

### 3 Задания для самостоятельного выполнения

#### Вариант 1

Разработать пользовательскую форму для вычисления длины гипотенузы по заданным длинам катетов.

#### Вариант 2

Разработать пользовательскую форму для нахождения площади грани и объёма куба по заданной стороне.

#### Вариант 3

Разработать пользовательскую форму, в которой запрашивается радиус круга, рассчитывается его площадь.

#### Вариант 4

Разработать пользовательскую форму, в которой по знаку арифметической операции (+, -, ·, :, =) выводится ее название.

#### Вариант 5

Разработать пользовательскую форму, определяющую четность или нечетность введенного числа.

#### Вариант 6

Разработать пользовательскую форму для нахождения минимального из трех введенных чисел.

#### Вариант 7

Разработать пользовательскую форму для нахождения максимального из трех введенных чисел.

#### Вариант 8

Разработать пользовательскую форму, в которой можно определить принадлежит ли точка  $M(x, y)$  кругу с центром в начале координат и радиусом  $R$ , вводимым с формы.

#### Вариант 9

Разработать пользовательскую форму, для вычисления синуса, косинуса, тангенса или котангенса по введенному значению угла.

#### Вариант 10

Ввести число  $X$  и определить, делится ли оно на 2, используя пользовательскую форму.

#### Вариант 11

Разработать пользовательскую форму для вычисления

$$c = \begin{cases} x^3 + y^2 + \cos x, & \text{если } y - x = 0; \\ \sqrt[5]{(x + y)^2} + \sin(x + 1), & \text{если } y - x > 0; \\ (y + x)^4 + \operatorname{tg}^2 x, & \text{если } y - x < 0. \end{cases}$$

#### Вариант 12

Разработать пользовательскую форму для вычисления значения функции

$$Z = f(T, P), \text{ если } T = \sin(x + y), P = \frac{x}{x - y}, Z = \cos\left(\frac{T}{P}\right), \text{ для } x \neq y.$$

*Вариант 13*

Разработать пользовательскую форму для расчета длины окружности или площади круга по введенному радиусу.

*Вариант 14*

Разработать пользовательскую форму для определения корней квадратного уравнения по введенным коэффициентам.

*Вариант 15*

Разработать пользовательскую форму для определения времени нахождения в пути пешехода, велосипедиста или автомобиля для прохождения заданного расстояния. Скоростью задаться самостоятельно.

*Вариант 16*

Разработать пользовательскую форму для вычисления значения функции по введенным аргументам  $f(x, y) = \frac{x}{1+y} + \frac{y}{1+x} + \frac{1}{x+y}$ .

*Вариант 17*

Разработать пользовательскую форму для вычисления значения функции по введенным аргументам  $f(x, y, z) = \frac{x+y+z}{x \cdot y \cdot z}$ .

*Вариант 18*

Разработать пользовательскую форму для нахождения периметра или площади прямоугольника по заданным сторонам.

*Вариант 19*

Разработать пользовательскую форму для вычисления значения функции по введенным аргументам  $f(x, y, z) = \frac{x+y+z}{x^2+y^2+z^2}$ .

*Вариант 20*

Разработать пользовательскую форму для вычисления

$$a = \begin{cases} (x-y)^2 - \sqrt{x \cdot y}, & \text{если } x \cdot y > 0; \\ (x+y)^2 - 2,5, & \text{если } x \cdot y < 0; \\ (x-y)^2 + 1, & \text{если } x \cdot y = 0. \end{cases}$$

*Вариант 21*

Разработать пользовательскую форму для вычисления

$$b = 1 + |y-x| + \frac{\sqrt[5]{(y-x)^2}}{2} + \frac{|y-x|^3}{3}.$$

*Вариант 22*

Разработать пользовательскую форму для вычисления

$$c = \begin{cases} x^3 + y^3 + \sin x, & \text{если } x - y = 0; \\ \sqrt[3]{(x - y)^2} + \cos x, & \text{если } x - y > 0; \\ (y - x)^2 + \operatorname{tg}^3 x, & \text{если } x - y < 0. \end{cases}$$

*Вариант 23*

Разработать пользовательскую форму для перевода расстояния, заданного в метрах, в километры, мили, футы или ярды на выбор пользователя (1 миля = 1,609 километра, 1 метр = 1,094 ярда, 1 метр = 3,281 фута).

*Вариант 24*

Разработать пользовательскую форму для вычисления произведения или суммы трех введенных чисел.

*Вариант 25*

Разработать пользовательскую форму для перевода объема, заданного в литрах, в объем в пинтах, галлонах, бушелях и квартах (английские меры объема жидких и сыпучих тел) на выбор пользователя (1 литр = 1,706 пинты, 1 литр = 0,220 галлона, 1 бушель = 36,35 литра, 1 кварта = 1,136 литра).

## Список литературы

1 **Хорев, П. Б.** Объектно-ориентированное программирование: учебное пособие / П. Б. Хорев. – 4-е изд., стер. – Москва: Академия, 2012. – 448 с.

2 **Нагина, Е. К.** Visual Basic for Applications. Практика: учебно-методическое пособие по информатике для вузов / Е. К. Нагина, В. А. Ищенко. – Воронеж: ВГУ, 2015. – 88 с.

3 **Васильев, А. Н.** Excel 2010 на примерах / А. Н. Васильев. – Санкт-Петербург: БХВ-Петербург, 2010. – 432 с.