

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

К а ф е д р а «Электромеханики и автомобильного электрооборудования»

С.П.МИНЕЕВ

Методические указания к лабораторным работам № 1-19

по дисциплине «Информатика»

Самара Самарский государственный технический университет 2015 Печатается по решению кафедры Электромеханика и Автомобильное Электрооборудование

УДК

Лабораторная работа №1-19: метод. указ. / Сост. С.П. Минеев. – Самара: Самар. гос. техн. ун-т, 2015. – 193 с.: ил.

Предназначены для бакалавров, обучающихся по направлению 140400.

Рецензент

УДК

© С.П. Минеев, составление, 2015 © Самарский государственный технический университет, 2015

1.1. Введение

Создать прикладную программу, дающую действительно нужный результат, может лишь человек, работающий непосредственно в той области, для которой она пишется. Только специалист знает все тонкости своего дела и сможет предусмотреть в программе все необходимые условия. Например, какая программа для расчета характеристик двигателя будет лучше: выполненная опытным специалистом в области электротехники, и прекрасно знающим ее, или написанная опытным программистом, хорошо разбирающимся в языках программирования, но незнакомым с электротехникой? Даже если инженер и обратится к программисту с просьбой помочь ему написать программу, то он все равно не сможет посвятить того во все тонкости своего дела — на это уйдет слишком много времени и сил. Поэтому инженер, знакомый с программированием, больше подойдет для написания программы, чем профессиональный программист, не знающий электротехники.

В настоящее время существуют среды программирования, овладеть которыми без проблем может каждый. Мы будем изучать такую среду программирования – VBA (Visual Basic for Aplication). В этой среде любой специалист сумеет использовать в своей работе новейшие достижения информационных технологий, чтобы написать нужную ему программу, разработав ее алгоритм на основе собственного опыта.

Преимуществом использования VBA является то, что данный язык программирования интегрирован (встроен) практически во все программы пакета Microsoft Office, который имеется практически на любом компьютере с операционной системой Windows. Этим достигается универсализм в решении различных задач. Так, если надо сформировать отчет с формулами, таблицами, графиками для какогонибудь расчета, можно использовать VBA для Word. Если нужно обработать таблицы, то можно использовать VBA для Excel. Для созда-

ния и обработки баз данных имеется VBA для Access. Кроме того, данная среда программирования интегрирована в AutoCAD и Corel-Draw, что позволяет автоматизировать построение чертежей.

Несмотря на различное назначение этих программных продуктов, структура языка программирования в этих приложениях остается той же самой. Отличие состоит в том, что VBA оперирует, различными объектами, свойственными этим приложениям. Так, в Word'е VBA работает с текстом, таблицами и другими элементами, которые могут содержаться в документе. В AutoCAD'е работа идет с графическими объектами – линиями, окружностями и т.д.

Подводя итог, можно сказать, что с помощью VBA можно решить практически любую поставленную инженерную задачу.

Стоит сказать об использовании программирования на VBA в процессе обучения в школе, институте, да и в самостоятельном образовании. Даже если человек и не намерен посвятить свою жизнь информационным технологиям, все равно умение программировать ему всегда пригодится, так как ПК сейчас применяется почти в любой области человеческой деятельности.

1.2. Интерфейс среды VBA

Для изучения VBA будем использовать текстовый редактор **Word**. Изучение следует начать со среды программирования. Из Word'а данную среду можно вызвать несколькими способами:

- нажав на клавиатуре комбинацию клавиш [Alt+F11];

- выбрать вкладку Разработчик – Редактор Visual Basic;

Если вкладка Разработчик отсутствует, нужно сделать следующее: нажать вкладку Файл – Параметры – Настройка ленты – поставить галочку напротив пункта Разработчик – нажать кнопку Ок (см. Рис..).



Рис. 1.1. Вызов редактора программирования



Рис. 1.2. Рис. Включение вкладки разработчик

После вызова редактора с помощью вышеперечисленных способов, на экране появится окно среды программирования (см. рис. 3)



Рис. 1.3. Рис. 1. Среда программирования VBA

Среда программирования состоит из следующих частей: меню, панелей инструментов, окна проекта, окна свойств, окна кода программ, окна дизайна форм, и некоторых других. Каждое из этих окон можно как включить, так и выключить, тем самым, освобождая рабочее пространство редактора.

Каждый документ или шаблон, загруженный в данный момент (в том числе автозагружаемые), отображаются в окне проектов. Автозагружаемые документы и шаблоны загружаются из специальной папки при загрузке Word. При этом в окне проектов они не доступны для редактирования, кроме шаблона Normal.dot.

Код (текст) программ, который пишет программист, размещается в модулях и в коде форм. Модули и формы отображаются в окне проектов. В новом документе (проекте) и формы и модули отсутствуют. Их можно вставить в проект с помощью меню редактора VBA, раздела Insert (вставить).

Модуль (Module) - это контейнер для текста программ, который может содержать любое количество процедур и функций.

Пользовательская форма (UserForms) - это программа, которая выглядит как обычное окно других приложений и предназначенное для организации диалога между пользователем и программой. Этот диалог выглядит со стороны пользователя, как нажатие на кнопку с помощью мыши, выбор элемента из выпадающего списка, ввод данных в текстовое поле, включение-выключение флажков и использование других элементов управления. Программа общается с пользователем с помощью этих же элементов управления.

При разработке дизайна формы эти элементы помещаются на форму из панели элементов управления. Стандартных элементов на этой панели немного, но их достаточно для разработки вполне серьезного приложения. Кроме того, можно использовать дополнительные элементы. Для этого нужно навести на панель курсор мыши, нажать правую клавишу мыши и из появившегося контекстного меню выбрать пункт "Additional controls". После этого действия появится окно с этим же названием (см. рис. 4).

L Microsoft I	abStrip Control, vers	ion 5.0 (SP2) 🔨		OK
3 Microsoft 1	erminal Services Co	ntrol		
3 Microsoft 1	oolbar Control 6.0 (S	iP4)		Cancel
3 Microsoft 1	oolbar Control, versi	on 5.0 (SP2)	_	
Microsoft 1	reeView Control 6.0	(SP4)		
Microsoft T	reeView Control, ver	sion 5.0 (SP2		
🗆 Microsoft L	IpDown Control 6.0 ((SP4)		
□ Microsoft \	VinSock Control, ver	sion 6.0 👘		
MMC Icon	Control class			
MMCCtrl c	ass			
D MoveBvr 0	lass		Show	
I MPSButtor	hAx Class	~	E Colored	have Out
1		>	i selected	items Unly

Рис. 1.4. Рис. 4. Окно выбора дополнительных элементов

В этом окне для нужного элемента нужно поставить галочку и нажать ОК. После чего выбранный элемент появится на панели элементов управления.

С выбором дополнительного элемента управления нужно быть осторожней. Во-первых, вполне вероятно, что для данного элемента отсутствует справочная информация, из-за чего будет сложно писать код для данного элемента. Во-вторых, данный элемент, зарегистрированный на данном компьютере, на котором пишется программа, но он может быть не зарегистрирован на другом компьютере, вследствие чего данная программа, просто не будет работать на другой машине.

Каждый элемент управления имеет свои свойства. Эти свойства отображаются в окне свойств.

Основное свойство для элементов управления, в том числе и для формы, является имя (**Name**). Имя для элементов управления обычно начинается с префикса, состоящего из трех букв, и собственно имени объекта. **Префикс** – это сокращенное название элемента. Использование префикса элементов в программах улучшает читаемость кода программы (понятно с каким типом элемента управления идет работа в данный момент), что в свою очередь является хорошим стилем программирования. Рассмотрим примеры названия элементов управления (см. табл.).

Название элемента		Префикс	Пример свой-
			ства Name
Form (Форма)		frm	frmSample
Label (Надпись)	Α	lbl	lblPick
TextBox (Текстовое	abl	txt	txtEdit
окно)			
ComboBox		cbo	cboChoose
(Поле со списком)			
ListBox		lst	lstSpisok
(Список)			
Check Box (Флажок)		chk	chkTop
Option Button (Кнопка	•	opt	optBlue
выбора)			
ToggleButton		tog	togVkluch

Таблица № 1.1 Таблица №1 Элементы управления формы

(Выключатель)			
Frame (Рамка, группа)	[XVZ]	fra	fraStud
CommandButton		cmd	cmdBeep
(Кнопка)			
ScrollBar (Полоса про-	A F	scr	scrPolosa
крутки)			
SpinButton (Счетчик)		spn	spnSchet
Immage (Рисунок)		Img	imgPicture

2. Лабораторная работа №1

2.1. Создание простейшей линейной программы

Цель работы:

- 1. Создавать простейшие линейные программы.
- 2. Использовать элементы управления Форма, Кнопка, Надпись
- 3. Изучить понятия: Объект, Свойство объекта, Метод, Событие, префикс,
- 4. Изучить операторы: Sub, End Sub, Dim, MsgBox, оператор присваивания
- 5. Изучить интерфейс программы: Окно проекта, Окно свойств, Окно проектирования формы, Панель элементов, Окно кода программы, списки объектов, списки свойств
- 6. Изучить типы данных
- 7. Научиться организовывать простейший вывод информации в документ.

Порядок выполнения работы

После изучения среды программирования перейдем к практической части и создадим первую программу на языке VBA. Эта программа будет называться «Здравствуй Мир». При изучении многих языков программирования дается эта программа, как первая. Создадим ее и мы.

Выполните следующие действия.

1. Загрузите редактор Word.

2. Перейдите в редактор VBA одним из вышеперечисленных способов.

3. С помощь меню Insert вставьте в проект новую форму (UserForm).

4. Поместите на форму кнопку (CommandButton) из панели элементов управления.

5. Также поместите на форму метку (Label).

6. Сделайте форму активной, нажав на левую клавишу мыши при наведенном на форму курсоре. Признаком активности формы будет то, что на форме по углам и серединам сторон появятся рычаги управления в виде маленьких квадратиков. При этом в окне свойств будут отображаться свойства этой формы.

7. В окне свойств измените свойство Name с UserForm1 на frm-FirstProgram. Свойство Caption измените на «Моя первая программа». Это свойство меняет заголовок формы. В свойстве формы больше ничего менять не надо.

8. Сделайте активной кнопку. В окне свойств теперь отображаются свойства кнопки. Задайте свойству Name кнопки cmdRunMe. Свойство Caption измените на «Нажми меня». После этого вы заметите, что изменилась надпись на кнопке.

9. Теперь активируйте метку. Задайте свойству Name метки lblMessage. Свойство Caption сделайте пустым, при этом на метке не станет надписи.

После выполненных действий, форма должны выглядеть следующим образом (см. рис. 5):

	•	•				-	•	•	•							-	•	•	•		-																-	-	-	-	
•	•	•	•	•	2		•	•	•	•	•	•		Ċ	8		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
			-				s.,										s.,				-																				
•	•	ł				1															U																1		•		
•	•	Ť	Ξ																																		ř	۳.	•	•	•
•	•	2																																			5	۳.	•	•	•
•	•	Ĩ	1			1			1	10	10	10							0	10		10	10	10					0		0			0		0	Ľ	Ξ.	•	•	
		1																																				0			1
	•						•		•									•										•		•		•		•							
•	•	•	•	•			•	•	•	•	•	•				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•
•	•	•	•	•	1		•	•	•	•	•	•	•	•	2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	2		•	•	•	•	•	•			8	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•							•		•		•		1				•		•			•		•	•		•	•	•			•		•				•	•	•	
															3																										
•	•	•					•	•	•	•	•																	•	•	•	•	•		•	•	•					
•	•	•	•	•			•	•	•	•	•	•	•				6										Ł	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•			•	•	•	•	•	•	•				F	la	D#	٢M	И	Μ	e	15			Ł	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•			•	•	•	•	•	•	1	8													1	•	•	•	•	•	•	•	•	•	•	•	•	•	•
			:	:			:			:	:	:																:	:	:		:		:	:		:	:	:	:	
															9																										
															8																										
	•	•						•	•		•				2	•		•	•				•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•		

Рис. 2.1. Рис. 5. Форма с эле-

ментами управления.

Таблица № 2.1 Таблица №

Окно свойств

Объект	Свойство	Значение
	Name	frmFirstProgram
UserForm1	Caption	Моя первая программа
CommandButton1	Name	cmdRunMe
	Caption	Нажми меня
Label1	Name	lblMessage
	Caption	(оставить пустым)

Теперь, цель нашей программы состоит в том, чтобы при нажатии кнопки появлялась надпись «Здравствуй Мир!» в трех местах: на метке, в документе Word, и в стандартном окне сообщений. Нажатие на кнопку является **событием**, которое нужно обработать. В нашем случае помещение надписи «Здравствуй Мир!» в различные объекты и есть обработка событий.

10. Щелкните два раза курсором мыши на кнопку cmdRunMe. Появится окно кода (см. рис. 6)



Рис. 2.2. Рис. 6. Окно кода.

В редакторе кода появится две строки:

Private Sub cmdRunMe_Click()

End Sub

Это первая и последняя строчка обработки **события** нажатия кнопки. Первая строчка говорит о том, что это процедура (Private Sub) обработки события (cmdRunMe_Click()). Вторая строчка (End Sub) завершает обработку события. Между этими двумя строчками помещается код обработки события. В нашем случае - это вывод надписи «Здравствуй Мир!». Но прежде, чем переходить к набору кода изучим поподробнее редактор кода.

В левой верхней части редактора есть выпадающее поле со списком существующих объектов. Названия объектов берутся из свойства **Name**. В нашем случае этих объектов всего три: форма, кнопка и метка (см. рис. 7).

🗷 Лекции и лабораторные - UserForm1 (Code)	
cmdRunMe 🗸	Click
(General)	-
Label1	
	_

Рис. 2.3. Рис. . Окно кода – список объектов

Выбирая объект из списка, мы сообщаем редактору кода, с обработкой события какого объекта будем в данный момент работать.

В правой верхней части есть выпадающее поле со списком, в котором содержатся варианты событий для текущего элемента. Если мы в данный момент обрабатываем **событие** нажатия кнопки, и при этом нажмем на кнопку выбора события, то появятся варианты событий именно для этой кнопки (см. рис.).



Рис. 2.4. Рис. . Окно кода – список событий

Как видно из рисунка, кроме события нажатия курсором мыши на кнопку (событие Click), имеется множество других событий: двойное нажатие курсором мыши на кнопку (DblClick), перемещение курсором мыши над кнопкой (MouseMove) и другие. Но основным событием для кнопки, конечно же, является именно событие Click, которое мы и обрабатываем в нашей программе.

Для каждого элемента, в том числе и для формы, имеются свои возможные события, часто одинаковые, но есть и свои отличия.

Вернемся к нашей программе.

11. В коде обработке события cmdRunMe_Click() наберем следующий текст:

```
Private Sub cmdRunMe_Click()
Dim Str As String
Str = "Здраствуй мир!"
```

```
lblMessage.Caption = Str
MsgBox Str
frmFirstProgram.Hide
Documents.Add
Selection.TypeText Str
frmFirstProgram.Show
End Sub
```

Этот код решает все поставленные задачи.

12. Теперь мы запускаем нашу программу, нажав на кнопку 🖃 из панели инструментов.

Наша спроектированная форма появится на экране (Рис.). Она будет ждать, когда мы нажмем на кнопку.

13. Нажимаем на кнопку.

После нажатия на кнопку курсором мыши, произойдет следующие: надпись "Здравствуй мир!" сначала появится на метке, затем появится новое окно, с этой же надписью и кнопкой ОК (Рис.) и, наконец, надпись появится в документе (Рис.).

Зарастауй кир! Зарастауй кир!

Рис. 2.5. Рис.



Рис. 2.6. Рис.

Рассмотрим каждую строчку кода подробнее.

Dim Str As String

Оператором **Dim** мы объявляем переменную **Str**. **As String** означает, что переменная Str имеет строковой (текстовый) тип. Кроме текстового типа существует множество других типов. В следующей таблице перечислены поддерживаемые типы данных, размеры, требуемые для сохранения значений, и диапазоны допустимых значений.

Таблица № 2.2 Таблица №2.

Тип данных	Размер	Диапазон значений
Вуtе (байт)	1 байт	От 0 до 255.
Boolean (логиче-	2 байт	True или False.
ский)		
Integer (целое)	2 байт	От -32 768 до 32 767.
Long(длинное це-	4 байт	От -2 147 483 648 до 2 147 483 647.
лое)		
Single (с плаваю-	4 байт	От -3,402823Е38 до -1,401298Е-45 для
щей точкой		отрицательных значений; от

Типы данных

обычной точно-		1,401298Е-45 до 3,402823Е38 для по-
сти)		ложительных значений.
Double (с плава-	8 байт	От -1,79769313486232Е308 до -
ющей точкой		4,94065645841247Е-324 для отрица-
двойной точно-		тельных значений; от
сти)		4,94065645841247Е-324 до
		1,79769313486232Е308 для положи-
		тельных значений.
Currency (денеж-	8 байт	От -922 337 203 685 477,5808 до
ный)		922 337 203 685 477,5807.
Decimal (масшта-	14 байт	+/-
бируемое целое)		79 228 162 514 264 337 593 543 950 335
		без дробной части; +/-
		7,9228162514264337593543950335 c 28
		знаками справа от запятой; минималь-
		ное ненулевое значение имеет вид +/-
		0,0000000000000000000000000000000000000
Date (даты и вре-	8 байт	От 1 января 100 г. До
мя)		31 декабря 9999 г.
Object (объект)	4 байт	Любой указатель объекта.
String (строка пе-	10 байт +	От 0 до приблизительно 2 миллиардов.
ременной длины)	длина строки	
String(строка по-	Длина стро-	От 1 до приблизительно 65 400.
стоянной длины)	КИ	
Variant(числовые	16 байт	Любое числовое значение вплоть до
подтипы)		границ диапазона для типа Double.
Variant(строковые	22 байт +	Как для строки (String) переменной
подтипы)	длина строки	длины.
Тип данных,	Объем опре-	Диапазон каждого элемента определя-
определяемый	деляется	ется его типом данных.
пользователем(с	элементами	
помощью ключе-		
вого слова Туре)		

Str = "Здравствуй мир!"

Этой строкой кода, мы присваиваем переменной Str значение "Здравствуй мир!". Обратите внимание, что в кавычки заключаются текстовые данные.

Следующая строка нашей программы

Этой строкой мы помещаем надпись "Здравствуй мир!" на метку, расположенную на нашей форме. Так выглядит типичная команда VBA. Читается она следующим образом. Лучше всего читать справа налево: свойству **Caption** объекта **lblMessage**, который находится в составе объекта **frmFirstProgram**, присваивается значение переменной **Str.** По-другому, свойству **надпись** объекта **метка**, находящемуся на **форме**, присваивается значение переменной, находящейся справа от знака равно. После названия объекта ставится точка, после которой указывается свойство объекта или метод обработки объекта, с которым мы работаем.

Набирая текст программы, вы должны были обратить внимание на то, что после того, как вы ставите точку, появляется меню, из которого можно выбрать мышкой или стрелками клавиатуры нужное свойство или метод (см. рис. 9).



Рис. 2.7. Рис. 9. Выбор свойства или метода

Это очень облегчает работу, поскольку не нужно набирать все буквы команды. Еще облегчает ввод команд служба завершения слова - можно не писать целиком слово frmFirstProgram, а набрать frm и нажать Ctrl + пробел. Редактор допишет слово до конца сам или предоставит возможность выбора слова, если его однозначно нельзя определить по первым буквам. Эта помощь позволяет сосредоточиться на решаемой программной задаче, а не на синтаксисе. И не нужно держать в памяти правила написания каждой команды. На экране, соответствующему рис. 9, также видно, что одни предлагаемые варианты имеют слева значок серого цвета, а другие зеленого. Серым цветом обозначаются свойства объекта. Зеленым метод действия над объектом. В нашем случае мы меняем свойство объекта.

Свойство - это характеристика объекта (цвет, размер, название и др.) Метод - это какое-либо действие, действие совершаемое самим объектом. Например, для метки есть метод Move, с помощью которого метка перемещается на форме, согласно указанным параметрам.

Когда мы проектировали форму, в окне свойств мы выставляли начальные значения свойств объекта. Но мы можем менять их программно, что и происходит в нашей программе. Начальное свойство **Caption** метки мы оставили пустым, а теперь мы его меняем программно, присваивая ему значение переменной Str. А это значение равно «Здравствуй мир!». При выполнении программы, эта надпись появляется на форме.

Следующая строка нашей программы:

MsgBox Str

MsgBox вызывает стандартное диалоговое окно с кнопкой ОК и помещает на это окно надпись, содержащуюся в переменной **Str**. Пока мы не нажмем кнопка ОК, программа не будет дальше работать

MsgBox может использоваться как оператор, или как функция. **MsgBox** выводит информацию на экран монитора (при использовании в качестве оператора), но имеет и много дополнительных возможностей, которые позволяют управлять ходом программы (при использовании в качестве функции).

Более подробную информацию по функции MsgBox можно получить в справочной системе VBA.

После нажатия ОК на стандартном диалоговом окне, будет выполнена следующая строчка нашей программы:

```
frmFirstProgram.Hide
```

Эта строка скрывает нашу форму. Она больше не нужна, т.к. выполнила свое предназначение.

Последние три строки предназначены для вывода информации в документ Word:

Documents.Add

Этой строкой мы создаем новый документ Word, в который помещаем нужную нам информацию. Метод Add добавляет новый документ в коллекцию открытых документов

```
Selection.TypeText Str
```

Эта строчка помещает текст из переменной Str в документ Word. Читается эта строка так: с помощью объекта Selection (выделение) напечатать методом TypeText данные, находящиеся в переменной Str. Должно быть понятно, что теперь идет работа не с объектами формы, а с объектами Microsoft Word. Selection - это один из основополагающих объектов Word.

Мы рассмотрели подробно нашу первую программу.

1. Назовите префиксы элементов управления: формы (Form), кнопки (Command Button), метки (Label), текстового поля (Text Box).

2. Что такое объект, свойство объекта, метод, событие, обработка события?

3. Назовите и охарактеризуйте типы данных.

4. Назовите математические функции языка VBA.

5. Дан текст программы:

```
Private Sub cmdRunMe_Click()
Dim Str As String
Str = "Здраствуй мир!"
lblMessage.Caption = Str
MsgBox Str
frmFirstProgram.Hide
```

```
Documents.Add
Selection.TypeText Str
frmFirstProgram.Show
End Sub
```

Опишите назначение каждой строки программы.

Лабораторная работа № 2-3

Решение задач с помощью математических функций VBA

Цель работы:

- 1. научиться создавать программы, используя математические функции языка программирования VBA
- 2. Изучить математические функции языка программирования VBA
- 3. Уметь преобразовывать функции, отсутствующие в языке программирования к стандартным функциям языка VBA

Прежде чем решать задачи рассмотрим основные математические функции языка VBA. В VBA имеется большой набор встроенных функций и процедур, использование которых существенно упрощает программирование.

В табл. 3 рассмотрены математические функции.

Таблица № 2.3 Таблица №3 Встроенные математические функции Visual Basic

Функция	Возвращаемое значение
Abs (X)	Модуль (абсолютная величина) числа. Х - числовая константа, переменная или арифметическое выражение
Atn(X)	Арктангенс. Аргументы тригонометрических функций задают- ся в радианах. Для перехода от градусов к радианам исполь- зуйте формулу: (α·π/180), где ос - величина угла в градусах; π = 3.1415926
Cos (X)	Косинус.
Exp (X)	Экспонента, т.е. результат возведения основания натурального логарифма в указанную степень
Log(X)	Натуральный логарифм ln(x)
Rnd(X)	Случайное число из интервала [0,1). Если Х меньше нуля, то
	Rnd возвращает каждый раз одно и то же число. Если X боль-
	ше нуля или опущено, то Rnd возвращает следующее случай-

	ное число в последовательности. Если Х равняется нулю, то
	Rnd возвращает случайное число, возвращенное при предыду-
	щем вызове этой функции. Перед вызовом функции Rnd ис-
	пользуйте оператор Randomize
Sgn (X)	Знак числа. Возвращает: -1 - если Х меньше 0; 1 - больше 0; 0 -
	равно 0
Sin (X)	Синус
Sqr(X)	Квадратный корень из числа
Tan (X)	Тангенс
Fix (X)	Обе функции Int и Fix отбрасывают дробную часть числа и воз-
Int (X)	вращают целое значение. Различие между функциями Int и Fix
	состоит в том, что для отрицательного значения параметра Х
	функция Int возвращает ближайшее отрицательное целое чис-
	ло, меньшее либо равное указанному, а Fix - ближайшее отри-
	цательное целое число, большее либо равное указанному

Для вычисления функций, не приведенных в табл. 3 можно использовать эквивалентные формулы. Например

$$\lg x = \frac{\ln(x)}{\ln(10)}$$
$$Log_a B = \frac{Ln(b)}{Ln(a)}$$
$$ctg(x) = \frac{1}{tg(x)}$$
$$\arccos(x) = \arctan\left(\frac{\sqrt{1-x^2}}{x}\right)$$
$$\arcsin(x) = \arctan\left(\frac{x}{\sqrt{1-x^2}}\right)$$
$$ctg^{-1}(x) = tg^{-1}\left(\frac{1}{x}\right)$$

Для обозначения арифметических операций используйте знаки: «+» – сложение,

«-» – вычитание,

«*» – (умножение),

«/» – (деление),

«\»-(целочисленное деление),

«^» – (возведение в степень).

При сложении строковых величин используйте символ & (амперсанд). Допускается использовать символ +, однако использование символа & более наглядно.

Порядок выполнения операций в арифметических выражениях определяется старшинством операций:

1) вычисление функций;

2) возведение в степень;

3) умножение и деление;

4) сложение и вычитание.

Желаемый порядок выполнения операций задается с помощью скобок. Выражения в скобках выполняются в первую очередь. При отсутствии скобок в арифметическом выражении операции, имеющие одинаковый приоритет, выполняются слева направо. Любые выражения на языке Visual Basic записываются в одну строку. Надстрочные и подстрочные записи не допускаются.

При записи арифметических выражений используются следующие правила:

• если в знаменателе произведение нескольких сомножителей, то его нужно заключить в скобки:

 $\frac{\sin x}{a \cdot \cos ax} \quad \rightarrow \quad Sin(X) \,/\, (A * Cos(A * X))$

• если числитель и знаменатель многочлены, то их нужно заключить в скобки:

$$\frac{a+x}{c-k} \ \rightarrow \ (A+X) \, / \, (C-K)$$

• два знака арифметических операций не должны следовать друг за другом:

$$\frac{a}{-b} \rightarrow A/(-B)$$

• извлечение корня любой степени, кроме квадратного, заменяется возведением подкоренного выражения в дробную степень, т.е.

$$\sqrt[3]{a+x} = (a+x)^{\frac{1}{3}} \rightarrow (A+X)^{(1/3)}$$

Для закрепления вышеизложенных правил рассмотрим следующие примеры:

$$\frac{\sqrt{\alpha + \beta} - \ln|\alpha - \beta|}{\sqrt[5]{\cos^2 \alpha}} \to (\operatorname{Sqr}(a+b)-\operatorname{Log}(\operatorname{Abs}(a-b)))/(\operatorname{Cos}(a)^2)^{(1/5)}$$

$$\frac{tg\chi - \sin 0.3\pi + e^{ax}}{\cos^3 2x^2 \cdot \sqrt{a + x}} \to (\operatorname{Tan}(x)-\operatorname{Sin}(0.3*3.1415926) + \operatorname{Exp}(a*x)/(\operatorname{Cos}(2*x^2)^3*\operatorname{Sqr}(a+x)))$$

$$\frac{3f_1 + 1.7 \cdot 10^{-3} \cdot e^{\sin a}}{\ln|\sin f_1|} - \frac{1}{2 - \sqrt[3]{a/f_1}} \to (3*f11.7\text{E}-3*\operatorname{Exp}(\operatorname{Sin}(a)))/\operatorname{Log}(\operatorname{Abs}(\operatorname{Sin}(f1))) - 1/(2 - (a/f1)^{(1/3)}))$$

$$\frac{\operatorname{arctg}(a + b^2) - \lg ab}{\pi \cdot tg^2(a + b)} \to (\operatorname{Atn}(a+b^2) - \operatorname{Log}(a*b)/\operatorname{Log}(10))/3.14*\operatorname{Tan}(a+b)^2)$$

$$\operatorname{cos} \frac{s}{t} - \arcsin st \to \operatorname{Cos}(s/t) - \operatorname{Atn}(s*t/\operatorname{Sqr}(1 - s^2 * t^2)))$$

$$\frac{4\sin^2 17^{\circ}}{1.8 \cdot 10^2} \to 4*\operatorname{Sin}(17*3.14/180)^2/1.8\text{E}2$$

Теперь попробуем решить задачу при помощи VBA.

Задача. Составить программу вычисления объема цилиндра высотой h и радиусом основания r. Расположить на форме все элементы ввода и вывода информации. Вывести также задание, результаты расчета в документ Word.

Формулой для расчета будет

$$\mathbf{v} = \pi \mathbf{r}^2 \mathbf{h}$$

Входными данными для расчета, которые заранее неизвестны, являются радиус основания цилиндра – r, и высота цилиндра – h. Выходными данными программы является объем – v.

Для решения поставленной задачи, выполним следующие действия.

1. Создадим форму с элементами для расчета (рис. 10)



Рис. 2.8. Рис. 10. Форма для расчета объема цилиндра

Таблица № 2.4 Таблица №

Таблица свойств:

Объект	Свойство	Значение
UserForm1	Name	frmObem
	Caption	Расчет объема цилин-
		дра
CommandButton1	Name	cmdRaschet
	Caption	Рассчитать
Label1	Caption	Расчет объема цилин-
		дра
Label2	Caption	Высота цилиндра h=
Label3	Caption	Радиус основания ци-
		линдра r=

Label4	Caption	Объем цилиндра равен V=
Label5	Name	lblOtvet
TextBox1	Name	txtVysota
TextBox2	Name	txtRadius

2. В текстовые поля мы будем вводить исходные данные: h и r. Нажав на кнопку cmdRaschet мы должны получить в метке lblOtvet результат расчета. Также условие задачи, исходные данные и ответ должны быть выведены в документ Word.

3. Создадим программу, которая обрабатывает событие нажатия кнопки.

```
Private Sub cmdRaschet Click()
    Dim h As Single
    Dim r As Single
    Dim V As Single
    Dim Otvet As String
   h = txtVysota.Text
    r = txtRadius.Text
    V = 3.14 * h * r * r
    lblOtvet.Caption = V
    Otvet = "При высоте h = " & h & " и радиусе r = " & r & "
Объем цилиндра V = " & V
    frmObem.wide
    Documents.Add
    Selection.TypeText Text:="Задача. Составить программу
вычисления объема цилиндра высотой h и радиусом основания r.
Расположить на форме все элементы ввода и вывода информации.
Вывести также задание, результаты расчета в документ Word."
    Selection.TypeParagraph
    Selection.TypeParagraph
    Selection.TypeText Text:=Otvet
    frmObem.Show
End Sub
```

Эта программа решает все поставленные задачи. Рассмотрим подробнее текст программы.

```
Dim h As Single
Dim r As Single
Dim V As Single
```

Операторами Dim объявляются переменные, которые участвуют при расчете объема цилиндра. Они имеют числовой тип Single. (См. тип данных).

```
Dim Otvet As String
```

Здесь объявляется переменная **Otvet**, которая будет использоваться для формирования отчета, отправляемого в документ Word.

```
h = CSng txtVysota.Text)
```

```
r = CSng (txtRadius.Text)
```

В этих строках переменным h и r присваиваются значения, которые берутся из текстовых полей txtVysota и txtRadius. За информацию, которая находится в текстовых полях, отвечает свойство Text. Поскольку свойство Text является строковым данным, а переменные h и r являются числовыми данными, то эти данные надо перевести из строковой формы в числовую. Для этого служит функция CSng. Она преобразуют строку, содержащую число в числовую форму типа Single. Для перевода в число типа Double используется функция CDbl. Существует еще одна функция для перевода строки в число – Val, но эта функция имеет языковое ограничение. Дело в том, что в английском языке для разделения целой части числа от дробной части используется точка – «2.5», а в русском языке разделителем является запятая – «2,5». Функция Val преобразует строку, содержащую только число с точкой в качестве разделителя, а число с запятой не воспринимает. В текстовые поля нужно вводить число с запятой в качестве разделителя. При этом в тексте программ все равно используется точка, как в строке ниже:

V = 3.14 * h * r * r

В этой строке рассчитывается объем цилиндра и присваивается переменной V.

```
lblOtvet.Caption = V
```

Этой строкой полученный ответ помещается в метку lblOtvet с помощью свойства Caption.

```
Otvet = "При высоте h = " & CStr(h) & " и радиусе r = " & CStr(r) & " Объем цилиндра V = " & CStr(V)
```

Данная строка формирует ответ для дальнейшего помещения в документ Word. Функция **CStr** обратна по своему назначению функциям **CSng**, **CDbl**, **Val**. Она преобразует числовые данные в строковые данные. Таким образом, выражения CStr(h) и CStr(r) являются строковыми данными. Операнд & может объединять данные различных типов (текстовых, числовых и других типов) в одну строку типа String. В нашем примере происходит объединение строковых (текстовых) данных.

Selection.TypeText Text:="Задача. Составить программу вычисления объема цилиндра высотой h и радиусом основания r. Расположить на форме все элементы ввода и вывода информации. Вывести также задание, результаты расчета в документ Word."

Данная строка печатает в документ Word текст, указанный в параметре Text:=. Здесь это условие задачи.

Selection.TypeParagraph Selection.TypeParagraph

Метод **ТуреРагадгарh** создаёт новый параграф (абзац) в объекте **Выделение**, перенося курсор на новую строку. Соответственно, в нашем примере две строки с этим методом позволяют разделить условие задачи и решение пустой строкой.

Selection.TypeText Text:=Otvet

Данная строка печатает в документ Word текст, указанный в параметре Text:=. Здесь это содержимое переменной Otvet.

Лабораторная работа №3

Цель работы:

Закрепить знания, полученные при изучении темы «Решение задач с помощью математических функций VBA»

Расположить на форме все элементы ввода и вывода информации. Вывести также задание, результаты расчета в документ Word.

Варианты

1. Даны два числа. Найти среднее арифметическое кубов этих чисел.

2. Определить катет прямоугольного треугольника по известным значениям гипотенузы и второго катета.

3. Определить площадь квадрата, вписанного в окружность радиусом **r**.

4. Вычислить площадь равностороннего треугольника, вписанного в окружность радиусом **r**.

5. Найти площадь круга, вписанного в равносторонний треугольник со стороной **b**.

6. Вычислить площадь равнобедренного треугольника со сторонами **a** и основанием **b**.

7. Найти площадь трапеции с высотой **h**, основанием **a** и противолежащей стороной **b**.

8. Вычислить площадь круга, вписанного в квадрат со стороной **а**.

9. Найти площадь прямоугольного параллелепипеда с ребрами **a**, **b**, **c**.

10. Вычислить площадь параллелограмма со сторонами а и b.

11. Вычислить площадь полого цилиндра с внешним радиусом **r1**, внутренним **r2** и высотой **h**.

12. Даны **x** и **y**. Получить $\frac{|x|-|y|}{1+|xy|}$.

13. Даны **x, y, z**. Вычислить **u** и **v**, если $u = \sin \left| (y - \sqrt{x})(x - \frac{y}{z^2 + x^3}) \right|, v = \cos(z^2 + \frac{x}{4}).$

и = $\frac{1 + \sin^2(\mathbf{x} + \mathbf{y})}{2 + \mathbf{x} - \frac{2\mathbf{x}}{1 + \sin(\mathbf{x} + \mathbf{y})}}$, $\mathbf{v} = \mathbf{x} - \frac{\mathbf{x}^2}{1 + \sin^2(\mathbf{x} + \mathbf{y})}$.

15. Определить координату середины отрезка (**a**, **b**).

16. Найти периметр квадрата, площадь которого равна s.

17. Найти площадь равностороннего треугольника, периметр которого равен **p**.

18. Вычислить среднее арифметическое трех чисел x, y, z.

19. Вычислить среднее геометрическое четырех положительных чисел **x**, **y**, **z**, **v**.

20. Треугольник задан координатами своих вершин: **x1,y1; x2,y2; x3,y3**. Найти периметр треугольника.

21. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.

22. Известны количество жителей в государстве и площадь его территории. Определить плотность населения в этом государстве.

23. Даны катеты прямоугольного треугольника. Найти его периметр.

24. Даны основания и высота равнобедренной трапеции. Найти ее периметр.

25. Даны длины сторон прямоугольного параллелепипеда. Найти его объем и площадь боковой поверхности.

26. Известны координаты двух точек на плоскости. Составить программу вычисления расстояния между ними.

27. Вычислить площадь равностороннего треугольника, вписанного в окружность радиусом **r**.

28. Заданы координаты трех вершин треугольника — (x_1, y_1) , (x_2, y_2) , (x_3, y_3) Найти его периметр и площадь.

29. Найти площадь треугольника, две стороны которого равны *a* и *b*, а угол между этими сторонами — γ .

30. Найти площадь круга, вписанного в треугольник с заданными сторонами.

31. Окружность вписана в квадрат заданной площади. Найти площадь квадрата, вписанного в эту окружность. Во сколько раз площадь вписанного квадрата меньше площади заданного?

32. Перевести градусы в радианы. Градусная мера задается в формате: deg (градусы) min (минуты) sec (секунды).

33. Написать программу округления числа с любой точностью. Число и точность округления вводятся с клавиатуры.

34. Смешано V_1 литров воды температуры t_1 с V_2 литрами воды температуры t_2 . Найти объем и температуру образовавшейся смеси.

35. Три сопротивления R1, R2, R3 соединены параллельно. Найти сопротивление соединения.

36. Определить координату середины отрезка (**a**, **b**).

37. Вычислить площадь параллелограмма со сторонами а и b.

38. Вычислить площадь круга, вписанного в квадрат со стороной **a**.

39. Найти площадь прямоугольного параллелепипеда с ребрами **a**, **b**, **c**.

40. Найти периметр квадрата, площадь которого равна s.

2.2. Контрольные вопросы

Лабораторная работа № 4-5

2.3. Создание нового документа

Мы уже рассматривали создание нового документа с помощью следующей строки:

Documents.Add

В этой строке **Documents** представляет собой коллекцию документов. После точки указывается метод **Add**, применяемый к данной коллекции и добавляющий к этой коллекции новый документ. Этот метод может иметь несколько параметров, которые указываются в скобках сразу после метода. Дополнительно о параметрах можно узнать в справочной системе VBA. Если этот метод используется без параметров, то новый документ создается с параметрами (размер страницы, поля, название и размер шрифта и т.д.) по умолчанию. Все величины и названия параметров по умолчанию находятся в автозагружаемом шаблоне **Normal.dot**. Эти параметры, в принципе, можно изменить.

Кроме метода Add коллекция Documents имеет другие методы, важнейшими из которых являются:

Open – открывает уже существующий документ;

Save – сохраняет открытый документ;

Close – закрывает открытый документ.

Также эта коллекция имеет свойства, важным из которых является свойство **Count**, которое возвращает количество открытых документов.

Подробнее об этих методах и свойствах можно узнать в справочной системе.

Но иногда при написании программы требуется обращаться к созданному документу в тексте программы как объекту. Например, если открыто много документов. Тогда можно воспользоваться следующим способом:

```
Sub New_Doc()

'Объявляем документ

Dim Doc1 As Document

'Создаем новый документ

Set Doc1 = New Document

'Делаем его активным

Doc1.Activate

End Sub
```

2.4. Ввод текста в отчет (документ) с помощью VBA.

Для ввода текста можно также использовать объект Selection, применяя к нему следующие методы:

InsertAfter – вставляет текст после выделения (не удаляя выделения);

InsertBefore – вставляет текст перед выделением (не удаляя выделения);

InsertParagraphAfter – вставляет новый параграф (абзац) после выделения (не удаляя выделения);

InsertParagraphBefore - вставляет новый параграф (абзац) перед выделением (не удаляя выделения);

InsertSymbol – вставляет символ, указанный в параметрах этого метода;

TypeText – вставляет текст. Если выделение не снято с ранее выделенного текста, то этот метод удаляет выделенный текст, и вместо него печатает свой;

ТуреРагадгарh – вставляет параграф. Если выделение не снято с текста, то этот метод данный текст удаляет, а вместо него вставляет параграф.

Пример:

```
Sub write()
Selection.TypeText Text:="Привет!"
Selection.TypeParagraph
Selection.InsertAfter "Пока"
End Sub
```

Может потребоваться снять выделение с текста. Это можно сделать несколькими способами:

Selection.Collapse Direction:=wdCollapseEnd – перемещает курсор в конец выделения и снимает выделение.

Selection.Collapse Direction:=wdCollapseStart – наоборот перемещает курсор в начало выделения и затем снимает.

Selection.EndOf Unit:= (константа) – перемещает курсор в конец объекта (символа, слова, предложения, параграфа, раздела и др.), указанного параметром константа.

Выравнивание абзаца осуществляется с помощью свойства .Alignment. В примере ему задается значение wdAlignParagraphJustify, т.е. выравнивание по ширине. Кроме этой константы можно использовать другие:

- wdAlignParagraphCenter (по центру);
- wdAlignParagraphLeft (по левому краю);
- wdAlignParagraphRight (по правому краю)

Интервал между строками задается свойством .LineSpacingRule. Это свойство может принимать несколько значений:

- wdLineSpace1pt5 устанавливает полуторный интервал;
- wdLineSpaceDouble устанавливает двойной интервал;

• wdLineSpaceSingle – устанавливает одинарный интервал.

Другие значения этого свойства (wdLineSpaceAtLeast, wdLineSpaceExactly, wdLineSpaceMultiple) используются совместно с другим свойством .LineSpacing, например:

```
With Selection.ParagraphFormat
  .LineSpacingRule = wdLineSpaceExactly
  .LineSpacing = 5
End With
```

Этот пример устанавливает межстрочный интервал в режим точно, измеряемый в пунктах. В этом примере устанавливается интервал между строками в 5 пунктов (.LineSpacing = 5).

```
With Selection.ParagraphFormat
   .LineSpacingRule = wdLineSpaceMultiple
   .LineSpacing = LinesToPoints(3)
End With
```

Этот пример устанавливает межстрочный интервал в режим **множитель**. В этом примере интервал задается в 3 раза больше одинарного (.LineSpacing = LinesToPoints(3)).

Абзац можно оформить так, чтобы он был окружен границами, которые могут выглядеть по-разному в зависимости от указанных параметров. Для того, чтобы получить границы вокруг абзаца используется объект **Borders (границы)**. Этот объект входит в состав объекта **ParagraphFormat**:

```
Selection.ParagraphFormat.Borders
```

Этот объект имеет множество свойств (тип границы, тип, ширина, цвет линии, внутренние поля границы и др.).

Чтобы использовать заливку в абзаце можно с помощью объекта .Shading(заливка), который также как и Borders входит в состав объекта ParagraphFormat:

```
Selection.ParagraphFormat.Shading
```

Этот объект имеет свойства цвет и узор.
Подробнее об использование на VBA объектов **Shading** и **Borders,** а также их свойств смотрите в справочной системе.

2.5. Вставка формул в отчет (документ) с помощью объекта Omath.

Практически не один инженерный или научный расчет не обходится без формул. Отчет является наглядным, если в нем присутствуют формулы. Стандартный вид формулы в расчетах выглядит так:

Величина = Формула с обозначениями = Формула с цифрами = Значение величины (единицы измерения).

Например:

$$I = \frac{U}{R} = \frac{4}{2} = 2$$

Такую формулу можно ввести с помощью VBA. Обычно формулы вводятся с помощью редактора формул. Однако редактор формул нельзя использовать на VBA. В этом случае мы можем использовать такой объект как **Omath**. Объект Omath представляет собой формулы в формате, появившемся в Microsoft Word 2007.

Рассмотрим некоторые основные действия на примерах. **41. Дробь**: «/».

Пример:

```
Sub formula1()
'Объявляем документ
Dim Doc As Document
'Создаем новый документ
Set Doc = New Document
```

```
'Делаем его активным
     Doc.Activate
     'Объявляем переменные
     Dim I As Single, U As Single, R As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
    R = 2: U = 4
     I = U / R
     'В переменную заносим код формулы
     formula = "I=U/R=" & U & "/" & R & "=" & I
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт оbjЕq, как формулу
     Dim objEq As OMath
     ' Устанавливаем объект objRange, как выделенную часть
     документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект оbjRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     ' создаем объект objEq, образуя его из объекта objRange
     SetobjEq = objRange.OMaths(1)
     'Преобразуем формулу из линейного вида к профессиональному
     objEq.BuildUp
End Sub
```

После выполнения данной программы будет создан новый документ со следующей формулой:

$$I = \frac{U}{R} = \frac{4}{2} = 2$$

Вид формулы можно преобразовать из профессионального в линейный и наоборот. В линейном виде формула выглядит как строка со всеми символами, формирующими данную формулу. Для преобразования нужно щелкнуть правой кнопкой мыши на формуле и выбрать нужный вид (Рис..).



Рис. 2.9. Изменение вида формулы.

После преобразования данная формула будет выглядеть следующим образом:

$$I = U/R = 4/2 = 2$$

Формирование формул в VBA происходит в линейном виде, после чего преобразуется в профессиональный.

Рассмотрим каждую строчку кода подробнее.

Dim Doc As Document

Оператором **Dim** мы объявляем объект **Doc**. **As Document** означает, что объект Doc объявляется документом.

```
Set Doc = New Document
```

Этой строкой мы создаем новый документ.

Doc.Activate

Делаем созданный документ активным.

Dim I As Single, U As Single, R As Single

Оператором **Dim** мы объявляем переменные. **As Single** означает, что переменные имеют дробный тип.

```
Dim formula As String
```

Оператором **Dim** мы объявляем переменную **formula**. **As String** означает, что переменная имеет строковой тип. В данной переменной будет содержаться код формулы.

R = 2: U = 4

Присваиваем значения переменным R и U. Знак «:» позволяет писать несколько операторов в одну строку.

formula = "I=U/R=" & U & "/" & R & "=" & I

В переменную заносим код формулы. Символ «/» используется в коде формулы для организации деления. Кавычки используются для вставки в код формулы текстовых данных: букв, символов. Чтобы внедрить в код формулы числовые значения, взятые из переменных, используется знак &(амперсанд).

Внимание! Следите, чтобы в коде формулы не было лишних пробелов!

```
Dim objRange As Range
```

Оператором **Dim** мы объявляем объект **objRange**, как объект Range, который представляет собой часть документа. Каждый объект **Range** определяется позицией начального и конечного символов (диапазоном).

```
Dim objEq As OMath
```

Оператором **Dim** мы объявляем объект **objEq**, который является формулой. Set objRange = Selection.Range

Устанавливаем объект **objRange**, как выделенную часть документа. В нашем случае место под курсором.

```
objRange.Text = formula
```

Свойству текст, объекта **objRange** присваивается значение переменной **formula**.

```
Set objRange = Selection.OMaths.Add(objRange)
```

Преобразуем объект objRange в математическую формулу.

```
Set objEq = objRange.OMaths(1)
```

Создаем объект objEq, образуя его из объекта objRange.

objEq.BuildUp

Преобразуем формулу из линейного вида в профессиональный.

42. Верхний индекс: «^».

Пример:

```
Sub formula2()

'Объявляем документ

Dim Doc As Document

'Создаем новый документ

Set Doc = New Document

'Делаем его активным

Doc.Activate

'Объявляем переменные

Dim I As Single, P As Single, R As Single

'Объявляем переменную, в которой будет содержаться код

формулы

Dim formula As String

'Присваеваем значение переменным

R = 2: I = 4
```

```
P = I^{2} R
'В переменную заносим код формулы
formula = "P=I^2·R=" & I ^ 2 & "." & R & "=" & P
'Объявляем объект - диапазон objRange
Dim objRange As Range
'Объявляем объкт орјЕд, как формулу
Dim objEq As OMath
' Устанавливаем объект objRange, как выделенную часть
документа. В нашем случае место под курсором
Set objRange = Selection.Range
'свойству text объекта objRange присваиваем код формулы
objRange.Text = formula
'преобразуем объект оbjRange в математическую формулу
Set objRange = Selection.OMaths.Add(objRange)
'создаем объект оbjEq, образуя его из объекта objRange
Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному
objEq.BuildUp
```

End Sub

Для того чтобы поставить знак умножения нужно нажать следующую комбинацию [ALT + 0183] или воспользоваться таблицей символов. Таблицу символов можно вызвать через меню Пуск, выполнив следующие действия: Открыть меню Пуск – Стандартные– Служебные – Таблица символов.

После выполнения данной программы будет создан новый документ со следующей формулой:

$$P = I^2 \cdot R = 16 \cdot 2 = 32$$

43. Нижний индекс:

Пример:

Sub formula3() 'Объявляем документ Dim Doc As Document 'Создаем новый документ Set Doc = New Document

```
Doc.Activate
     'Объявляем переменные
     Dim I As Single, I1 As Single, I2 As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
     I1 = 2: I2 = 4
     I = I1 + I2
     'В переменную заносим код формулы
     formula = "I=I 1 + I 2=" & I1 & "+" & I2 & "=" & I
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт оbjEq, как формулу
     Dim objEq As OMath
     'Устанавливаем объект objRange, как выделенную часть
     документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект оbjRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     'создаем объект objEq, образуя его из объекта objRange
     Set objEq = objRange.OMaths(1)
     'Преобразуем формулу из линейного вида к профессиональному
     objEq.BuildUp
End Sub
```

После выполнения данной программы будет создан новый документ со следующей формулой:

 $I = I_1 + I_2 = 2 + 4 = 6$

44. Скобки: «(..)».

'Делаем его активным

Пример:

```
Sub formula4()
'Объявляем документ
```

```
Dim Doc As Document
     'Создаем новый документ
     Set Doc = New Document
     'Делаемегоактивным
     Doc.Activate
     'Объявляем переменные
     Dim a As Single, b As Single, x As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
     a = 2: b = 4
     x = (a / b) ^ 2
     'В переменную заносим код формулы
     formula = "x=(a/b)^2=(" & a & "/" & b & ")^2=" & x
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт орјЕд, как формулу
     Dim objEq As OMath
     'Устанавливаем объект objRange, как выделенную часть
     документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект оbjRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     'создаем объект оbjEq, образуя его из объекта objRange
     Set objEq = objRange.OMaths(1)
     'Преобразуем формулу из линейного вида к профессиональному
     objEq.BuildUp
End Sub
```

После выполнения данной программы будет создан новый документ со следующей формулой:

$$x = \left(\frac{a}{b}\right)^2 = \left(\frac{2}{4}\right)^2 = 0,25$$

45. Тригонометрические функции: «sin, cos, tg, ctgu т.д.».

Пример:

```
Sub formula5()
     'Объявляем документ
     Dim Doc As Document
     'Создаем новый документ
     Set Doc = New Document
     'Делаем его активным
     Doc.Activate
     'Объявляем переменные
     Dim a As Single, b As Single, x As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
     a = 0: b = 0
     x = sin(a) * cos(b)
     'В переменную заносим код формулы
     formula = "x=sin a \cdot cos b=" & sin(a) & "\cdot" & cos(b) & "=" &
    Х
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт орјЕд, как формулу
     Dim objEq As OMath
     'Устанавливаем объект objRange, как выделенную часть
     документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект objRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     'создаем объект objEq, образуя его из объекта objRange
     Set objEq = objRange.OMaths(1)
     'Преобразуем формулу из линейного вида к профессиональному
     objEq.BuildUp
End Sub
```

После выполнения данной программы будет создан новый документ со следующей формулой:

```
x = \sin a \cdot \cos b = 0 \cdot 1 = 0
```

46. Модуль:

Пример:

```
Sub formula6()
     'Объявляем документ
     Dim Doc As Document
     'Создаем новый документ
     Set Doc = New Document
     'Делаем его активным
     Doc.Activate
     'Объявляем переменные
     Dim a As Single, b As Single, x As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
     a = -3: b = 4
     x = Abs(a * b)
     'В переменную заносим код формулы
     formula = "x=|a·b|=" & " | " & a & "·" & b & " | " & "=" & x
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт орјЕд, как формулу
     Dim objEq As OMath
     'Устанавливаем объект objRange, как выделенную часть
    документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект objRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     'создаем объект objEq, образуя его из объекта objRange
     Set objEq = objRange.OMaths(1)
     'Преобразуем формулу из линейного вида к профессиональному
    objEq.BuildUp
```

После выполнения данной программы будет создан новый документ со следующей формулой:

```
x = |a \cdot b| = |-3 \cdot 4| = 12
```

Внедрение специальных символов с помощью функции ChrW 47. Корень квадратный

Пример:

```
Sub formula7()
     'Объявляем документ
     Dim Doc As Document
     'Создаем новый документ
     Set Doc = New Document
     'Делаем его активным
     Doc.Activate
     'Объявляем переменные
     Dim a As Single, b As Single, x As Single
     Объявляем переменную, в которой будет содержаться код
     формулы
     Dim formula As String
     'Присваеваем значение переменным
     a = 3: b = 3
     x = Sqr(a * b)
     'В переменную заносим код формулы
     formula = "x=" & ChrW(8730) & "(a·b) = " & ChrW(8730) & "(" &
     a & "·" & b & ")=" & x & "·" & ChrW(8730) & b & "=" & x
     'Объявляем объект - диапазон objRange
     Dim objRange As Range
     'Объявляем объкт objEq, как формулу
     Dim objEq As OMath
     ' Устанавливаем объект objRange, как выделенную часть
     документа. В нашем случае место под курсором
     Set objRange = Selection.Range
     'свойству text объекта objRange присваиваем код формулы
     objRange.Text = formula
     'преобразуем объект objRange в математическую формулу
     Set objRange = Selection.OMaths.Add(objRange)
     'создаем объект objEq, образуя его из объекта objRange
```

```
Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному
objEq.BuildUp
```

End Sub

После выполнения данной программы будет создан новый документ со следующей формулой:

$$x = \sqrt{a \cdot b} = \sqrt{3 \cdot 3} = 3$$

Для отображения знака корня используется функция ChrW(8730), которая возвращает символ корня ($\sqrt{}$) по его коду (8730).

Коды некоторых специальных символов приведены в следующей таблице.

Таблица № 2.5 Таблица №

Код	Сим-	Пример. Проф. вид	Пример. Лин.вид
	вол		
8712	E		
960	π		
8719	Π		
8721	Σ		
8723	Ŧ		
8730			$x = \sqrt{(a \cdot b)} = \sqrt{(3 \cdot 3)} = 3$
		$x = \sqrt{a \cdot b} = \sqrt{3 \cdot 3}$	
		= 3	
8731	∛		
8732	$\sqrt[4]{}$		
8734	∞		
8747	ſ		$\int m 1/x dx$
		$\int \frac{1}{x} dx$	
8804	\leq		
8805	\geq		
8869	Ţ		

Коды символов

9618		_	$\int m 1/x dx$
		$\int \frac{1}{x} dx$	
9632			\blacksquare (<i>a</i> _11& <i>a</i> _12@ <i>a</i> _21& <i>a</i> _22)
		$a_{11} a_{12}$	
		a_{21} a_{22}	
9508	-	$\int F_1(x_1, x_2, \dots x_n = 0)$	$\{\blacksquare(F_1(x_1, x_2, \dots x_n)$
		$(F_2(x_1, x_2, \dots x_n = 0))$	$= 0)@F_2(x_1, x_2, \dots x_n)$
			= 0))+
9601	_	<u>abc</u>	_abc
175	—		⁻ ABC
		\overline{ABC}	
9645		$a^2 = b^2 + c^2$	$\Box(a^{2} = b^{2} + c^{2})$

Символы греческого алфавита

Код	Символ	Код	Символ
	α	946	β
945			
	γ	948	δ
947			
	3	955	λ
949			
	μ	957	v
956			
	π	961	ρ
960			
	Σ	963	σ
931			
	τ	966	φ
964			

Для того чтобы найти коды остальных символов можно воспользоваться программой или с помощью калькулятора.

Следующая программа поместит в документ все символы в нашем случае от 8704 до 9632. В этом интервале находится большинство нужных математических символов, наиболее часто используемых. Данные числа можно задавать произвольно.

```
Subasdfe()
Dim x As Long
For x = 8704 To 9632
With Active Document.Range
.InsertAfter "(" & x & " - " & ChrW(x) & ")"
.InsertParagraphAfter
End With
Next
End Sub
```

Для того чтобы узнать другие греческие символы которые не вошли в таблицу и в диапазон выше, измените параметры строки (For x = 8704 To 9632) на (For x = 913 To 969).

Чтобы вычислить код символа при помощи калькулятора нужно выполнить следующие действия:

- 1. Найдите необходимый символ в Таблице символов. Как открыть эту таблицу смотрите выше.
- Запишите шестнадцатеричный код символа, который отображен в нижнем левом углу. Например, для знака корня код: 221А.
- 3. Открыть стандартный калькулятор и изменить вид на **Программист** (для Windows7 и Windows8) или на **Инженерный** (для WindowsXP).
- 4. Выбрать шестнадцатеричную систему исчисления (Нех).
- 5. Набрать найденный код нужного символа.
- 6. Выбрать десятеричную систему исчисления (Dex).

В результате получим десятеричный код символа для функции ChrW.

48. Интеграл.

Иногда нужно, чтобы символ был в высоту всего выражения. Тогда используется специальный символ . Его код можно также найти в таблице. Он отделяет данный символ от внутреннего выражения.

$$\int \frac{1}{x} dx$$

В линейном виде данная формула будет выглядеть так:

$$\int m 1/x \, dx$$

Если удалить символ , то формула при переводе в профессиональный вид будет выглядеть так:

$$\frac{\int 1}{x} dx$$

Пример программы:

```
Sub formula10()
```

'Объявляем документ Dim Doc As Document 'Создаем новый документ Set Doc = New Document 'Делаем его активным Doc.Activate 'Объявляем переменные Dim x As Single 'Объявляем переменную, в которой будет содержаться код формулы Dim formula As String

```
'В переменную заносим код формулы
formula = ChrW(8747) & ChrW(9618) & "1/x dx"
'Объявляем объект – диапазон objRange
Dim objRange As Range
'Объявляем объкт objEq, как формулу
Dim objEq As OMath
```

' Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором

```
Set objRange = Selection.Range
'свойству text объекта objRange присваиваем код формулы
objRange.Text = formula
'преобразуем объект objRange в математическую формулу
Set objRange = Selection.OMaths.Add(objRange)
'создаем объект objEq, образуя его из объекта objRange
Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному
objEq.BuildUp
```

End Sub

Кроме интеграла данный символ так же использует при составлении формул со знаками суммы, первообразной, скобок.

49. Матрица.

Чтобы создать матрицу также понадобится специальный символ: ■. Его код представлен в таблице.

```
a_{11} a_{12}
a_{21} a_{22}
```

В линейном виде эта матрица выглядит следующим образом:

\blacksquare (*a*_11&*a*_12@*a*_21&*a*_22)

Амперсанд (символ &) нужен для разделения данных одного столбца от другого. Символ (а) разделяет строки.

Пример программы:

```
Sub formula10()
     'Объявляем документ
     Dim Doc As Document
     'Создаем новый документ
     Set Doc = New Document
     'Делаем его активным
     Doc.Activate
     Dim formula As String
     'В переменную заносим код формулы
```

```
formula = ChrW(9632) & "(a 11&a 12@a 21&a 22)"
'Объявляем объект – диапазон objRange
Dim objRange As Range
'Объявляем объект орјЕд, как формулу
```

Dim objEq As OMath

Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором Set objRange = Selection.Range
'свойству text объекта objRange присваиваем код формулы objRange.Text = formula
'преобразуем объект objRange в математическую формулу Set objRange = Selection.OMaths.Add(objRange)
'создаем объект objEq, образуя его из объекта objRange Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному objEq.BuildUp

50. Скобка для системы уравнений.

Для системы уравнений используются символы {, - . Обратите внимание, что символ - ставится в конце кода формулы.

Профессиональный вид формулы:

$$\begin{cases} F_1(x_1, x_2, \dots x_n = 0) \\ F_2(x_1, x_2, \dots x_n = 0) \end{cases}$$

Линейный вид:

 $\{\blacksquare (F_1 (x_1, x_2, \dots x_n = 0) @F_2 (x_1, x_2, \dots x_n = 0)) \mid$

Символ • используется тот же, что и при создании матрицы. Пример программы:

```
Subformula11()
```

```
'Объявляем документ
Dim Doc As Document
'Создаем новый документ
Set Doc = New Document
'Делаем его активным
Doc.Activate
Dim formula As String
'В переменную заносим код формулы
formula = "{" & ChrW(9632) & "(F_1 (x_1,x_2,...x_n=0)@F_2
(x 1,x 2,...x n=0) )" & ChrW(9508)
```

```
'Объявляем объект – диапазон objRange
Dim objRange As Range
'Объявляем объкт objEq, как формулу
Dim objEq As OMath
```

Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором
Set objRange = Selection.Range
'свойству text объекта objRange присваиваем код формулы objRange.Text = formula
'преобразуем объект objRange в математическую формулу
Set objRange = Selection.OMaths.Add(objRange)
'создаем объект objEq, образуя его из объекта objRange
Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному objEq.BuildUp

End Sub

51. Формула в рамке.

Для создания формулы в рамке используется символ □. Профессиональный вид формулы:

$$a^2 = b^2 + c^2$$

Линейный вид:

$$\Box(a^2 = b^2 + c^2)$$

Пример программы:

```
Sub formula12()

'Объявляем документ

Dim Doc As Document

'Создаем новый документ

Set Doc = New Document

'Делаем его активным

Doc.Activate

Dim formula As String

'В переменную заносим код формулы

formula = ChrW(9645) & "(a^2=b^2+c^2)"
```

'Объявляем объект – диапазон objRange Dim objRange As Range 'Объявляем объкт objEq, как формулу Dim objEq As OMath

'Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором Set objRange = Selection.Range 'свойству text объекта objRange присваиваем код формулы objRange.Text = formula 'преобразуем объект objRange в математическую формулу Set objRange = Selection.OMaths.Add(objRange) 'создаем объект objEq, образуя его из объекта objRange Set objEq = objRange.OMaths(1) 'Преобразуем формулу из линейного вида к профессиональному objEq.BuildUp

End Sub

52. Черта сверху.

Для того чтобы над числом, буквой или выражением появилась черта используется символ –.

Например:

ABC

В линейном виде это выражение будет выглядеть следующим образом:

⁻ABC

Пример:

Sub formula13()

'Объявляем документ Dim Doc As Document 'Создаем новый документ Set Doc = New Document 'Делаем его активным Doc.Activate Dim formula As String

'В переменную заносим код формулы formula = ChrW(175) & "ABC" 'Объявляем объект – диапазон objRange Dim objRange As Range 'Объявляем объкт objEq, как формулу Dim objEq As OMath

'Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором Set objRange = Selection.Range 'свойству text объекта objRange присваиваем код формулы objRange.Text = formula 'преобразуем объект objRange в математическую формулу Set objRange = Selection.OMaths.Add(objRange) 'создаем объект objEq, образуя его из объекта objRange Set objEq = objRange.OMaths(1) 'Преобразуем формулу из линейного вида к профессиональному objEq.BuildUp

End Sub

53. Черта снизу.

Для того чтобы над числом, буквой или выражением появилась черта используется символ _.

Например:

<u>abc</u>

В линейном виде это выражение будет выглядеть следующим образом:

```
_abc
```

Пример:

Sub formula14()

'Объявляем документ Dim Doc As Document 'Создаем новый документ Set Doc = New Document 'Делаем его активным Doc.Activate Dim formula As String

'В переменную заносим код формулы formula = ChrW(9601) & "abc" 'Объявляем объект – диапазон objRange Dim objRange As Range 'Объявляем объкт objEq, как формулу Dim objEq As OMath

'Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором Set objRange = Selection.Range 'свойству text объекта objRange присваиваем код формулы objRange.Text = formula 'преобразуем объект objRange в математическую формулу Set objRange = Selection.OMaths.Add(objRange) 'создаем объект objEq, образуя его из объекта objRange Set objEq = objRange.OMaths(1) 'Преобразуем формулу из линейного вида к профессиональному objEq.BuildUp

End Sub

Лабораторная работа №4

Задача. Составить программу, которая производит расчет по формуле:

$$\frac{b+\sqrt{b^2+4\cdot a\cdot c}}{2\cdot a}-a^3\cdot c+b^{-2}$$

Программа должна содержать форму, которая должна иметь текстовые поля для ввода величин, кнопки для расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA. Отчет должен содержать:

условие задачи;

формулу расчета с обозначениями и подставленными вместо них числами;

полученный результат.

Решение.

1. Создайте форму программы (см. рис.).

Расчет формулы				×
Расчет по формуле	9:	frm	Formula	
$b + \sqrt{b^2 + 4}$	$\frac{1}{1} \cdot a \cdot c}{-a^{3} \cdot c}$	$+h^{-2} =$		
2∙a	-a c	10	IblOtve	f.
a= b=	C=			
Рассчитать	Создать отчет	Выход		
cmdRaschet	cmdOtchet cr	ndExit		

Рис. 2.10. Рис.. Форма для расчета формулы.

На рисунке указаны названия для свойства Name элементов.

В текстовые поля будут вводиться данные для расчета. При нажатии Кнопки cmdRaschet (Рассчитать) должен быть произведен расчет по формуле и выведен в элемент Надпись lblOtvet. При нажатии кнопки cmdOtchet (Создать отчет), должен формироваться отчет. Нажатием на кнопку cmdExit программа заканчивает свое выполнение.

2. Объявляем все переменные вне всех процедур, так как в этом случае они видны во всех процедурах, а не только в той, в которой их объявили.

```
Dim a As Single, b As Single, c As Single, result As Single
```

3. Создадим код программы для обработки события нажатия кнопки cmdRaschet (Рассчитать):

```
Private Sub cmdRaschet_Click()

    a = CSng(txtA.Text)

    b = CSng(txtB.Text)

    c = CSng(txtC.Text)

    result = (b + Sqr(b ^ 2 + 4 * a ^ 3 * c)) / (2 * a) - a ^ 3 * c + b ^ (-2)

    lblOtvet.Caption = CStr(result)
```

В этом коде функция **CStr** преобразует числовое значение переменной **result** в текстовое значение для вывода в надпись **lblOtvet**.

4. Создадим код программы для обработки события нажатия кнопки cmdOtchet (Создать отчет):

Private Sub cmdOtchet_Click()

```
frmFormula.Hide
Dim docNew As Document 'объявляем новый документ
создаем новый документ и устанавливаем на него ссылку
Set docNew = Documents.Add
With Selection
  'Печатаем заголовок
  .Insertlfter "Программа для расчета формулы."
  'напечатанный заголовок форматируем по середине
  .ParagraphFormat.Alignment = wdAlignParagraphCenter
  создаем отступ для первой строки
  .ParagraphFormat.FirstLineIndent = CentimetersToPoints(0.7)
  устанавливаем для заголовка полужирный шрифт, 14 размера
  .Font.Bold = True
  .Font.Size = 14
  'вставляем два параграфа
  .InsertParagraphAfter
  .InsertParagraphAfter
  'Перемещаем курсор в конец документа
  .EndOf Unit:=wdSection
  Печатаем задание
  .InsertAfter "Задание. "
  .ParagraphFormat.Alignment = wdAlignParagraphJustify
  .EndOf Unit:=wdSection
  .Insertlfter "Составить программу для расчета формулы. Программа должна содержать форму, которая должна иметь" _
  « "текстовые поля для ввода величин, кнопки для расчета, формирования отчета, выхода из программы.
  .InsertParagraphAfter
  .Insertlfter "Сформировать отчет средствами VBl. Отчет должен содержать:"
  .InsertParagraphAfter
  .InsertAfter "условие задачи:"
  .InsertParagraphAfter
  .Insertlfter "формулу расчета с обозначениями и подставленными вместо них числами;"
  .InsertParagraphAfter
  .InsertAfter "полученный результат."
.Font.Bold = False
  .InsertParagraphAfter
  .EndOf Unit:=wdSection
  'Вывод результатов в отчет
  .Font.Bold = True
  .InsertAfter "Решение. "
  .InsertParagraphAfter
  .EndOf Unit:=wdSection
  .Font.Bold = False
  .InsertAfter "Для а = " & CStr(a) & ", для b = " & CStr(b) & ", для с = " & CStr(c) & " значение формулы:"
  .InsertParagraphAfter
  .InsertParagraphAfter
  .EndOf Unit:=wdSection
  Formula = "(b+" & ChrW(8730) & "(b^2+4.a.c))/(2.a)-a^3.c+b^-2=(" & b & "+" & ChrW(8730)
   ε "(" ε b ε "^2+4." ε a ε "." ε c ε "))/(2." ε a ε ")-" ε a ε "^3." ε c ε "+" ε b ε "^-2=" ε result
  'Объявляемобъект - диапазоноbjRange
Dim objRange As Range
'ОбъявляемобъкторјЕq, какформулу
Dim objEq As OMath
' УстанавливаемобъектоbjRange, как выделенную часть документа. В нашем случае место под курсором
Set objRange = Selection.Range
'свойствуtextoбъектаоbiRange присваиваем код формулы
objRange.Text = Formula
'преобразуемобъектоbjRange в математическую формулу
Set objRange = Selection.OMaths.Add(objRange)
' создаем объект objEq, образуя его из объекта objRange
Set objEq = objRange.OMaths(1)
'Преобразуем формулу из линейного вида к профессиональному
objEq.BuildUp
End With
```

frmFormula.Show

End Sub

Для нужной нам формулы код будет выглядеть следующим образом:

5. Создадим код программы для обработки события нажатия кнопки cmdExit (Выход):

```
Private Sub cmdExit_Click()
End
End Sub
```

Задание выполнено.

Обратите внимание на то, как переносятся длинные строки программы VBA:

.Insertåfter "Составить программу для расчета формулы. Программа должна содержать форму, которая должна иметь" _ & "текстовые поля для ввода величин, кнопки для расчета, формирования отчета, выхода из программы."

В месте где нужно разделить длинную строку ставится пробел и знак нижнего подчеркивания «_», далее на новой строке ставится знак «&», пробел, после чего продолжается печататься строка.

В тексте программы встречается функция CentimetersToPoints. Эта функция преобразует сантиметры в пункты. Программно все параметры страницы задаются в пунктах. CentimetersToPoints (1.2) преобразует 1,2 см в пункты. 1 см = 28.35 пунктам.

Существуют также и другие функции преобразования:

- MillimetersToPoints преобразует миллиметры в пункты 1 мм = 2,835 пунктам;
- InchesToPoints преобразует дюймы в пункты 1 дюйм = 72 пунктам;
- · PointsToCentimeters преобразует пункты в сантиметры;
- · PointsToMillimeters преобразует пункты в миллиметры;
- · PointsToInches преобразует пункты в дюймы
- и другие.

Лабораторная работа №5

Закрепить знания, полученные при изучении темы «Вставка формул в документ с помощью объекта Omath»

Варианты заданий

Составить программу, которая производит расчет по приведенной формуле.

Программа должна содержать форму с текстовыми полями для ввода величин, кнопками для расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA. Отчет должен содержать:

- условие задачи;

- формулу расчета с обозначениями и подставленными вместо них числами;

- полученный результат.

Варианты заданий

1.	$\frac{a}{c} \cdot \frac{b}{d} - \frac{ab-c}{cd}$	2. $\frac{\sin x + \cos y}{\cos x + \sin y} tg xy$
3.	$\frac{x+y}{y+1} - \frac{xy-12}{34+x}$	4. $\frac{3 + e^{y-1}}{1 + x^2 y - tg x }$
5.	$x - \frac{x^3}{3} + \frac{x^5}{5}$	6. $\ln\left(y - \sqrt{ x }\right) \left(x - \frac{y}{x + \frac{x^2}{4}}\right)$

$$25. \qquad x - 10^{\sin x} + \cos(x - y) \qquad 26. \qquad \cos^2\left(\sin\frac{1}{z}\right)$$

$$27. \qquad \frac{\cos^2 x}{\sin x} - xyz + \frac{ax^2 + bx + c}{dx^3 - f}$$

Лабораторная работа № 6

Управление ходом программы

Цель работы:

- 1. Изучить конструкции ветвления If...Then...ElseIf...EndIf
- 2. Изучить понятия: логические отношения, логические операции, логические операторы
- 3. Изучить виды логические отношений, логических операций, логических операторов
- 4. Изучить функцию InputBox

Конструкция ветвления If...Then...ElseIf...EndIf. Логические операции

Что же такое управление ходом программы? Чтобы это понять рассмотрим функцию:

$$F(x) = \begin{cases} x, & \text{если } x \le -1 \\ x^2, & \text{если } -1 < x < 1 \\ x, & \text{если } x \ge 1. \end{cases}$$

Как видно из условия задачи, функция высчитывается по одному из предложенных вариантов. В программе, с помощью которой будет решаться эта задача, необходимо предусмотреть выбор варианта для расчета функции в зависимости от значения *x*. Это и будет управление ходом программы.

Программа должна работать по следующему алгоритму:

Ввод числа x Если $x \le -1$ или $x \ge 1$, то вычислить по формуле F(x) = x Иначе если x > -1 и x < 1, то вычислить по формуле F(x) = x^2 Вывод результата F(x) Для решения подобной задачи в языке Basic есть специальная конструкция:

```
If условие-1 Then
Код программы для условия-1
ElseIf условие-2 Then
Код программы для условия-2
ElseIf условие-3 Then
Код программы для условия-3
ElseIf условие-n Then
Код программы для условия-n
Else
Код программы для всех оставшихся (нерассмотренных
выше)условиях
End If
```

Оператор If должен быть первым оператором в строке. Блок If должен заканчиваться оператором End If.

Оба оператора Else и ElseIf являются необязательными. В блоке If допускается любое число операторов ElseIf, но ни одно такое предложение не может появляться после предложения Else.

Условие представляет собой логическое выражение, имеющее значение типа *Boolean* - *True* (Истина) или *False* (Ложь)

Работа блока If...End If происходит следующим образом:

Проверяются последовательно условия после операторов If, ElseIf,

Если *условие-1* имеет значение *True* (Истина), ты выполняется код программы, следующий за оператором **Then** (*Код программы для условия-1*). Если же *условие-1* имеет значение *False* (Ложь), то по очереди осуществляется проверка *условия-2* после оператора **Elself**.

Если *условие-2* имеет значение *True* (Истина), ты выполняется код программы, следующий за оператором **Then** (Код программы для

условия-2). Если же *условие-2* имеет значение *False* (Ложь), то по очереди осуществляется проверка *условия-3* после оператора ElseIf.

И так далее, пока не будет обнаружено *условие* значение которого будет равно *True* (Истина)

При обнаружении условия со значением *True* выполняются операторы, непосредственно следующие за соответствующим предложением **Then**.

Если ни одно из *условий* не имеет значение *True* (или предложения **ElseIf** отсутствуют), выполняются операторы, следующие за **Else**. После выполнения операторов, следующих за **Then** или **Else**, выполнение продолжается со строки программы, следующей за **End If**.

Если в блоке будет несколько условий со значением True (истина), то будет выполняться код программы, после первого условия, значение которого равно True.

Если все условия будут равны False и будет отсутствовать оператор Else, то ни одна ветка условий не сработает и не будет выполнено ни одной строки программы внутри блока If...End If.

Допускаются вложенные блоки операторов **If**, например, ниже выделен курсивом вложенный блок операторов If.

```
If условие Then

If условие1 Then

Код программы для условия-1

ElseIf условие2 Then

Код программы для условия-2

Else

Код программы для всех оставшихся (нерассмотренных выше)условий

внутренней конструкции.

End If

ElseIf условие4 Then

Код программы для условия-4

Else

Код программы для всех оставшихся (нерассмотренных выше)условиях

внешней конструкции

End If
```

Как уже говорилось выше Условие представляет собой логическое выражение, имеющее значение типа *Boolean - True* (Истина) или *False* (Ложь). Логическое выражение может состоять из логических отношений и логических операций.

Логические отношения - это сравнение двух выражений. Результат сравнения тоже имеет тип *Boolean - True* (Истина) или *False* (Ложь). Для сравнения двух выражений используются операторы сравнения табл. 1:

Таблица 1

Оператор	Пример
<(Меньше)	If a < b Then
<= (Меньше или равняется)	If a <= b Then
>(Больше)	If $a > b$ Then
>= (Больше или равняется)	If $a \ge b$ Then
= (Равняется)	If $a = b$ Then
<>(Не равняется)	If $a <> b$ Then

Обратите внимание! Что при сравнении а с b, оператор равно (=) не является присваиванием, а является оператором сравнения!

Логические операции – это операции над логическими отношениями. Логические операции осуществляются с помощью логических операторов: Not (Her) (логическое отрицание), And (И) (логическое умножение), Eqv (эквивалентность), Or (Или) (логическое сложение) и Xor (исключающее Или). Логические операции используют, когда в одном условии нужно записать несколько простых условий

Рассмотрим подробнее каждый из них.

Логический оператор And используется для логического объединения двух логических отношений. Этот оператор проверяет, оба ли отношения одновременно истинны. Если оба отношения истинны (True), то и результат будет истинным (True); если же хотя бы одно из отношения ложно (False), то и результат тоже будет ложным.

	'	Во	звра	щa	ет	•	(и)			
8	>	10	And	8	>	6	'	Возвращает	False.	(и True)
8	>	10	And	6	>	8	'	Возвращает	False.	(False и False)
1() >	> 8	And	6	>	8	'	Возвращает	False.	(True и False)

a = 10, b = 8 c = 6			
Оператор	Первое усло-	Второе усло-	Результат логиче-
	вие	вие	ской операции
If $a > b$ And $b > c$ Then	True	True	True
If a < b And b \geq c Then	False	True	False
If a = b And a < c Then	False	False	False
If a $\langle \rangle$ b And b \rangle c Then	True	False	False

Логический оператор Eqv используется для проверки эквивалентности (одинаковости) двух логических отношений. Если их логические значения совпадают (оба равны *True* или оба равны *False*), то результат будет истинным; если логические значения различаются, то результат будет ложным.

a = 10, b = 8 c = 6			
Оператор	Первое усло-	Второе усло-	Результат логиче-
	вие	вие	ской операции
If a > b Eqv b > c Then	True	True	True
If a < b Eqv b >= c Then	False	True	False
If a = b Eqv a < c Then	False	False	True
If a <> b Eqv b > c Then	True	False	False

Для выяснения является ли хотя бы одно из отношений истинным, используется оператор **Or**. Если **хотя бы одно из отношений истин-но**, то и результат будет истинным; если же оба отношения ложны, то результат будет ложным.

a = 10, b = 8 c = 6			
Оператор	Первое усло-	Второе усло-	Результат логиче-

	вие	вие	ской операции
If $a > b$ Or $b > c$ Then	True	True	True
If a < b Or b >= c Then	False	True	True
If a = b Or a < c Then	False	False	False
If a <> b Or b > c Then	True	False	True

Если нужно выяснить, что истинно лишь одно из двух логических отношений, то используется операция **Xor**. В отличие от **Or** она дает истинный результат лишь в том случае, если истинно только одно выражение, но не оба одновременно.

a = 10, b = 8 c = 6			
Оператор	Первое усло-	Второе усло-	Результат логиче-
	вие	вие	ской операции
If a > b Xor b > c Then	True	True	False
If a < b Xor b >= c Then	False	True	True
If a = b Xor a < c Then	False	False	False
If a <> b Xor b > c Then	True	False	True

Логический оператор **Not** возвращает противоположное логическое значение.

a = 10, b = 8		
Оператор	Первое условие	Результат логической опе-
		рации
If Not (a > b) Then	True	False
If Not (a < b) Then	False	True

Not(10 > 8)' Возвращает False, .т.к. 10 > 8 Not(8 > 10)' Возвращает True.

Логические операции могут быть и более сложными, т.е. состоять больше чем из двух условий. Последовательность операций такая же, как и при вычислении математических функций. Операции в скобках имеют приоритет.

В таблице ниже даны примеры таких операций.

a = 10, b = 8 c = 6

Оператор	Результат логической опера-
	ции
If a>c xor a>b Or b>c Then	True
If a>c xor (a>b Or b>c) Then	False
If a>c and (a <b b<c)="" eqv="" td="" then<=""><td>True</td>	True
If a <> b Xor b > c Then	True

Теперь, основываясь на сказанном выше, составим программу для вычисления функции, заданной вначале согласно описанному выше алгоритму:

```
Sub Vetvl()

'объявляем переменные

Dim x, F As Single

'вводим значение x

x = CSng (InputBox("Введите x"))

'вычисляем значение F

If x <= -1 Or x >= 1 Then

F = x

ElseIf x > -1 And x < 1 Then

F = x ^ 2

End If

'Выводим значение F

MsgBox F

End Sub
```

В коде программы нам встречается новая функция **InputBox**, которая предназначена для ввода информации. Результатом работы этой функции является окно ввода информации на экране компьютера рис. 1.

ОК
Cancel

Рис. 1 Окно ввода информации InputBox

Эта функция пришла на замену оператору INPUT, из среды программирования QuickBasic для операционной системы DOS. Paнee, для ввода информации, мы использовали текстовые поля на форме. Однако для того, чтобы понять принципы программирования, необязательно каждый раз проектировать новую форму. Поэтому для ввода информации мы будем использовать также функцию **InputBox**. Pacсмотрим эту функцию подробнее.

Назначение. Выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа String, содержащее текст, введенный в поле.

Синтаксис.

InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

Синтаксис функции **InputBox** содержит следующие именованные аргументы:

рготрt (обязательный) - строковое выражение, отображаемое как сообщение в диалоговом окне («Введите *x*» на рис. 1). Максимальная длина строки **prompt** составляет приблизительно 1024 символов и зависит от ширины используемых символов. Строковое значение **prompt** может содержать нескольких физических строк. Для разделения строк допускается использование символа возврата каретки (Chr(13)), символа перевода строки (Chr(10)) или комбинацию этих символов (Chr(13) & Chr(10)).

title (необязательный) - строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения («Microsoft Word» на рис. 1).

default (необязательный) - строковое выражение, отображаемое в поле ввода как используемое по умолчанию, если пользователь не

введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым как на рис. 1.

хроѕ (необязательный) - числовое выражение, задающее расстояние по горизонтали между левой границей диалогового окна и левым краем экрана (в твипах). Если этот аргумент опущен, диалоговое окно выравнивается по центру экрана по горизонтали.

уроѕ (необязательный) - числовое выражение, задающее расстояние по вертикали между верхней границей диалогового окна и верхним краем экрана (в твипах). Если этот аргумент опущен, диалоговое окно помещается по вертикали примерно на одну треть высоты экрана.

helpfile (необязательный) - строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот аргумент указан, необходимо указать также и аргумент context.

context (необязательный) - числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот аргумент указан, необходимо указать также аргумент **helpfile**.

Если указаны оба аргумента, helpfile и context, пользователь имеет возможность нажатием клавиши F1 вызвать контекстную справку. В некоторых приложениях, например, в Microsoft Excel, также автоматически добавляется в диалоговое окно кнопка "Справка".

Если пользователь нажимает кнопку "*OK*" или клавишу *ENTER*, функция **InputBox** возвращает содержимое поля ввода. Если пользователь нажимает кнопку "Отмена", функция **InputBox** возвратит пустую строку ("").

В следующем примере приведены различные способы получения сведений от пользователя с помощью функции **InputBox**. Если аргументы *x* и *y* опущены, окно диалога автоматически выравнивается по центру по соответствующим осям. Переменная MyValue содержит значение, введенное пользователем, если была нажата кнопка *OK* или клавиша *ENTER*. Если же пользователь нажмет кнопку Отмена, то функция возвратит пустую строку.
```
Dim Message, Title, Default, MyValue

Message = "Введите число от 1 до 3" ' Сообщение-подсказка.

Title = "Пример" ' Заголовок.

Default = "1"' Значение по умолчанию.

' Вывод на экран сообщения, заголовка и значения по умолчанию.

MyValue = InputBox(Message, Title, Default)

' Получение справки. Кнопка "Справка" добавляется автоматически.

MyValue = InputBox(Message, Title, , , , "DEMO.HLP", 10)

' Размещение верхнего левого угла окна диалога в точке 100, 100.

MyValue = InputBox(Message, Title, Default, 100, 100)
```

Опять вернемся к программе вычисления функции. В рассмотренном ниже блоке в зависимости от введенного x вычисляется значение F, которое затем выводится на экран с помощью стандартного диалогового окна **MsgBox**.

```
If x \le -1 Or x \ge 1 Then

F = x

ElseIf x \ge -1 And x < 1 Then

F = x \land 2

End If
```

В нашем примере вместо оператора ElseIf можно использовать оператор Else без указания условия, т.к. это условие осталось единственным. Тогда эта часть программы будет выглядеть следующим образом:

```
If x \le -1 Or x \ge 1 Then

F = x

Else

F = x \land 2

End If
```

Важно! Хотелось бы обратить ваше внимание на правила хорошего стиля программирования. Из текста программы видно, что операторы смещены на несколько пробелов вправо. На выполнение программы это никак не влияет, но улучшает «читаемость» программы. Все операторы If, Else If, Else, End If должны писаться на одном вертикальном уровне, а операторы внутри них должны писаться с отступом вправо. Если внутри операторов встречаются вложенные блоки **If, Else If, Else, End If** то они также должны писаться с отступом, а операторы внутри вложенных блоков смещаются вправо относительно уже вложенных блоков. Например,

```
If x \le -1 Or x \ge 1 Then

IF x = 0

F= 5

Else

F = 3

End If

Else

F = x \land 2

End If
```

Это правило нужно соблюдать при написании ваших программ. И последнее замечание. Для вывода формулы

 $F(x) = \begin{cases} x, & \text{если } x \le -1 \\ x^2, & \text{если } -1 < x < 1 \\ x, & \text{если } x \ge 1. \end{cases}$

в документ Word можно использовать код объект OMath (см. предыдущие главы).

Варианты заданий.

Использование конструкций ветвления If...Then...ElseIf...EndIf

Задача. Составьте программу, которая производит расчет предложенной вам функции (табл. 2).

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин (для ввода величин функцию InputBox HE использовать!!!), кнопками для выполнения расчета, формирования отчета, выхода из программы. Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

Таблица 2

$F(x) = \begin{cases} x^2 - 3x + 9, \text{ если } x \le 3; \\ 1 \end{cases}$	$F(x) = \begin{cases} -x^2 + 3x + 9, \text{если } x \ge 3; \\ 1 \end{cases}$
1. $\left(\frac{1}{x^3+6}, \frac{1}{x^3+6}\right)$ если $x > 3$.	2. $\left(\frac{1}{x^3-6}\right)^{-1}$, eсли $x < 3$.
$F(x) = \begin{cases} 9, & \text{если } x \le -3; \\ \frac{1}{x^2 + 1}, & \text{если } x > -3. \end{cases}$ 3.	$F(x) = \begin{cases} 0, & \text{если } x \le 1; \\ \frac{1}{x+6}, & \text{если } x > 1. \end{cases}$
$F(x) = \begin{cases} -3x + 9, \ \text{если } x \le 7; \\ \frac{1}{x - 7}, \ \text{если } x > 7. \end{cases}$ 5.	$F(x) = \begin{cases} 3x - 9, & \text{если } x \le 7; \\ \frac{1}{x^2 - 4}, & \text{если } x > 7. \end{cases}$ 6.
$F(x) = \begin{cases} x^2, & \text{если } 0 \le x \le 3; \\ 4, & \text{если } x > 3 \text{ или } x < 0. \end{cases}$	$F(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \le 2; \\ \frac{1}{x^2 + 4x + 5}, & \text{если } x > 2. \end{cases}$ 8.
$F(x) = \begin{cases} x^2 - x, & \text{если } 0 \le x \le 1; \\ x^2 - \sin \pi x^2, \text{если } x > 1 \text{ или } x < 0. \end{cases}$	$F(x) = \begin{cases} -x^2 + x - 9, \text{ если } x \ge 8; \\ \frac{1}{x^4 - 6}, \text{ если } x < 8. \end{cases}$
11. $F(x) = \begin{cases} 4x^2 + 2x - 19, & \text{если } x \le -3,5; \\ -\frac{2x}{-4x+1}, & \text{если } x > 3,5. \end{cases}$	$F(x) = \begin{cases} -x^2 + 3x + 9, если x \le 3; \\ \frac{x}{x^2 + 1}, \qquad \text{если } x > 3. \end{cases}$ 12.
$F(x) = \begin{cases} -3x+9, & \text{если } x > 3; \\ \frac{x^3}{x^2+8}, & \text{если } x \le 3. \end{cases}$ 13.	$F(x) = \begin{cases} -x^3 + 9, & \text{если } x \le 13; \\ -\frac{3}{x+1}, & \text{если } x > 13. \end{cases}$ 14.
$F(x) = \begin{cases} 45x^2 + 5, & \text{если } x > 3,6; \\ \frac{5x}{10x^2 + 1}, & \text{если } x \le 3,6. \end{cases}$	$F(x) = \begin{cases} x^4 + 9, & \text{если } x < 3,2; \\ \frac{54x^4}{-5x^2 + 7}, & \text{если } x \ge 3,2. \end{cases}$ 16.
$F(x) = \begin{cases} 1,2x^2 - 3x - 9, \text{если } x > 3; \\ \frac{12,1}{2x^2 + 1}, & \text{если } x \le 3. \end{cases}$ 17.	$F(x) = \begin{cases} x^2 + 3x + 9, \text{ если } x \le 3; \\ \frac{\sin x}{x^2 - 9}, & \text{если } x > 3. \end{cases}$ 18.
$F(x) = \begin{cases} \cos 2x + 9, & \text{если } x > -4; \\ -\frac{\cos x}{x - 9}, & \text{если } x \le -4. \end{cases}$ 19.	$F(x) = \begin{cases} \ln x + 9, & \text{если } x > 0; \\ -\frac{x}{x^2 - 7}, & \text{если } x \le 0. \end{cases}$ 20.

Варианты заданий

$\left[-x^2-1,1x+9, \text{ если } x \le -3;\right]$	9-x, если $x > 1,1;$
$F(x) = \begin{cases} \frac{\ln(x+3)}{x^2+9}, & \text{если } x > -3. \end{cases}$	$F(x) = \begin{cases} -\frac{\sin 3x}{x^4 + 1}, & \text{если } x < -1,1. \end{cases}$ 22.
23. $F(x) = \begin{cases} -x^2, & \text{если } x > 7; \\ \frac{2^{-x}}{x^2 - 9}, & \text{если } x \le 7. \end{cases}$	24. $F(x) = \begin{cases} -x^2 - 9, & \text{если } x > 13; \\ -\frac{1}{x^2 + 9}, & \text{если } x \le 13. \end{cases}$
25. $F(x) = \begin{cases} 0, & \text{если } x \le 0; \\ x, & \text{если } 0 < x \le 1; \\ x^4, & \text{если } x > 1. \end{cases}$	26. $F(x) = \begin{cases} x^2 - x, & \text{если } 0 \le x \le 1; \\ x^2 - \sin \pi x^2, \text{если } x > 1 \text{ или } x < 0. \end{cases}$
27. $F(x) = \begin{cases} 45x^2 + 5, & \text{если } x > 3,6; \\ \frac{5x}{10x^2 + 1}, & \text{если } x \le 3,6. \end{cases}$	28. $F(x) = \begin{cases} -3x+9, & \text{если } x > 3; \\ \frac{x^3}{x^2+8}, & \text{если } x \le 3. \end{cases}$
29. $F(x) = \begin{cases} 9, & \text{если } x \le -3; \\ \frac{1}{x^2 + 1}, & \text{если } x > -3. \end{cases}$	30. $F(x) = \begin{cases} 4x^2 + 2x - 19, & \text{если } x \le -3,5; \\ -\frac{2x}{-4x+1}, & \text{если } x > 3,5. \end{cases}$
31. $F(x) = \begin{cases} \cos 2x + 9, & \text{если } x > -4; \\ -\frac{\cos x}{x - 9}, & \text{если } x \le -4. \end{cases}$	32. $F(x) = \begin{cases} -x^2 + x - 9, \text{ если } x \ge 8; \\ \frac{1}{x^4 - 6}, \text{ если } x < 8. \end{cases}$
33. $F(x) = \begin{cases} -x^2 + 3x + 9, если x \ge 3; \\ \frac{1}{x^3 - 6}, & если x < 3. \end{cases}$	34. $F(x) = \begin{cases} -x^2 + 3x + 9, если x \le 3; \\ \frac{x}{x^2 + 1}, & если x > 3. \end{cases}$
35. $F(x) = \begin{cases} 3x - 9, & \text{если } x \le 7; \\ \frac{1}{x^2 - 4}, & \text{если } x > 7. \end{cases}$	36. $F(x) = \begin{cases} -x^3 + 9, & \text{если } x \le 13; \\ -\frac{3}{x+1}, & \text{если } x > 13. \end{cases}$
37. $F(x) = \begin{cases} -3x+9, & \text{если } x \le 7; \\ \frac{1}{x-7}, & \text{если } x > 7. \end{cases}$	38. $F(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \le 2; \\ \frac{1}{x^2 + 4x + 5}, & \text{если } x > 2. \end{cases}$
39. $F(x) = \begin{cases} x^2 - 3x + 9, & \text{если } x \le 3; \\ \frac{1}{x^3 + 6}, & \text{если } x > 3. \end{cases}$	40. $F(x) = \begin{cases} 0, & \text{если } x \le 1; \\ \frac{1}{x+6}, & \text{если } x > 1. \end{cases}$

Лабораторная работа №7

Организация защиты программы от неверных действий пользователя

Цель работы:

В созданной программе при выполнении лабораторных работах № 4, 5 организовать защиту от неверных действий пользователя.

Порядок выполнения работы

В лабораторной работе № 4 был рассмотрен следующий пример.

Задача. Составьте программу, которая производит расчет по формуле $\frac{b + \sqrt{b^2 + 4 \cdot a \cdot c}}{2 \cdot a} - a^3 \cdot c + b^{-2}$ при различных значениях *a*, *b*, *c*.

Программа должна содержать форму, на которой должны располагаться текстовые поля для ввода величин, кнопки для выполнения расчета, формирования отчета и выхода из программы.

Сформировать отчет средствами VBA. Отчет должен содержать:

условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

Решая задачу, мы спроектировали следующую форму [1] рис. 2.

Расчет фор	мулы				
Pacu	нет по формул	ie:	frm	Formula	1
b	$+\sqrt{b^2+a^2}$	$4 \cdot a \cdot c$	o⊥ b- 2 –		
	2∙a	-a	C+U	۲ الا	Otvet
a=	b=	c=			
	Тассчитать	тхтв тхто Создать отчет	Выход		
	cmdRaschet	cmdOtchet	cmdExit		

Рис. 2. Форма для расчета формулы.

Мы написали программу, с помощью которой рассчитывается формула по введенным данным и создается отчет.

```
Dim a As Single, b As Single, c As Single, result As Single
Private Sub cmdRaschet Click()
     a = CSnq (txtA.Text)
    b = CSng (txtB.Text)
     c = CSng (txtC.Text)
     result = (b + Sqr(b ^ 2 + 4 * a^ 3 * c)) / (2 * a) - a ^ 3 *
     c + b ^ (−2)
     lblOtvet.Caption = CStr(result)
End Sub
Private Sub cmdOtchet Click()
    frmFormula.Hide
    Dim docNew As Document 'объявляем новый документ
    создаем новый документ и устанавливаем на него ссылку
    Set docNew = Documents.Add
    With Selection
      'Печатаем заголовок
      .InsertAfter "Программа для расчета формулы."
      'напечатанный заголовок форматируем по середине
      .ParagraphFormat.Alignment = wdAlignParagraphCenter
      создаем отступ для первой строки
      .ParagraphFormat.FirstLineIndent = CentimetersToPoints(0.7)
      устанавливаем для заголовка полужирный шрифт, 14 размера
      .Font.Bold = True
      .Font.Size = 14
      'вставляем два параграфа
      .InsertParagraphAfter
      .InsertParagraphAfter
      'Перемещаем курсор в конец документа
      .EndOf Unit:=wdSection
      'Печатаем задание
      .InsertAfter "Задание. "
      .ParagraphFormat.Alignment = wdAlignParagraphJustify
```

.EndOf Unit:=wdSection

.InsertAfter "Составить программу для расчета формулы. Программа должна содержать форму, которая должна иметь текстовые поля для ввода величин, кнопки для расчета, формирования отчета, выхода из программы."

.InsertParagraphAfter

.InsertAfter "Сформировать отчет средствами VBA. Отчет должен содержать:"

.InsertParagraphAfter

.InsertAfter "условие задачи;"

.InsertParagraphAfter

.InsertAfter "формулу расчета с обозначениями и подставленными вместо них числами;"

.InsertParagraphAfter

.InsertAfter "полученный результат."

.Font.Bold = False

.InsertParagraphAfter

.EndOf Unit:=wdSection

'Вывод результатов в отчет

.Font.Bold = True

```
.InsertAfter "Решение. "
```

.InsertParagraphAfter

.EndOf Unit:=wdSection

.Font.Bold = False

.InsertAfter "Для a = " & CStr(a) & ", для b = " & CStr(b) & ", для c = " & CStr(c) & " значение формулы:"

.InsertParagraphAfter

.InsertParagraphAfter

.EndOf Unit:=wdSection

'Формируем формулу с обозначениями

```
'Formula = "(b+" & ChrW(8730) & "(b^2+4 · a · c))/(2 · a)-
a^3 · c+b^-2=" & "(" & b & "+" & ChrW(8730) & "(" & b & "^2" & "+4"
& " · " & a & " · " & c & "))/" & "(2 · " & a & ")-" & a & "^3 · " & c &
"+" & b & "^-2=" & result
```

```
Formula = "(b+" \& ChrW(8730) \& "(b^{2}+4 \cdot a \cdot c))/(2 \cdot a) - a^{3} \cdot c + b^{-}
2=(" & b & "+" & ChrW(8730) & "(" & b & "^2+4·" & a & "·" & c &
"))/(2·" & a & ")-" & a & "^3·" & c & "+" & b & "^-2=" & result
     'Объявляем объект - диапазоноbjRange
    Dim objRange As Range
    'ОбъявляемобъктоbjЕq, какформулу
    Dim objEq As OMath
    ' УстанавливаемобъектоbjRange, как выделенную часть документа.
В нашем случае место под курсором
    Set objRange = Selection.Range
    'свойству text объекта objRange присваиваем код формулы
    objRange.Text = Formula
    'преобразуемобъектоbjRange в математическую формулу
    Set objRange = Selection.OMaths.Add(objRange)
    ' создаем объект objEq, образуя его из объекта objRange
    Set objEq = objRange.OMaths(1)
    'Преобразуем формулу из линейного вида к профессиональному
    objEq.BuildUp
    End With
    frmFormula.Show
End Sub
    Private Sub cmdExit Click()
        End
    End Sub
```

Кажется, что мы все сделали. Но это не так. Мы не учли следующее:

1. Пользователь может нажать кнопку **Рассчитать**, забыв ввести данные в текстовые поля. В формулу будет подставлена пустая величина и приложение, в этом случае, выдаст ошибку и прекратит выполнение программы.

2. Пользователь может ввести неверные данные, например, не числовые, что опять может привести к ошибке выполнения программы.

3. Пользователь может ввести данные правильно, однако следует учесть, что подкоренное выражение в формуле не может быть меньше 0, а также знаменатель формулы не может быть равен 0. Если пользователь введет такие данные, которые не удовлетворят математическим ограничениям, это также приведет к ошибке.

4. Пользователь может нажать кнопку Создать отчет, не рассчитав прежде значение по формуле.

Для того, чтобы программа правильно реагировала на введенные не верно данные, нужно создать код проверки правильности ввода. У программистов такой код обычно называется «защита от дурака».

Для кода проверки правильности ввода данных можно использовать рассмотренную ранее конструкцию

```
If условие Then
[операторы]
[ElseIf условие-n Then
[операторы_elseif] ...
[Else
[операторы_else]]
End If
```

Составим алгоритм проверки введенных данных при нажатии на кнопку Рассчитать:

Если а не введено Выдать сообщение и вернуться на форму Иначе если а введено не правильно (а - не число) Выдать сообщение и вернуться на форму Иначе если b не введено Выдать сообщение и вернуться на форму Иначе если b введено не правильно (b - не число) Выдать сообщение и вернуться на форму Иначе если с не введено Выдать сообщение и вернуться на форму Иначе если с введено не правильно (c - не число) Выдать сообщение и вернуться на форму

```
Иначе (если введено все правильно)

Если знаменатель равен 0 (а=0)

Выдать сообщение и вернуться на форму

Иначе если значение под корнем < 0

Выдать сообщение и вернуться на форму

Иначе если значение b^(-2)=0 (b=0)

Выдать сообщение и вернуться на форму

Иначе (если все верно)

Рассчитать значение формулы

Конец если

Конец если
```

Составим также алгоритм для проверки нажатия кнопки Создать отчет:

Если поле ответа пустое Выдать сообщение и вернуться на форму Иначе Создать отчет Конец если

Private Sub cmdRaschet Click()

С учетом первого алгоритма, изменим текст программы для события нажатия кнопки Рассчитать:

```
'Объявляем переменные, значения которых вводятся в текстовые поля
Dim a As Single
Dim b As Single
Dim c As Single
```

```
'Объявляем переменные, которые показывают то, что введенные в поля
данные являются числом или не числом. Если введенные в поля данные
яв-ляются числом, то значение будет истина, если не числом, то ложь.
Dim AA As Boolean
Dim BB As Boolean
Dim CC As Boolean
```

```
'очищаем метку, в которую выводится рассчитанный результат lbl0tvet.Caption = ""
```

' Функция IsNumeric проверяет, является ли значение данного выражения числом. И в зависимости от этого переменные принимают значения истина или ложь.

```
AA = IsNumeric(txtA.Text)
     BB = IsNumeric(txtB.Text)
     CC = IsNumeric(txtC.Text)
     If txtA.Text = "" Then 'заполнено ли текстовое поле а
       MsqBox ("Не введены данные для переменной а")
       txtA.SetFocus 'устанавливаем фокус на текстовое поле
     ElseIf Not (AA) Then 'Если введено не число
       MsqBox ("Не правильно введены данные для переменной а (а - не
число)")
       txtA.SetFocus
     ElseIf txtB.Text = "" Then 'заполнено ли текстовое поле b
       MsgBox ("Не введены данные для переменной b")
       txtB.SetFocus
     ElseIf Not (BB) Then 'Если введено не число
       MsgBox ("Не правильно введены данные для переменной b (b - не
число)")
      txtB.SetFocus
     ElseIf txtC.Text = "" Then 'заполнено ли текстовое поле с
      MsgBox ("Не введены данные для переменной с")
       txtC.SetFocus
     ElseIf Not (CC) Then 'Если введено не число
      MsqBox ("Не правильно введены данные для переменной с (с - не
число)")
       txtC.SetFocus
     Else
       'переводим данные в числовую форму
       a = CSng (txtA.Text)
      b = CSng (txtB.Text)
       c = CSng (txtC.Text)
       If a = 0 Then 'Если знаменатель равен 0
        MsgBox ("Число а не может быть равно 0, т.к. знаменатель в этом
случае станет равен 0")
         txtA.SetFocus
       ElseIf b ^ 2 + 4 * a * c ^ 3 < 0 Then 'Если значение под корнем <
0
        MsgBox ("Значение под корнем < 0")
         txtA.SetFocus
       ElseIf b = 0 Then 'Если число b^{(-2)} равное 0
        MsgBox ("Число b не может быть равно 0, т.к. b^(-2) будет равен
1/b^2, а знаменатель не может равняться 0")
         txtB.SetFocus
       Else 'Если все нормально
         'рассчитываем формулу
         result = (b + Sqr(b ^ 2 + 4 * a * c ^ 3)) / (2 * a) - a ^ 3 * c
+ b ^ (-2)
         'заносим ответ в метку на форме
```

```
lblOtvet.Caption = CStr(result)
End If
End If
End Sub
```

Используя второй алгоритм, напишем программу для проверки нажатия кнопки Создать отчет.

```
Private Sub cmdOtchet_Click()

If lblOtvet = "" Then

MsgBox ("Вы не расчитали формулу")

cmdRaschet.SetFocus

Else

'Обратите внимание! После оператора Else идет код программы, как

в примере из Лабораторной работы №4 для создания отчета

End If

End Sub
```

Лабораторная работа №8

Организация защиты программы от неверных действий пользователя

Цель работы:

Закрепить знания, полученные при изучении темы «Организация защиты программы от неверных действий пользователя»

Варианты заданий

Задача. Для программы созданной в лабораторной работе №5 создайте программный код защиты от неправильного ввода данных.

Лабораторная работа №9

Конструкция выбора Select Case

Цель работы:

1. Изучить конструкцию выбора Select Case

При составлении вашей программы может так случиться, что потребуется проверять очень много условий. Если для этого использовать рассмотренную выше конструкцию **If...Then...Else...EndIf**, то можно столкнуться с трудностями при создании кода программы, т.к. в этом случае код может стать громоздким и трудным для понимания и восприятия. В этом случае можно воспользоваться конструкцией выбора **Select Case**, которая выполняет один из нескольких блоков операторов в зависимости от значения проверяемого выражения.

Синтаксис:

```
Select Case выражение
 [Case список выражений - 1
      [операторы- 1]] ...
 [Case список выражений - 2
      [операторы- 2]] ...
 ...
 [Case список выражений - п
      [операторы- n]] ...
 [Case Else
      [операторы_else]]
End Select
```

Синтаксис конструкции Select Case содержит следующие элементы:

Выражение (обязательный) - любое числовое или строковое выражение.

Список выражений (обязательный при наличии предложения Case) - список с разделителями, состоящий из одной или нескольких

форм следующего вида: выражение; выражение То выражение; Is, оператор сравнения, выражение. Ключевое слово То задает диапазон значений от меньшего до большего. При использовании ключевого слова То перед ним записывается меньшее значение. Ключевое слово Is задает диапазон значений с помощью операторов сравнения.

Операторы (необязательный) - один или несколько операторов, выполняемых в том случае, если выражение совпадает с любым компонентом (условием) списка выражений.

Операторы else (необязательный) - один или несколько операторов, выполняемых в том случае, если выражение не совпадает ни с одним из предложений Case.

Если выражение совпадает с любым выражением из списка выражений в предложении Case, выполняются все операторы, следующие за данным предложением Case до следующего предложения Case, или, для последнего предложения, до оператора End Select. Затем управление передается к следующему за End Select оператору. Если выражение совпадает с выражениями из списка в нескольких предложениях Case, выполняется только первый подходящий набор операторов.

Предложение Case Else задает блок операторов else, которые будут выполнены, если не обнаружено ни одно совпадение выражения и компонентов списка выражений ни в одном из остальных предложений Case. Хотя данное предложение не является обязательным, рекомендуется помещать предложение Case Else в блок Select Case, чтобы предусмотреть появление неожиданных значений выражения в ходе выполнения программы. Если ни в одном предложении Case список выражений не содержит компонента, отвечающего аргументу выражение, и отсутствует оператор Case Else, выполнение продолжается с оператора, следующего за оператором End Select.

В каждом предложении **Case** допускается использование нескольких выражений или диапазонов. Например, допустима следующая строка программного кода:

86

Case 1 To 4, 7 To 9, 11, 13, Is > MaxNumber

Как и для конструкции ветвления If...Then...ElseIf...EndIf допускаются вложенные операторы Select Case. Каждому вложенному оператору Select Case должен соответствовать оператор End Select.

Задача. Составить программу для вычисления функции:

$$z = \begin{cases} \log_3 |ab| , \text{при } a + b = 2, \\ a^2 + \operatorname{ctg} b, \text{при } a + b = 3, a + b = 4 \text{ или } a + b = 5, \\ a - \sin^2 b, \text{при } 7 \le a + b \le 10, \\ \sqrt{a^2 + b^2}, \text{при } a + b > 12, \\ 1, \text{для всех остальных вариантов.} \end{cases}$$

Приведем код составленной программы:

```
Sub SelectCaseUse()
  'объявляем переменные
  Dim a As Single, b As Single, z As Single
 Dim n As Integer
 'вводим данные
 a = CSng(InputBox("Введите a"))
 b = CSng(InputBox("Введите b"))
 Select Case a + b 'анализируемое выражение
   Сазе 2 'если сумма равна 2
     z = Log(Abs(a * b)) / Log(3)
     n = 1
    Case 3, 4, 5 'если сумма равна 3 или 4 или 5
      z = a ^ 2 + 1 / Tan(b)
      n = 2
    Case 7 То 10 'если сумма лежит в диапазоне от 7 до 10
     z = a - Sin(b) ^ 2
     n = 3
    Case Is > 12 / если сумма больше 12
     z = Sqr(a^{2} + b^{2})
     n = 4
    Case Else 'во всех остальных случаях
      z = 1
     n = 5
  End Select
 MsgBox "a = " & CStr(a) & Chr(13) & "b = " & CStr(b)
  & Chr(13) & "z = " & CStr(z) & Chr(13) _
  & "Вычислено по формуле № " & CStr(n)
End Sub
```

По правилу хорошего стиля программирования операторы внутри конструкций Case должны писаться с отступом вправо. Это улучшает читаемость программы. В рассматриваемом примере, рассмотрено как это делается.

При использовании функции **MsgBox**, применяется функция Chr(13). Данная функция соответствует коду символа клавиши Enter (перевод строки), равному 13. В результате функция **MsgBox** работает следующим образом: сначала печатается текст, затем, дойдя до функции Chr(13) осуществляется перевод на новую строку и последующий текст печатается в новой строке. В приведенном примере текст в окне **MsgBox** печатается в четырех строках.

Варианты заданий.

Задача. Составьте программу для решения задачи.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин (для ввода величин функцию InputBox HE использовать!!!), метками (Label) для вывода вычисленных значений (для вывода функцию MsgBox HE использовать!!!), кнопками для выполнения расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты заданий

- 1. Создайте программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество уроков в вашем классе в этот день.
- 2. Создайте программу, позволяющую по последней цифре числа определить последнюю цифру квадрата этого числа.
- 3. Составьте программу, которая по заданным году и номеру месяца определяет количество дней в этом месяце.
- 4. Для каждой введенной цифры (0 9) выведите соответствующее ей название на английском языке (0 zero, 1 one, 2 two,...).
- 5. Составьте программу, которая по данному числу (1-12) выводит название соответствующего ему месяца.
- Составьте программу, позволяющую получить словесное описание школьных отметок (1 «плохо», 2 «неудовлетворительно», 3 «удовлетворительно», 4 «хорошо», 5 «отлично»).
- 7. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент), Составьте программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.
- 8. Пусть элементами прямоугольного равнобедренного треугольника являются: катет *a*, гипотенуза *b*, высота *h*, опущенная из вершины прямого угла на гипотенузу, площадь *S*. Составьте программу, которая по заданному обозначению и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.
- 9. Создайте программу, которая по номеру месяца выдает название следующего за ним месяца (при *m* 1 получаем февраль, 4 май и т.д.).
- Создайте программу, которая бы по введенному номеру времени года (1 - зима, 2 - весна, 3 - лето, 4 - осень) выдавала бы соответствующие этому времени года месяцы и количество дней в каждом из месяцев.
- 11. В старояпонском календаре был принят 12-летний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра,

зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Создайте программу, которая выводит номер некоторого года и печатает его название по старояпонскому календарю (для справки 1996 г. — год Крысы — начало очередного цикла).

- 12. Для целого числа *k* от 1 до 99 напечатайте фразу «Мне *k* лет», учитывая при этом, что при некоторых значениях *k* слово «лет» надо изменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.
- Создайте программу, которая бы по введенному номеру единицы измерения (1 дециметр, 2 километр, 3 метр, 4 -миллиметр, 5 сантиметр) и длине отрезка *L* выдавала бы соответствующее значение длины отрезка в метрах.
- 14. Создайте программу, которая по вводимому числу от 1 до 11 (номеру класса) выдает соответствующее сообщение «Привет, *k*-классник». Например, если *k* = 1, «Привет, первоклассник»; если *k* = 4 «Привет, четвероклассник».
- 15. Создайте программу, которая по введенному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то должно получиться 1 января - Новый год, 7 января - Рождество).
- 16. Дано натуральное число *N*. Если оно делится на 4, вывести на экран ответ N = 4k (где k соответствующее частное); если остаток от деления на 4 равен 1, то N = 4k+1; если остаток от деления на 4 равен 2, то N = 4k+2; если остаток от деления на 4 равен 3, то N = 4k+3. Например, $12 = 4\cdot3$, $22 = 4\cdot5+2$.
- 17. Имеется пронумерованный список деталей: 1 шуруп, 2 гайка,
 3 винт, 4 гвоздь, 5 болт. Составьте программу, которая по номеру детали выводит на экран ее название.
- 18. Составьте программу, позволяющую по последней цифре данного числа определить последнюю цифру куба этого числа.
- 19. Составьте программу, которая для любого натурального числа печатает количество цифр в записи этого числа.

- 20. Даны два действительных положительных числа x и y. Арифметические действия над числами пронумерованы (1 - сложение, 2 вычитание, 3 - умножение, 4 - деление). Составьте программу, которая по введенному номеру выполняет то или иное действие над числами.
- 21. Создайте программу, которая бы по введенному номеру единицы измерения (1 килограмм, 2 миллиграмм, 3 грамм, 4 тонна, 5 центнер) и массе *М* выдавала бы соответствующее значение массы в килограммах.
- 22. Пусть элементами равностороннего треугольника являются: сторона *a*; площадь *S*; высота *h*; радиус вписанной окружности *r*; радиус описанной окружности *R*. Составьте программу, которая по заданному обозначению и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.
- 23. Найдите произведение цифр заданного *k*-значного числа.
- 24. Вычислите номер дня в невисокосном году по заданным числу и месяцу.
- 25. Пусть элементами прямоугольного равнобедренного треугольника являются: катет *a*, гипотенуза *b*, высота *h*, опущенная из вершины прямого угла на гипотенузу, площадь *S*. Составьте программу, которая по заданному обозначению и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.
- 26. Составьте программу, позволяющую по последней цифре данного числа определить последнюю цифру куба этого числа.
- 27. Составьте программу, позволяющую получить словесное описание школьных отметок (1 «плохо», 2 «неудовлетворительно», 3 «удовлетворительно», 4 «хорошо», 5 «отлично»).
- 28. Для целого числа k от 1 до 99 напечатайте фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо изменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.

- 29. Составьте программу, которая по заданным году и номеру месяца определяет количество дней в этом месяце.
- 30. В старояпонском календаре был принят 12-летний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Создайте программу, которая выводит номер некоторого года и печатает его название по старояпонскому календарю (для справки 1996 г. — год Крысы — начало очередного цикла).
- 31. Создайте программу, которая по введенному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то должно получиться 1 января - Новый год, 7 января - Рождество).
- 32. Создайте программу, которая бы по введенному номеру единицы измерения (1 дециметр, 2 километр, 3 метр, 4 -миллиметр, 5 сантиметр) и длине отрезка *L* выдавала бы соответствующее значение длины отрезка в метрах.
- 33. Создайте программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество уроков в вашем классе в этот день.
- 34. Составьте программу, которая для любого натурального числа печатает количество цифр в записи этого числа.
- 35. Создайте программу, которая бы по введенному номеру времени года (1 - зима, 2 - весна, 3 - лето, 4 - осень) выдавала бы соответствующие этому времени года месяцы и количество дней в каждом из месяцев.
- 36. Для каждой введенной цифры (0 9) выведите соответствующее ей название на английском языке (0 zero, 1 one, 2 two,...).
- 37. Имеется пронумерованный список деталей: 1 шуруп, 2 гайка,
 3 винт, 4 гвоздь, 5 болт. Составьте программу, которая по номеру детали выводит на экран ее название.
- 38. Создайте программу, позволяющую по последней цифре числа определить последнюю цифру квадрата этого числа.

- 39. Составьте программу, которая по данному числу (1-12) выводит название соответствующего ему месяца.
- 40. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент), Составьте программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.

Лабораторная работа №10 Использование конструкций ветвления If...Then...ElseIf...EndIf, операторов выбора Select Case

Задача. Составьте программу для решения задачи либо с помощью конструкций ветвления If...Then...ElseIf...EndIf либо оператором выбора Select Case, в зависимости от условия и Ваших предпочтений.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

1. Определить, является ли число а делителем числа b.

2. Известны два значения расстояния: одно выражено в километрах, другое — в футах (1 фут = 0,45 м). Определите, какое из значений расстояния меньше?

3. Известны два значения скорости: одно выражено в километрах в час, другое — в метрах в секунду. Определите, какое из значений скорости больше?

4. Даны радиус окружности и сторона квадрата. Определите для различных значений, у какой фигуры площадь больше?

5. Известны площади круга и квадрата. Определите:

а) впишется ли круг в квадрат;

б) впишется ли квадрат в круг.

6. Даны объемы и массы двух тел из разных материалов. Определите, материал какого из тел имеет большую плотность?

7. Известны сопротивления двух не соединенных друг с другом участков электрической цепи и напряжение на каждом из них. Определите, по какому участку протекает меньший ток?

8. Дано двузначное число. Определите:

а) какая из его цифр больше, первая или вторая;

б) одинаковы ли его цифры.

9. Дано двузначное число. Определите:

а) является ли сумма его цифр двузначным числом;

б) больше ли числа *а* сумма его цифр.

10. Дано трехзначное число. Выяснить, является ли оно палиндромом («перевертышем»), то есть таким числом, десятичная запись которого читается одинаково слева направо и справа налево.

11. Дано трехзначное число. Определите, какая из его цифр больше:

а) первая или последняя;

б) первая или вторая;

в) вторая или последняя.

12. Даны три вещественных числа. Необходимо возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — значения которых отрицательны.

13. Даны две точки в прямоугольной системе координат $A(x_1, y_1)$ и $B(x_2, y_2)$. Составьте алгоритм, определяющий, которая из точек находится ближе к началу координат.

14. Даны два угла треугольника измеренные в градусах. Определите, существует ли такой треугольник, и если да, то является ли он прямоугольным.

15. На плоскости X0У задана своими координатами точка А. Укажите, где она расположена (на какой оси или в каком квадранте).

16. Грузовой автомобиль выехал из одного города в другой со скоростью v_1 км/ч. Через некоторое время t ч в этом же направлении вы-

ехал легковой автомобиль со скоростью v_2 км/ч. Составьте программу, определяющую, догонит ли легковой автомобиль грузовой через t_1 ч после своего выезда.

17. Создайте программу — моделирующую состояние пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60 °C.

18. Создайте программу, которая анализирует данные о возрасте и относит человека к одной из четырех групп: дошкольник, учащийся, работающий, пенсионер. Возраст вводится с клавиатуры.

19. Составьте программу, определяющую, пройдет ли график функции $y = ax^2 + bx + c$ через заданную точку с координатами (*m*, *n*).

20. Заданы два натуральных числа. Определите, является ли среднее арифметическое этих чисел целым числом.

21. Заданы три стороны треугольника *a*,*b* и *c*. Определите, является ли этот треугольник прямоугольным и какая из сторон является гипотенузой.

22. Заданы два целых числа. Определите, являются они оба четными или оба нечетными, или какое из них четное, а какое нечетное.

23. Заданы координаты вершин треугольника и точки. Определите, лежит ли точка внутри треугольника.

24. Создайте программу нахождения суммы большего и меньшего из трех чисел.

25. Создайте программу, по длинам сторон распознающую среди всех треугольников *АВС* прямоугольные. Если таковых нет, то вычислите величину угла *С*.

26. Определите, являются ли они вершинами параллелограмма четыре точки: $A_1(x_1, y_1), A_2(x_2, y_2), A_3(x_3, y_3), A_4(x_4, y_4)$.

27. Даны три точки: $A(x_1, y_1), B(x_2, y_2)$ и $C(x_3, y_3)$. Определите, расположены ли они на одной прямой. Если нет, то вычислить угол *ABC*.

28. На оси 0Х расположены три точки: *a*,*b*,*c*. Определите, какая из точек, *b* или *c*, расположена ближе к *a*.

96

29. Дан круг с радиусом *R*. Определите, впишется ли правильный треугольник со стороной *a* в этот круг.

30. Дано число x. Напечатайте в порядке возрастания числа $\sin x, \cos x, \ln x$. Если при каком-либо значении x некоторые из выражений теряют смысл, выведите сообщение об этом и сравните значения только тех выражений, которые имеют смысл.

Задания для самостоятельной работы №11

Использование конструкций ветвления If...Then...Elself...EndIf, оператором выбора Select Case

Задача. Составьте программу для решения задачи либо с помощью конструкций ветвления If...Then...ElseIf...EndIf либо оператором выбора Select Case, в зависимости от условия и Ваших предпочтений.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

1. Заданы размеры *А*, *В* прямоугольного отверстия и размеры *x*, *y*, *z* кирпича. Определите, пройдет ли кирпич через отверстие.

2. Составьте программу, осуществляющую перевод величин из радиан в градусы и наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.

3. Создайте программу, которая по заданным трем числам определяет, является ли сумма каких-либо двух из них положительной.

4. Создайте программу, которая по паролю определяет уровень доступа сотрудника к секретной информации в базе данных. Доступ к базе имеют только шесть человек, разбитых на три группы по уровням доступа. Они имеют следующие пароли: 9583, 1747 — доступны модули баз *A*, *B*, *C*; 3331, 7922 - доступны модули баз *B*, *C*; 9455, 8997 доступен модуль базы *C*.

5. Составьте программу, имитирующую применения компьютера в книжном магазине. Компьютер запрашивает стоимость книг, сумму денег, внесенную покупателем; если сдачи не требуется, печатает на экране «спасибо»; если денег внесено больше, чем необходимо, то печатает «возьмите сдачу» и указывает сумму сдачи; если денег недостаточно, то печатает сообщение об этом и указывает размер недостающей суммы

6. Даны три вещественных числа *a*,*b*,*c*. Проверьте:

а) выполняется ли неравенство *a < b < c*;

б) выполняется ли неравенство b > a > c.

7. Определите, является ли треугольник со сторонами *a*,*b*,*c* равнобедренным.

8. Определите, является ли треугольник со сторонами *a*,*b*,*c* равносторонним.

9. Год является високосным, если его номер кратен 4. Из кратных 100 високосными являются лишь те года, которые кратны также 400 (например, 1700, 1800 и 1900 - не високосные года, 2000 — високосный). Дано натуральное число *п*. Определите, является ли високосным год с номером равным *n*.

10. Даны вещественные положительные числа a,b,c,x,y. Выясните, пройдет ли кирпич с ребрами a,b,c в прямоугольное отверстие со сторонами x и y. Вносить кирпич в отверстие разрешается только так, чтобы

каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

11. Дано натуральное число *n*(*n* ≤ 9999). Выясните, различны ли все четыре цифры этого числа (если оно записано четырьмя цифрами). Например, в числе 3678 все цифры различны, в числе 0023 — есть две одинаковые.

12. Определите, является ли заданное шестизначное число «счастливым». («Счастливым» называют такое шестизначное число, у которого сумма его первых трех цифр равна сумме трех последних цифр.)

13. Дано целое число *k*(1≤*k*≤365). Определите, каким будет *k*-й день года, выходным (суббота и воскресенье) или рабочим, если 1 января — понедельник.

14. Траектория снаряда, вылетающего из орудия под углом α задается уравнениями $x = v_0 t \cos \alpha; y = v_0 t \sin \alpha - gt^2/2$ где g = 9,8 м/с² — ускорение свободного падения, t — время, v_0 - начальная скорость. При заданных значениях α и v_0 определить, поразит ли снаряд цель высотой p, расположенную в вертикальной плоскости ствола орудия на расстоянии r и на высоте h рис. 3.

15. Дана точка A(x, y). Определите, принадлежит ли она треугольнику, вершины которого имеют координаты $(x_1y_1), (x_2, y_2), (x_3, y_3)$.



Рис. 3.

16. Создайте программу, определяющую, перпендикулярны ли прямые, $A_1x + B_1y + C_1 = 0$ и $A_2x + B_2y + C_2 = 0$. Если нет, то найдите угол между ними.

17. Найдите координаты точек пересечения прямой y = kx+b и окружности радиусом *R* с центром в начале координат. В каких квадрантах находятся точки пересечения? Если точек пересечения нет или прямая касается окружности, выдайте соответствующее сообщение.

18. Для функций, заданных графически рис. 4, определите значение *у* при заданном значении *х*.



Рис. 4.

19. Для функций, заданных графически рис.5, определите значение *у* при заданном значении *х*.



Рис. 5.

20. Для функций, заданных графически рис.6, определите значение *у* при заданном значении *х*.



Рис. 6.

21. Даны три различных целых числа. Определите, какое из них (первое, второе или третье):

а) самое большое;

б) самое маленькое;

в) среднее (средним назовем число, которое больше наименьшего из данных чисел, но меньше наибольшего).

22. Составьте программу нахождения произведения двух наименьших из трех различных чисел.

23. Создайте программу, которая на ввод знака препинания выдает на экран его название. Например, на ввод точки выводится текст: «Это точка».

24. Создайте программу, которая при вводе названия фигуры (треугольник, квадрат, ромб, прямоугольник и т. п.) выводит формулу, по которой вычисляется площадь этой фигуры

25. Создайте программу, которая при вводе времени суток выводит соответствующее пожелание (доброго утра, доброго дня, доброго вечера или спокойной ночи).

26. В некотором учебном заведении действуют следующие правила приема. Абитуриенты сдают три экзамена. Если они набирают не менее 13 баллов, то это дает право поступить на дневное отделение, от 10 до 12 — на вечернее, от 8 до 9 — на заочное; ниже 8 баллов означает отказ в приеме на обучение. Создайте программу, которая в

зависимости от суммы набранных баллов сообщает абитуриенту о его правах на поступление.

27. Все клетки шахматной доски пронумерованы от 1 до 64 так, что первая строка клеток имеет номера от 1 до 8 вторая — от 9 до 16 и т. д. Создайте программу, которая при вводе номера клетки выводит номера всех клеток, имеющих с ней общую сторону.

28. В древнем японском календаре был принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цветов: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крыса, корова, тигр, заяц, дракон, змея, лошадь, овца, обезьяна, курица, собака и свинья. Например, 1984 год — год начала очередного цикла назывался годом зеленой крысы.

Составьте программу, которая по заданному номеру года нашей эры *n* выводит его название по древнему японскому календарю. Рассмотрите два случая:

a) значение $n \ge 1984$;

б) значение *п* может быть любым натуральным числом.

29. Создайте программу, которая по номеру дня недели (целому числу от 1 до 7) и четной или нечетной недели выводит в качестве результата количество занятий (пар) в вашей группе в этот день.

30. Даны два вещественных положительных числа *x* и *y*. Арифметические действия над числами пронумерованы (сложение — 1, вычитание — 2, умножение — 3, деление — 4). Составьте программу, которая по введенному номеру выполняет то или иное действие над числами.

102

Лабораторная работа №12

Циклические конструкции. Циклы со счетчиком (For...Next). Вычисление рядов.

Циклы являются важнейшими структурами программ. Ни одна сложная программа не обходится без циклических конструкций.

Циклом называется многократное повторение одного или нескольких операторов, составляющих тело цикла. Существует два основных вида циклов: циклы со счетчиком (For...Next) и циклы по условию (Do...Loop).

Циклы со счетчиком часто еще называют циклами по параметру. Конструкция **For...Next** используется для выполнения группы операторов указанное число раз. Циклы **For** используют в качестве счетчика переменную, значение которой увеличивается или уменьшается при каждом выполнении цикла.

Синтаксис:

```
For счетчик = начало То конец [Step шаг]
[операторы]
[Exit For]
[операторы]
Next [счетчик]
```

Синтаксис конструкции **For...Next** содержит следующие элементы:

Счетчик (обязательный) - числовая переменная, используемая в качестве счетчика цикла. Эта переменная записывается после ключевого слова For и не может принадлежать к типу Boolean или быть элементом массива.

Начало (обязательный) - начальное значение переменной счетчик. Является числом или числовой переменной, определяемой перед входом в цикл и записывается после параметра счетчик и знака равно (=). Конец (обязательный) - конечное значение переменной счетчик. Является числом или числовой переменной, определяемой перед входом в цикл и записывается после ключевого слова **То**.

Шаг (необязательный) - является числом или числовой переменной, определяемой перед входом в цикл. Записывается после ключевого слова **Step**, обозначающего значение, на которое изменяется счетчик при каждом выполнении тела цикла. Если это значение не задано, по умолчанию шаг равен единице. Аргумент параметра **шаг** может быть как положительным, так и отрицательным. Если **шаг** больше нуля, то параметр **начало** должен быть меньше параметра **конец**. Если шаг меньше нуля, то параметр **начало** должен быть больше параметра **конец**. Если эти условия не соблюдаются, то операторы внутри цикла выполнятся не будут и программа начнет выполнятся с оператора следующего после ключевого слова **Next**.

Операторы (необязательный) - один или несколько операторов между For и Next, которые выполняются указанное число раз.

Альтернативный способ выхода из цикла предоставляет оператор **Exit For**. В любых местах цикла может размещаться любое число таких операторов. Оператор **Exit For** часто применяется вместе с проверкой некоторого условия (например, **If...Then**). Этот оператор передает управление оператору, непосредственно следующему за оператором **Next**.

Цикл For...Next работает следующим образом:

1. Переменной счетчик присваивается значение параметра начало:

счетчик = начало

2. Далее осуществляется вход в тело цикла и выполняются операторы в теле цикла.

Если встречается оператор Exit For происходит выход из цикла и выполняется оператор непосредственно следующий за ключевым словом Next.

3. Дойдя до ключевого слова Next программа изменяет значение счетчик на величину указанную в параметре шаг после ключевого

слова Step (счетчик = счетчик + шаг) или на единицу если ключевое слово Step отсутствует (счетчик = счетчик + 1).

4. Если значение счетчик выйдет за пределы параметра конец, то выполнение цикла прекратится и будет выполнен оператор непосредственно следующий за ключевым словом Next. Если значение счетчик меньше или равно параметру конец, то произойдет возврат на первый оператор после внутри цикла.

Операторы внутри цикла выполняются до тех по, пока значение **счетчика** не сравняется со значением параметра **конец**. После этого цикл завершается, и выполнение программы продолжается с оператора, следующего за оператором **Next**.

Допускается организация вложенных циклов For...Next (один цикл For...Next располагается внутри другого). В этом случае счетчик каждого цикла должен иметь уникальное имя.

```
For I = 1 To 10
For J = 1 To 10
For K = 1 To 10
...
Next K
Next J
Next I
```

По правилу хорошего стиля программирования операторы внутри конструкции For...Next должны писаться с отступом вправо. Это улучшает читаемость программы. Если в программе встречаются вложенные конструкции For...Next или другие конструкции (If...Then...Else If...End If, Select Case...End Select, Do...Loop и другие), то внутри этих конструкции операторы также сдвигаются на несколько пробелов вправо. Далее, в примерах, видно как это делается.

Пример. Вычислить сумму ряда $S = \sum_{k=3}^{8} \frac{k + \sqrt{k}}{2 - \sin k}$

Создадим модуль, который содержит следующую процедуру:

Sub summa()

```
Dim S As Single 'переменная (Сумма)
Dim k As Integer 'счетчик
S = 0 'Начальное значение суммы
For k = 3 To 8
 'Накапливаем сумму
S = S + (k + Sqr(k)) / (2 - Sin(k))
Next k
MsgBox "Сумма ряда = " & S
End Sub
```

В этом примере отсутствует ключевое слово **Step**, т.к. шаг равен 1. Если требуется вычислить произведение ряда, то в рассмотренном примере необходимо начальное значение произведения задать равным 1.

Лабораторная работа №12 Вычисление рядов с помощью циклов.

Варианты заданий

Задача. Составьте программу, которая производит расчет согласно варианту задания.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений и пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

1.	Дано натуральное	число	n.	Вычислите
	$S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + (-1)^n \cdot \frac{1}{2^n}.$			
2.	Дано натуральное	число	<i>N</i> .	Вычислите
	$S = \frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2}$	$\frac{1}{2+\cdots+\sin N}.$		
3.	Дано натуральное число N	. Вычислит	е прои	зведение пер-
	вых N сомножителей			
	$P = \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdots \frac{2N}{2N+1}.$			
4.	Дано натуральное	число	N.	Вычислите
	$\frac{\cos 1}{\cos 1} \cdot \frac{\cos 1 + \cos 2}{\cos 1 + \cos 2 + \dots + \cos 2} \cdots \frac{\cos 1 + \cos 2 + \dots + \cos 2}{\cos 1 + \cos 2 + \dots + \cos 2}$	$+\cos N$		
	$\frac{\sin 1}{\sin 1 + \sin 2} \frac{\sin 1 + \sin 2 + \dots + \sin 2}{\sin 1 + \sin 2 + \dots + \sin 2}$	$-\sin N$		
5.	Дано действительное	число	x.	Вычислите
	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}.$			
6.	Даны натуральное число п	и действит	сельное	е число х. Вы-
	числите			
	$S = \sin x + \sin \sin x + \dots + \underbrace{\sin \sin \dots \sin x}_{n \text{ pas}}$	<u>y</u> .		
7.	Даны действительное числе	о <i>а</i> и натур	альное	е число <i>п</i> . Вы-
	числите			
	$P = a(a+1)\dots(a+n-1).$			
8.	Даны действительное число	о а и натур	альное	число <i>п</i> . Вы-
	числите			
	$P = a(a-n)(a-2n)\dots(a-n^2).$			
9.	Даны действительное числе	о <i>а</i> и натур	альное	е число <i>п</i> . Вы-
	числите			
	$S = \frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2n-2}}.$			
10.	Дано действительное	число	X.	Вычислите
	$\frac{(x-1)(x-3)(x-7)\dots(x-63)}{(x-2)(x-4)(x-8)\dots(x-64)}.$			
11.	Вычислите			

	$(1 + \sin 0, 1)(1 + \sin 0, 2)(1 + \sin 10).$
12.	Даны натуральное число <i>n</i> и действительное число <i>x</i> . Вы-
	числите
	$\sin x + \sin x^2 + \ldots + \sin x^n .$
13.	Дано натуральное число <i>п</i> . Вычислите
	$S = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + \ldots + n \cdot (n+1) \ldots 2n$.
14.	Дано натуральное число <i>п</i> . Вычислите
	$P = \left(1 - \frac{1}{2^2}\right) \left(1 - \frac{1}{3^2}\right) \cdots \left(1 - \frac{1}{n^2}\right), \text{где } n > 2.$
15.	Дано натуральное число <i>п</i> . Вычислите
	$P = \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{4}\right) \left(1 - \frac{1}{6}\right) \cdots \left(1 - \frac{1}{2n}\right).$
16.	Дано натуральное число <i>n</i> . Вычислите
	$S = 1! + 2! + 3! + \ldots + n! (n > 1).$
17.	Дано натуральное число <i>п</i> . Вычислите
	$S = \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots + \frac{1}{(2n+1)^2}.$
18.	Для данного действительного числа х вычислите по схеме
	Горнера
	$y = x^{10} + 2x^9 + 3x^8 + \ldots + 10x + 11.$
19.	Числа Фибоначчи (f _n) определяются формулами
	$f_0 = f_1 = 1, f_n = f_{n-1} + f_{n-2}$ ïðè $n = 2, 3, \dots$
	Определите f_{40} .
20.	Дано натуральное число <i>n</i> . Вычислите
	$y = 1 \cdot 3 \cdot 5 \dots (2n-1).$
21.	Дано натуральное число <i>n</i> . Вычислите $y = 2 \cdot 4 \cdot 6 \dots (2n)$.
22.	Вычислите
	$y = \cos x + \cos x^2 + \cos x^3 + \ldots + \cos x^n .$
23.	Вычислите $y = \sin 1 + \sin 1, 1 + \sin 1, 2 + \dots + \sin 2$.
24.	Даны натуральные числа <i>n</i> и <i>k</i> . Вычислите
	$\sqrt{k+\sqrt{2k+\ldots+\sqrt{k(n-1)}+\sqrt{kn}}}.$
25.	Дано натуральное число <i>n</i> . Вычислите
	$\frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \dots + \frac{n+1}{n}.$
26.	Вычислите
-----	--
	$(1 + \sin 0, 1)(1 + \sin 0, 2)(1 + \sin 10).$
27.	Даны действительное число а и натуральное число n. Вы-
	числите
	$S = \frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2n-2}}.$
28.	Дано натуральное число <i>N</i> . Вычислите
	$S = \frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin N}.$
29.	Дано натуральное число <i>п</i> . Вычислите
	$S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + (-1)^n \cdot \frac{1}{2^n}.$
30.	Дано натуральное число <i>N</i> . Вычислите произведение пер-
	вых N сомножителей
	$P = \frac{2}{2} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdots \frac{2N}{2N+1}$
31	55721V+1 Лано лействительное число r Вычислите
51.	r^3 r^5 r^7 r^9 r^{11} r^{13}
	$x - \frac{x}{3!} + \frac{x}{5!} - \frac{x}{7!} + \frac{x}{9!} - \frac{x}{11!} + \frac{x}{13!}.$
32.	Дано натуральное число <i>п</i> . Вычислите
	$P = \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{4}\right) \left(1 - \frac{1}{6}\right) \cdots \left(1 - \frac{1}{2n}\right).$
33.	Даны натуральное число <i>n</i> и действительное число <i>x</i> . Вы-
	числите
	$S = \sin x + \sin \sin x + \dots + \underbrace{\sin \sin \dots \sin x}_{n \text{ pas}}.$
34.	Дано действительное число <i>х</i> . Вычислите
	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}.$
35.	Числа Фибоначчи (f _n) определяются формулами
	$f_0 = f_1 = 1, f_n = f_{n-1} + f_{n-2}$ ïðè $n = 2, 3, \dots$
	Определите f_{40} .
36.	Дано натуральное число <i>n</i> . Вычислите
	$\frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \dots + \frac{n+1}{n}$.
37.	Даны действительное число а и натуральное число n. Вы-

	числите		
	$P = a(a-n)(a-2n)(a-n^{2}).$		
38.	Дано натуральное число	$\sim N.$	Вычислите
	$\frac{\cos 1}{\sin 1} \cdot \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} \cdots \frac{\cos 1 + \cos 2 + \ldots + \cos N}{\sin 1 + \sin 2 + \ldots + \sin N}.$		
39.	Дано натуральное число <i>n</i> . Вычисл	ите	
	$S = 1! + 2! + 3! + \ldots + n! (n > 1).$		
40.	Дано действительное чис	ло х.	Вычислите
	$\frac{(x-1)(x-3)(x-7)\dots(x-63)}{(x-2)(x-4)(x-8)\dots(x-64)}.$		

Лабораторная работа№ 13 Использование конструкции For...Next. Табулирование функции.

2.6. Работа с таблицами

Существует много задач, результаты которых нужно заносить в таблицу. С помощью VBA очень легко организовать занесение результатов расчета в таблицы.

Основным объектом для работы с таблицами является **Tables**. Этот объект входит в состав объекта **Documents**, т.е. наряду с объектом **Selection**, является одним из основных объектов Word. Формирование таблицы начинается с ее создания. Для этого применяется метод **Add**:

```
Tables.Add(Range, NumRows, NumColumns)
```

где **Range** – область документа, где будет расположена таблица; **NumRows** – количество строк таблицы; **NumColumns** – количество столбцов.

Чтобы было удобно работать с таблицами, нужно задать специфическое имя для таблицы. Это делается для того, чтобы у каждой таблицы было свое имя и не возникало путаницы.

```
'объявляем объект tblNew как таблицу
Dim tblNew As Table
'Создаем таблицу.
Set tblNew = Tables.Add(Selection.Range, 10, 10)
```

В этом примере Selection.Range означает, что таблица будет вставлена в область текущего выделения. У этой таблицы будет 10 строк и 10 столбцов. Если выполнить эти две команды, то в документе появится пустая таблица без границ.

Чтобы сделать границы таблицы видимыми есть несколько способов:

Использовать метод автоформата таблицы:

tblNew.AutoFormat (wdTableFormatList6),

где AutoFormat, это метод применяется к объекту Tables. Он производит автоформатирование таблицы с помощью стиля, указанного в скобках. Для каждого стиля таблицы имеется своя константа, которую можно найти с помощью браузера объектов, вызываемого из VBA с помощью клавиши F2 или через меню View – Object Browser.

Из редактора Word стили автоформатирования таблицы можно увидеть с помощью вкладки Конструктор в разделе Стили таблиц (см. рис.). Как правило, этой вкладки нет на панели инструментов. Она появляется автоматически, как только вы начинаете работать с таблицами то есть она должна быть выделена или в ней должен находиться курсор.

Ссылки	Рассылки	Рецензи	рование	Вид Разр	аботчик	Конструктор	
Простые так	блицы						
Встроенные	e						
					<u></u>		
			<u></u> 				
							•
Измени	ить стиль табл	ицы			N		
Очисти	ть				3		
Создат	ь стиль таблиг	цы					

Рис. 2.11. Рис. Вкладка Стили таблиц.

Другой способ обозначить границы вставляемой таблицы с помощью объекта Borders (границы), и его свойств OutsideLineStyle (стиль внешних линий), InsideLineStyle(стиль внутренних линий), InsideLineWidth (толщина внутренних линий), OutsideLineWidth (толщина внешних линий):

```
With tblNew
   .Borders.OutsideLineStyle = wdLineStyleSingle
   .Borders.InsideLineStyle = wdLineStyleSingle
   .Borders.InsideLineWidth = wdLineWidth100pt
   .Borders.OutsideLineWidth = wdLineWidth100pt
End With
```

Для ввода информации в таблицу используют объект Cell, который представляет собой ячейку таблицы и входит в состав объекта Tables.

```
tblNew.Cell(Row, Column).Range.InsertAfter Информация
```

где параметры Row и Column – адрес ячейки таблицы. Row – номер строки, а Column – номер столбца таблицы. Свойство Range возвращает диапазон указанной ячейки. InsertAfter – метод объекта Range (диапазон), вставляющий информацию в диапазон ячейки.

Как уже говорилось, объект **Range** также как и объект **Selection** является основополагающим объектом Word. Он также может быть главным объектом, но может входить в состав других объектов, как в нашем примере. Объект **Selection** представляет собой какое-либо выделение, над которым работают. А объект **Range** – это какая либо область, обозначенная границами.

Параметр Информация может содержать как текст, так и переменные.

```
tblNew.Cell(1,1).Range.InsertAfter "Первая ячейка"
```

В этом примере в ячейку, находящуюся в первом столбце и первой строке вводится фраза «Первая ячейка». Естественно, что координаты ячейки можно изменять.

Возможности по оформлению текста, таблиц очень велики. И их изучением лучше заниматься самостоятельно, используя справочную систему, браузер объектов и запись макросов.

Задача. Сформировать таблицу Пифагора (умножения). Вывести ее в виде отчета в документ Word.

Для решения данной задачи не нужно проектировать новую форму. Достаточно будет создать новый модуль, поместив туда процедуру.

```
Sub NewTable()
    Dim docNew As Document 'объявляем новый документ
    Dim tblNew As Table 'объявляем новую таблицу
     'Объявляем счетчики циклов.
    Dim X As Integer
    Dim Y As Integer
    создаем новый документ и устанавливаем на него ссылку
    Set docNew = Documents.Add
    устанавливаем параметры страницы
    With Selection.PageSetup
      устанавливаем стандартный лист формата А4
      .PageWidth = CentimetersToPoints(21)
      .PageHeight = CentimetersToPoints(29.7)
      .LeftMargin = CentimetersToPoints(2) 'левое поле (1,2 см)
      .RightMargin = CentimetersToPoints(1) 'правое поле (1 см)
      .TopMargin = CentimetersToPoints(1) 'верхнее поле (1 см)
      .BottomMargin = CentimetersToPoints(1.7) 'нижнее поле (1 см)
      .Orientation = wdOrientPortrait 'портретная ориентация
    End With
```

```
With Selection
  'Печатаем заголовок
  .InsertAfter "Таблица умножения"
  'напечатанный заголовок форматируем по середине
  .ParagraphFormat.Alignment = wdAlignParagraphCenter
  устанавливаем для заголовка полужирный шрифт, 14 размера
  .Font.Bold = True
  .Font.Size = 14
  'вставляем два параграфа
  .InsertParagraphAfter
  .InsertParagraphAfter
  'Перемещаем курсор в конец документа
  Selection.EndOf Unit:=wdSection
End With
создаем таблицу, состоящую из 11 строк и 11 столбцов
Set tblNew = docNew.Tables.Add(Selection.Range, 11, 11)
With tblNew
  'Заполняем первую строку и столбец параметрами умножения
  For X = 1 To 10
  числа заполняются со второй строки и второго столбца
    .Cell(X + 1, 1).Range.InsertAfter X
    .Cell(1, X + 1).Range.InsertAfter X
  Next X
  'Заполняем оставшуюся часть таблицы результатами умножения
  For X = 2 To 11
      For Y = 2 To 11
          .Cell(X, Y).Range.InsertAfter (X - 1) * (Y - 1)
     Next Y
  Next X
  'используем стиль таблицы для ее оформления
  .AutoFormat (wdTableFormatList8)
  Осуществляем автоподбор ширины столбцов по содержимому
  .Columns.AutoFit
  'Помещаем таблицу по центру листа
  .Rows.Alignment = wdAlignRowCenter
End With
```

```
End Sub
```

Задача. Составить программу для вычисления значений функции $F(x) = \frac{x^3}{x+1} \sin x$, на отрезке [a,b]с шагом *h*. Результат представить в виде таблицы в документе Word, первый столбец которой содержит значения аргумента, второй — соответствующие значения функции. Программе должна соответствовать форма с элементами управления для ввода данных. В результате работы программы составляется отчет, который должен содержать задание и таблицу расчета функции.

Решение. Подобные задачи называются табулированием функции и применяются для построения графиков функций.

1. Создадим форму рис. 7. Значение свойств объектов формы даны в таблице 2.



Рис. 7. Форма для расчета табуляции функции.

Таблица №2

Таблица значений свойств программы «Табулирование функции»

Объект (Элемент	Свойство	Значение
управления)		
UserForm1	Name	frmTabulation
	Caption	«Табулирование функции»
Label1	Caption	«Протабулировать функ- цию на отрезке [a,b] с ша- гом h»
Image1	Picture	Указать путь к рисунку
Label2	Caption	«Начальное значение а =»

Label3	Caption	«Конечное значение b =»
Label4	Caption	«Шаг h =»
ToytDoy1	Name	txtA
TEXIDUXI	Tabindex	1
Toxt Dox 2	Name	txtB
TEXIDOX2	Tabindex	2
ToxtDox2	Name	txtH
TEXIDOXS	Tabindex	3
CommandButton1	Name	cmdStart
	Caption	«Рассчитать»
	Tabindex	4
CommandButton2	Name	cmdExit
	Caption	«Выход»
	Tabindex	5

2. Для кнопки Рассчитать (cmdStart) создадим код:

```
Private Sub cmdStart Click()
    Dim docNew As Document 'объявляем новый документ
    Dim tblNew As Table 'объявляем новую таблицу
    Dim a As Single, b As Single, h As Single, Pi As Single 'Объявляем переменные
    a = CSng(txtA.Text)
   b = CSng(txtB.Text)
   h = CSng(txtH.Text)
    Pi = 3.14159265358979
    frmTabulation.Hide
    'создаем новый документ
    Set docNew = Documents.Add
    With Selection
        'Печатаем заголовок
        .InsertAfter "Использование цикла For...Next. Табулирование функции"
        'напечатанный заголовок форматируем по середине
        .ParagraphFormat.Alignment = wdAlignParagraphCenter
        'создаем отступ для первой строки
        .ParagraphFormat.FirstLineIndent = CentimetersToPoints(0.7)
        устанавливаем для заголовка полужирный шрифт, 14 размера
        .Font.Bold = True
        .Font.Size = 14
        'вставляем два параграфа
        .InsertParagraphAfter
        .InsertParagraphAfter
        'Перемещаем курсор в конец документа
        .EndOf Unit:=wdSection
        'Печатаем задание
        .InsertAfter "Задание. "
        .ParagraphFormat.Alignment = wdAlignParagraphJustify
        .Font.Bold = True
        .EndOf Unit:=wdSection
        .Insertlfter "Составить программу для вычисления значений функции "
        .Font.Bold = False
        .EndOf Unit:=wdSection
```

```
'Создаем код формулы
        Formula = "f(x)=x^3/(x+1) \cdot \sin x"
        Вызываем процедуру создания формулы
        Create_formul
        .Font.Bold = False
        .InsertAfter " на отрезке [a, b] с шагом h. Результат представить в виде таблицы "
        « в документе Word, первый столбец которой – значения аргумента, второй – соответствующие "
        ٤ "значения функции. Программа должна содержать форму с элементами управления для ввода данных. "_____
        🕻 "Программа должна составить отчет, который должен содержать задание, таблицу расчета функции."
          .Font.Bold = False
        .InsertParagraphAfter
        .InsertAfter "Решение. "
        .EndOf Unit:=wdSection
        .InsertParagraphAfter
        'создаем таблицу, состоящую из 1 строки и 2 столбцов
        Set tblNew = docNew.Tables.Add(Selection.Range, 1, 2)
        'заполняем первую строку программы
        With tblNew
            .Cell(1, 1).Range.InsertAfter "x"
            .Cell(1, 2).Range.InsertAfter ("F(x)")
            For x = a To b Step h
                Selection.InsertRowsBelow 'вставляем новую строку в конец таблицы
                If x = -1 Then 'Если знаменатель формулы равен нулю
                    .Cell(.Rows.Count, 1).Range.InsertAfter x
                    .Cell(.Rows.Count, 2).Range.InsertAfter "Значение не определено"
                Else
                    'рассчитываем и заполняем таблицу
                    Fx = x^{3} * Sin(x * Pi / 180) / (x + 1)
                    .Cell(.Rows.Count, 1).Range.InsertAfter x
                    .Cell(.Rows.Count, 2).Range.InsertAfter Fx
                End If
            Next x
            'используем стиль таблицы для ее оформления
            .AutoFormat (wdTableFormatClassic1)
            Осуществляем автоподбор ширины столбцов по содержимому
            .Columns.AutoFit
            'Помещаем таблицу по центру листа
            .Rows.Alignment = wdAlignRowCenter
        End With
        'Показываем форму
        frmTabulation.Show
    End With
End Sub
```

3. Отдельно создадим процедуру для вставки формулы в документ. Данная процедура будет вызываться из основной процедуры по её названию Create formul.

```
Sub Create formul()
    Устанавливаем объект objRange, как выделенную часть документа. В нашем случае место под курсором
   Set objRange = Selection.Range
    'свойству text объекта objRange присваиваем код формулы
   objRange.Text = Formula
    'преобразуем объект objRange в математическую формулу
    Set objRange = Selection.OMaths.Add(objRange)
    ' создаем объект objEq, образуя его из объекта objRange
    Set objEq = objRange.OMaths(1)
    'Преобразуем формулу из линейного вида к профессиональному
    objEq.BuildUp
    With Selection
        'Так после создания формулы курсор оказывается перед формулой, его необходимо переместить за формулу
        'Для это выполняем следующие действия:
        'Пермещаем объект Selection по конца формулы (формула оказывается выделенной)
        .MoveEnd Unit:=wdLine, Count:=1
        'Снимаем выделение с формулы (курсор оказывается внутри формулы, в её конце)
        .Collapse Direction:=wdCollapseEnd
        'Перемещаем курсор на один символ вправо (курсор оказывается за формулой)
        .MoveRight Unit:=wdCharacter, Count:=1
        'Вставляем новый параграф
        .InsertParagraphAfter
        'Снимаем выделение (курсор оказывается на новой строке)
        .Collapse Direction:=wdCollapseEnd
    End With
End Sub
```

4. Для кнопки Выход (cmdExit) создадим код:

Private Sub cmdExit_Click() End End Sub

5. Вне всех процедур объявим переменную Formula:

Dim Formula As String

Лабораторная работа №13 Использование конструкции For...Next. Табулирование функции.

Задача.

Составьте программу для вычисления значений функции F(x), на отрезке [a,b] с шагом h.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, выхода из программы.

Сформировать отчет средствами VBA (обязательно!). Отчет должен содержать: текст задачи; формулу по которой рассчитывается таблица. Результаты расчёта представьте в виде таблицы в документе Word, первый столбец которой содержит значения аргумента, второй — соответствующие им значения функции.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

$1. F(x) = x - \sin x.$	$2. F(x) = \sin^2 x.$
$3. F(x) = 2\cos x - 1$	$4. F(x) = \operatorname{tg} x.$
$5. F(x) = \operatorname{ctg} x + 1.$	$6. F(x) = \sin x - \cos x.$
$7. F(x) = x \sin x.$	$F(x) = \sin\left(\frac{1}{x}\right) + 2.$ 8.
$F(x) = x \cos\left(\frac{1}{x}\right) + 2.$	10. $2\sin^2 x + 1$
11. $F(x) = \sqrt{x} \cos^2 x.$	12. $F(x) = \sin x + \operatorname{tg} x.$
13. $F(x) = \cos x + \operatorname{ctg} x.$	14. $F(x) = 2 \operatorname{tg} \frac{x}{2} + 1.$
$F(x) = \operatorname{tg} \frac{x}{2} + 2\cos x.$	16. $F(x) = \operatorname{ctg} \frac{x}{3} + \frac{1}{2} \sin x$.
17. $F(x) = \frac{1}{2}\sin\frac{x}{4} + 1.$	18. $F(x) = 2\cos\sqrt{x} + 0.5$.
19. $F(x) = x^2 \sin^2 x + 1.$	$F(x) = \frac{1}{2} \operatorname{ctg} \frac{x}{4} + 4.$
$21. \qquad F(x) = \sin^2 x - \cos 2x.$	22. $F(x) = 7\sin^2 x - \frac{1}{2}\cos x.$
$23. \qquad F(x) = -\cos 2x$	24. $F(x) = \operatorname{tg} 2x - 3$.
$25. \qquad F(x) = \sin x + 0.5 \cos x.$	$F(x) = \frac{x}{\cos x}.$
$F(x) = \operatorname{tg} \frac{x}{2} + 2\cos x.$	$F(x) = \sin\left(\frac{1}{x}\right) + 2.$
29. $F(x) = 7\sin^2 x - \frac{1}{2}\cos x$.	$30. \qquad F(x) = 2\cos x - 1$

31.	$F(x) = 2\cos\sqrt{x} + 0.5.$	32.	$F(x) = \operatorname{ctg} \frac{x}{3} + \frac{1}{2}\sin x.$
33.	$F(x) = \sqrt{x} \cos^2 x.$	34.	$F(x) = x \sin x .$
35.	$F(x) = x \cos\left(\frac{1}{x}\right) + 2.$	36.	$F(x) = \sin x + 0.5 \cos x.$
37.	$F(x) = \operatorname{tg} x$.	38.	$F(x) = \frac{x}{\cos x}.$
39.	$F(x) = \operatorname{tg} \frac{x}{2} + 2\cos x.$	40.	$2\sin^2 x + 1$

Лабораторная работа №14 Циклы с проверкой условия (Do…Loop)

Назначение данного вида циклов - повторение набора операторов, при неизвестном количестве повторов (итераций). В этом случае для завершения работы цикла используется проверка некоторого условия до выполнения операторов внутри цикла или после выполнения операторов внутри цикла.

Существует 4 вида синтаксиса данной конструкции:

Синтаксис 1.

```
Do While условие
[операторы]
[Exit Do]
Loop
```

где **Do** – оператор начинающий циклическую конструкцию.

While – ключевое слово, говорящее о том, что операторы внутри цикла должны выполняться до тех пор, пока выполняется условие, т.е. пока условие истинно (*True*).

Условие (необязательный) - числовое или строковое выражение, которое имеет значение *True* или *False*. Если условие имеет значение *Null*, то аргумент условия рассматривается как значение *False*. Более подробно этот параметр уже был описан при изучении конструкции **If...Then...Elself...EndIf**.

Exit Do (необязательный) - оператор, обеспечивающая альтернативную возможность выхода из цикла Do...Loop. Часто используемый вместе с определением некоторого условия (например, If...Then), оператор Exit Do передает управление оператору, непосредственно следующему за оператором Loop. Во вложенных циклах Do...Loop оператор Exit Do передает управление циклу охватывающего уровня по отношению к циклу, в котором она вызывается.

Loop – оператор, завершающий циклическую конструкцию.

В этом синтаксисе условие проверяется до начала выполнения операций внутри цикла. Цикл будет выполняться до тех пор, пока условие выхода из цикла – истинно (*True*). Когда параметр условие становится ложным (*False*), произойдет выход из цикла. Управление перейдет оператору, непосредственно следующему за оператором Loop. Операторы внутри цикла могут не выполняться, если параметр условие изначально является ложным (*False*).

Синтаксис 2.

```
Do Until условие
[операторы]
[Exit Do]
Loop
```

где Until – ключевое слово, говорящее о том, что операторы внутри цикла должны выполняться до тех пор, пока условие не выполняется, т.е. пока условие является ложным (*False*).

Остальные параметры этого синтаксиса рассмотрены выше.

В этом синтаксисе, так же как и в синтаксисе 1, условие проверяется до начала выполнения операторов внутри цикла. Операторы внутри цикла будут выполняться до тех пор, пока условие выхода из цикла не станет истинным. Когда условие становится истинным (*True*), произойдет выход из цикла. Управление перейдет оператору, непосредственно следующему за оператором Loop.

Операторы внутри цикла могут не выполняться, если параметр **условие** изначально является ложным (*False*)..

Синтаксис 3.

```
Do
[операторы]
[Exit Do]
Loop While условие
```

Все параметры данного синтаксиса рассмотрены выше.

В этом синтаксисе условие проверяется после выполнения операторов внутри цикла. Так же, как и в синтаксисе 1, операторы внутри цикла будут выполняться до тех пор, пока условие выхода из цикла – истинно (*True*). Когда параметр условие становится ложным (*False*), произойдет выход из цикла. Управление перейдет оператору, непосредственно следующему за оператором Loop. В отличие от синтаксиса 1, операторы внутри цикла выполняются хотя бы один раз, независимо от условия выхода из цикла.

Синтаксис 4.

```
Do
[операторы]
[Exit Do]
Loop Until условие
```

Все параметры данного синтаксиса также рассмотрены выше.

В этом синтаксисе условие проверяется после выполнения оператора внутри цикла. Так же, как и в синтаксисе 2, операторы внутри цикла будут выполняться до тех пор, пока условие выхода из цикла не станет истинным. Когда условие становится истинным (*True*), произойдет выход из цикла. Управление перейдет оператору, непосредственно следующему за оператором **Loop**.

В отличие от синтаксиса 2, операторы внутри цикла выполняются хотя бы один раз, независимо от условия выхода из цикла.

Резюмируя, можно сказать, что в конструкции **Do...Loop** условие проверяется либо в начале, либо в конце конструкции. Операторы внутри конструкции выполняются либо когда условие истинно (при использовании ключевого слова **While**), либо они выполняются, когда условие ложно (при использовании ключевого слова **Until**).

В рассматриваемой далее процедуре ChkFirstWhile условие проверяется до входа в цикл с использованием ключевого слова While (применяется синтаксис 1).

```
Sub ChkFirstWhile()
```

```
counter = 0
myNum = 20
Do While myNum > 10
myNum = myNum - 1
counter = counter + 1
Loop
MsgBox "Выполнено " & counter & " итераций цикла."
End Sub
```

При выполнении программы цикл будет повторяться до тех пор, пока переменная **myNum** больше 10. Значение этой переменной уменьшается внутри цикла на единицу (myNum = myNum – 1). Переменная **counter** предназначена для подсчета количества итераций цикла. С каждым проходом цикла эта переменная увеличивается на единицу (counter = counter + 1). Когда переменная **myNum** станет меньше 10 (myNum = 9), произойдет выход из цикла. Оператор **MsgBox** выдаст сообщение о количестве итераций цикла рис. 8



Рис. 8

Если начальное значение myNum задать равным 9 вместо 20, операторы внутри цикла выполняться не будут. В окне **MsgBox** в этом случае будет напечатано «Выполнено 0 итераций цикла»

В следующей процедуре ChkFirstUntil условие проверяется также как и в предыдущем примере до входа в цикл, но с использованием ключевого слова Until (применяется синтаксис 2).

```
Sub ChkFirstUntil()
counter = 0
myNum = 20
Do Until myNum = 10
myNum = myNum - 1
counter = counter + 1
Loop
MsgBox "В цикле выполнено " & counter & " итераций."
End Sub
```

Цикл будет повторяться до тех пор, пока переменная **myNum** не станет равной 10. Значение этой переменной уменьшается внутри цикла на единицу (myNum = myNum – 1). Переменная **counter** предназначена для подсчета количества итераций цикла. С каждым проходом цикла эта переменная увеличивается на единицу (counter = counter + 1). Когда переменная **myNum** станет равной 10, произойдет выход из цикла. Оператор **MsgBox** выдаст сообщение о количестве итераций цикла: «Выполнено 10 итераций цикла».

Для демонстрации работы синтаксиса 3 предназначена следующей процедуре ChkLastWhile, в которой условие проверяется в конце цикла с использованием ключевого слова While.

```
Sub ChkLastWhile()
counter = 0
myNum = 9
Do
myNum = myNum - 1
counter = counter + 1
Loop While myNum > 10
MsgBox "В цикле выполнено " & counter & " итераций."
End Sub
```

Цикл должен повторяться до тех пор, пока переменная **myNum** больше 10. Однако при начальном значении 9, эта переменная станет равна 8 (myNum = myNum – 1). Следовательно, в процедуре оператора внутри цикла выполняются только один раз до того как условие примет значение *False*. Оператор **MsgBox** выдаст сообщение о количестве итераций цикла: «В цикле выполнено 1 итераций».

И, наконец, в следующей процедуре ChkLastUntil условие проверяется в конце цикла с использованием ключевого слова Until (применяется синтаксис 4).

```
Sub ChkLastUntil()
  counter = 0
  myNum = 1
  Do
```

```
myNum = myNum + 1
counter = counter + 1
Loop Until myNum = 10
MsgBox "В цикле выполнено " & counter & " итераций."
End Sub
```

Цикл будет повторяться до тех пор, пока переменная **myNum** не станет равной 10. Значение этой переменной увеличивается внутри цикла на единицу (myNum = myNum + 1). Переменная **counter** предназначена для подсчета количества итераций цикла. С каждым проходом цикла эта переменная увеличивается на единицу (counter = counter + 1). Когда **myNum** станет равной 10, произойдет выход из цикла. Оператор **MsgBox** выдаст сообщение о количестве итераций цикла: «Выполнено 9 итераций цикла».

Замечание. Нужно быть внимательным в выборе параметра условие. При неправильном выборе этого параметра, можно получить при выполнении программы так называемый бесконечный цикл. Рассмотрим пример

```
Sub ChkLastWhile()
counter = 0
myNum = 9
Do
myNum = myNum - 1
counter = counter + 1
Loop While myNum = 10
MsgBox "В цикле выполнено " & counter & " итераций."
End Sub
```

В этом примере переменная **myNum** изначально имеет значение 9. Внутри цикла это значение уменьшается. Однако при проверке условия ожидается, что переменная **myNum** станет равной 10. Очевидно, что этого не может произойти. Следовательно, программа не выйдет из цикла. Это выглядит, как «зависшая» программа, которая не реагирует на внешние воздействия. Если эта ошибка произошла во время отладки программы, то прервать бесконечный цикл можно с помощью комбинации клавиш **Ctrl+Break**. По правилу хорошего стиля программирования операторы внутри конструкции Do...Loop должны писаться с отступом вправо. Это улучшает читаемость программы. Если в программе встречаются вложенные конструкции Do...Loop или другие конструкции (If...Then...Else If...End If, Select Case...End Select, For...Next и другие), то внутри этих конструкции операторы также сдвигаются на несколько пробелов вправо. В рассматриваемых далее примерах эти правила учтены.

Пример. Составьте программу табулирования (вычисления значений) функции

$$f(x) = \begin{cases} \sqrt[3]{|e^{x} - a|}, & \text{при } x < 2a, \\ a + \sin x, & \text{при } x > 2a, \\ nponyctute, npu & x = 2a, \end{cases} a = 0,5, \ 0,1 \le x \le 2, \ \Delta x = 0,3. \end{cases}$$

В результате выполнения программы создается отчет, в котором будет записываться таблица расчета функции.

Решение

1. Создайте в редакторе Word новый документ.

2. Перейдите в редактор VBA.

3. В проект соответствующий созданному документу вставьте новый модуль (меню Insert – Module).

4. Создайте в новом модуле процедуру:

```
Sub UsingDoLoop()
Dim x As Single, f As Single
Const a = 0.5
Set docNew = Documents.Add 'создаем новый документ
'создаем таблицу, состоящую из 1 строки и 2 столбцов
Set tblNew = docNew.Tables.Add(Selection.Range, 1, 2)
'заполняем первую строку программы
With tblNew
.Cell(1, 1).Range.InsertAfter "x"
.Cell(1, 2).Range.InsertAfter ("F(x)")
x = 0.1 'Начальное значение x
```

```
'Цикл по переменной х
         Do Until x > 2 'Проверка условия выхода из цикла
           If x \ll 2 * a Then
             If x < 2 * 2 Then
               f = Abs(Exp(x) - a) ^ (1 / 3)
             Else
               f = a + Sin(x)
             End If
             Selection.InsertRowsBelow 'вставляем новую строку в конец
таблицы
             'заполняется ячейка х
             .Cell(.Rows.Count, 1).Range.InsertAfter x
             'заполняется ячейка F(x)
             .Cell(.Rows.Count, 2).Range.InsertAfter f
           End If
           х = х + 0.3 'Переход к следующему значению х
         Loop 'Конец цикла, переход на Do
       End With
   End Sub
```

Пример. Цикл **Do . . . Loop** с постусловием. Вычислите сумму ряда

 $S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + (-1)^{i+1} \frac{1}{i}$, с точностью $\xi = 0.001$.

Решение

1. Создайте в редакторе Word новый документ.

2. Перейдите в редактор VBA.

3. В проект соответствующий созданному документу вставьте новый модуль (меню Insert – Module).

4. Создайте в новом модуле процедуру:

```
Sub UsingDoLoopEnd()

Dim i As Integer, S As Single

Const epsilon = 0.001

S = 1 'Начальное значение суммы ряда

i = 1 'Начальное значение аргумента

Do 'Начало цикла

i = i + 1 'Следующее значение аргумента

S = S + (-1) ^ (i + 1) / i 'Вычисление суммы

Loop While 1 / i > epsilon 'Точность достигнута?

MsgBox "Сумма ряда = " & S 'Печать результата

End Sub
```

Лабораторная работа №14 Циклы с проверкой условия (Do…Loop)

Задача. Составьте программу для выполнения предложенного вам задания. Программа должна:

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: текст задачи; формулу по которой рассчитывается таблица, конечный результат. Результаты расчёта представьте в виде таблицы в документе Word, первый столбец которой содержит значения шага(1, 2, 3 и т.д.) аргумента, второй – величина с которой происходит сравнение (п, очередное слагаемое, третий – текущий результат).

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

1.	Вычисляйте произведение последовательности натуральных
	чисел <i>i</i> =1, 2, 3 до тех пор, пока очередной сомножитель
	не превышает заданного числа <i>n</i> . Выведите полученное произ-
	ведение и последний сомножитель.
2.	Дано положительное число <i>a</i> <1. Вычисляйте и выводите числа
	вида $\frac{1}{2^n}$, пока они остаются больше <i>a</i> (при этом <i>n</i> >1).

3.	Дано действительное число х. Вычислите приближенное зна-
	чение бесконечной суммы: $x + \frac{x^2}{2} + \frac{x^3}{3} +(x < 1).$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше 0,0001.
4.	Дано действительное число х. Вычислите приближенное зна-
	чение бесконечной суммы: $x - \frac{x^2}{2} + \frac{x^3}{3}(x < 1).$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше 0,0001.
5.	Дано действительное число х. Вычислите приближенное зна-
	чение бесконечной суммы: $x + \frac{x^3}{3} + \frac{x^5}{5} + (x < 1).$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше 0,0001.
6.	Дано действительное число х. Вычислите приближенное зна-
	чение бесконечной суммы: $\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots$ (x > 1).
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше 0,0001.
7.	Найдите и выведите на дисплей значения степеней числа 3: 3 <i>n</i> ;
	<i>n</i> =1,2,3Вычисления продолжайте, пока 3 <i>n</i> не превыша-
	ет заданное число d. Вычисления проводите, не используя опе-
	рацию возведения в степень.
8.	Вычислите сумму последовательности натуральных чисел
	<i>i</i> = 1,2,3 Вычисления продолжайте до тех пор, пока сумма не
	превышает заданное число <i>n</i> . Определите, сколько чисел во-
	шло в сумму.
9.	Вычислите произведение последовательности натуральных чи-
	сел <i>i</i> = 1,2,3 Вычисления продолжайте до тех пор, пока сумма

	не превышает заданное число <i>n</i> . Определите, сколько чисел
	вошло в произведение.
10.	Вычислите сумму квадратов последовательности натуральных
	чисел до тех пор, пока значение суммы не превышает заданное
	число <i>n</i> .
11.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее точное значение, с которым можно срав-
	нить полученный ответ): $_{1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\dots}$ $\left(\frac{\pi}{6}\right)$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
12.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее значение, с которым можно сравнить по-
	лученный ответ): $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$ (0,6931478)
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
13.	Вычислите приближенное значение бесконечной суммы (спра-
	ва, в скобках дано ее точное значение, с которым можно срав-
	нить полученный ответ): $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$ $\left(\frac{\pi}{4}\right)$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
14.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее точное значение, с которым можно срав-
	нить полученный ответ): $\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \dots$ (1)
	Нужное приближение будем считать полученным. если вычис-

	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
15.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее точное значение, с которым можно срав-
	нить полученный ответ): $\frac{1}{1\cdot 3} + \frac{1}{2\cdot 4} + \frac{1}{3\cdot 5} + \dots$
	$\left(\frac{3}{4}\right)$ Нужное приближение будем считать полученным, если
	вычислена сумма нескольких первых слагаемых и очередное
	слагаемое оказалось по модулю меньше данного положитель-
	ного числа є.
16.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее точное значение, с которым можно срав-
	нить полученный ответ). $\frac{1}{1\cdot 2\cdot 3} + \frac{1}{2\cdot 3\cdot 4} - \frac{1}{3\cdot 4\cdot 5} + \dots$
	$\begin{pmatrix} 1 \end{pmatrix}$
	$\left(\frac{1}{4}\right)$
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
17.	Находите и выводите на дисплей значения степеней числа 2:
	2^{n} ; $n = 1, 2, 3$ Вычисления продолжайте до тех пор,
	пока 2 ^{<i>n</i>} не превысит заданное число <i>d</i> . Вычисления прово-
	дите, не используя операцию возведения в степень.
18.	Находите и выводите на дисплей значения степеней числа 2:
	2^{n} ; $n = 1, 2, 3$ Вычисления продолжайте до тех пор,
	пока сумма найденных чисел не превысит заданное число d.
	Вычисления проводите, не используя операцию возведения в
	степень.
19.	Вычисляйте сумму кубов последовательности натуральных чи-
	сел до тех пор, пока очередное слагаемое не превысит заданное
	число <i>n</i> .

20.	Вычисляйте произведение последовательности натуральных
	чисел <i>i</i> = 1,2,3 до тех пор, пока произведение не превысит
	заданного числа <i>n</i> . Выведите полученное произведение и по-
	следний сомножитель.
21.	Вычисляйте и выводите члены арифметической прогрессии,
	заданные рекуррентной формулой $a_{i+1} = a_i + 2$ до тех пор, пока оче-
	редной выводимый член прогрессии остается меньше заданно-
	го числа b.
22.	Вычисляйте и выводите члены арифметической прогрессии,
	заданные рекуррентной формулой $a_{i+1} = a_i + 5$ до тех пор, пока оче-
	редной выводимый член прогрессии остается меньше заданно-
	го числа <i>b</i> .
23.	Вычисляйте и выводите значения функции $y = \frac{a^n}{n}$ $n = 1, 2, 3,$
	<i>а</i> <1 до тех пор, пока очередное вычисленное значение остается
	больше заданного положительного числа <i>b</i> .
24.	Сформируйте последовательность, элементы которой вычис-
	ляются по формуле $y = \frac{n}{n+1}$; $n = 1, 2, 3$ Вычисления продолжайте
	до тех пор, пока $(1-y) \ge \varepsilon$.
25.	Вводите числа и суммируйте их до тех пор, пока не будет вве-
	дено число, большее чем первое по абсолютной величине.
26.	Вводите числа и находите их произведение до тех пор, пока не
	будет введено число, делящееся на 3 без остатка.
27.	Не используя операцию возведения в степень, вычисляйте и
	выводите члены последовательности, определяемые выраже-
	нием _{(-1)^{<i>i</i>}<i>i</i>²;<i>i</i>=1,2,3} .Вычисления проводите до тех пор, пока абсо-
	лютное значение очередного члена последовательности не ста-
	нет меньше заданного числа <i>b</i> .
28.	Вводите числа и находите их произведение до тех пор, пока не
	будет введено число, меньше первого по абсолютной вели-
•	чине.
29.	Вычисляйте произведение кубов последовательности нату-

	ральных чисел до тех пор, пока очередной сомножитель не							
	превысит заданное число <i>n</i> .							
30.	Начав тренировки, спортсмен в первый день пробежал 10							
	км.Каждую следующую тренировку он увеличивал норму на							
	10% от предыдущей. Вычислите через сколько тренировок							
	спортсмен будет пробегать за тренировку больше 20 км.							
31.	Дано действительное число х. Вычислите приближенное зна-							
	чение бесконечной суммы: $\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots$ (x > 1).							
	Нужное приближение будем считать полученным, если вычис-							
	лена сумма нескольких первых слагаемых и очередное слагае-							
	мое оказалось по модулю меньше 0,0001.							
32.	Вычисляйте и выводите члены арифметической прогрессии,							
	заданные рекуррентной формулой $a_{i+1} = a_i + 2$ до тех пор, пока оче-							
	редной выводимый член прогрессии остается меньше заданно-							
	го числа b.							
33.	Находите и выводите на дисплей значения степеней числа 2:							
	2^{n} ; $n = 1, 2, 3$ Вычисления продолжайте до тех пор,							
	пока 2 ^{<i>n</i>} не превысит заданное число <i>d</i> . Вычисления прово-							
	дите, не используя операцию возведения в степень.							
34.	Сформируйте последовательность, элементы которой вычис-							
	ляются по формуле _{$y=\frac{n}{n+1}$} ; $n=1,2,3$ Вычисления продолжайте							
	до тех пор, пока $(1-y) \ge \varepsilon$.							
35.	Вычислите приближенное значение бесконечной суммы (спра-							
	ва в скобках дано ее точное значение, с которым можно срав-							
	нить полушенный отрет): $\frac{1}{1,2} + \frac{1}{2,3} + \frac{1}{3,4} + \dots$ (1)							
	Нитв полученный ответ). 12 23 334 (1) Нужное приближение булем снитать полученным если вынис-							
	пена сумма нескольких первых спагаемых и онередное слагае-							
	ное оказалось по модулю меньше данного положительного							
26								
30.	Б ЫЧИСЛЯИТЕ И ВЫВОДИТЕ ЗНАЧЕНИЯ ФУНКЦИИ $y = \frac{a^n}{n}$ $n = 1, 2, 3,$							

	<i>а</i> <1 до тех пор, пока очередное вычисленное значение остается
	больше заданного положительного числа <i>b</i> .
37.	Находите и выводите на дисплей значения степеней числа 2:
	2^{n} ; $n = 1, 2, 3$ Вычисления продолжайте до тех пор,
	пока 2 ⁿ не превысит заданное число <i>d</i> . Вычисления прово-
	дите, не используя операцию возведения в степень.
38.	Вычислите приближенное значение бесконечной суммы (спра-
	ва в скобках дано ее точное значение, с которым можно срав-
	$\frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1}$
	нить полученный ответ): $1 \cdot 2^{-2} \cdot 3^{-3} \cdot 4^{-111}$ (1)
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.
39.	Дано положительное число <i>a</i> <1. Вычисляйте и выводите числа
	вида $\frac{1}{2^n}$, пока они остаются больше <i>а</i> (при этом <i>n</i> >1).
40.	Вычислите приближенное значение бесконечной суммы (спра-
	ва, в скобках дано ее точное значение, с которым можно срав-
	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
	нить полученный ответ): $\frac{1-3}{3}+\frac{5}{5}-\frac{7}{7}+\cdots$ (4)
	Нужное приближение будем считать полученным, если вычис-
	лена сумма нескольких первых слагаемых и очередное слагае-
	мое оказалось по модулю меньше данного положительного
	числа є.

Лабораторная работа№15 Использование циклов For...Next, Do...Loop

Задача. Составьте программу для вычисления значений функции *у* при изменении *x* от x_{μ} до x_{κ} с шагом Δx в соответствии с условием варианта из табл. 3.

Сформировать отчет средствами VBA (обязательно!). Отчет должен содержать: текст задачи; формулу по которой рассчитывается таблица. Результаты расчёта представьте в виде таблицы в документе Word, первый столбец которой содержит значения аргумента, второй — соответствующие им значения функции.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Таблица 3

№	Вид функции	Исходные данные				
п/п	при условии					
		а	b	$X_{ m H}$	$X_{ m K}$	Δx
1		2	0,4	-1,72	2,88	0,35
	$ \begin{cases} e^{x-a} - \ln b-x & \text{, если } x > a \end{cases} $					
	$y = \begin{cases} \sum_{i=3}^{8} \frac{x - \sqrt{i}}{i} & \text{, если } b < x < a \end{cases}$					
	$((x-a)^2 + tg^3(b-x))$, если $x < -0.5$					
2	$\begin{cases} \sqrt[3]{a+\ln x^2} & \text{, если } a < x < b \end{cases}$	1,7	6,2	9,2	0,4	-0,8
	$y = \left\{ \sin^2(a-x) + \sqrt[4]{ a-b }, \text{если } x > b \right\}$					
	$\prod_{k=4}^{8} \frac{10 + \sin x}{k - x} , \text{если } x < 1,2$					

Варианты задания

3	$y = \begin{cases} \prod_{j=2}^{7} \frac{x + \sin j}{j - 1} & \text{, если } x \le a \\ \ln \left a^2 - bx \right & \text{, если } x > 6,5 \end{cases}$	0,2	4,8	7,3	-0,45	-0,9
	$\frac{e^{a-x}}{1,78\cos^2(b^2 - ax)}, $ если $a < x < b$					
4	$\left\{ \frac{\operatorname{tg}(x+a)^2}{e^{x-a}} \right\}, \text{если } b \le x < a$	2,7	1,5	0,2	6,2	0,6
	$y = \left\{ \sum_{i=4}^{12} \left(xi - \sqrt{i+a} \right), \text{если } x \ge a \right.$					
	$\left \ln^2 ax - b^2 \right , если x < 1$					
5	$\left\{\frac{\sqrt[3]{e^{ax}-b}}{\cos^2(a+bx)}, \text{если } x < a\right\}$	1,25	6,3	0,4	10,3	0,8
	$y = \left\{ \ln \left x^2 - a^2 \right , \text{если } a \le x < b \right. \right.$					
	$\left(\prod_{n=3}^{n} \frac{n-a}{x}, ecлu \ x > 9,1\right)$					
6	$ \begin{cases} \sum_{c=4}^{9} \frac{\sqrt{x+c}}{c} & \text{, если } x < b \end{cases} $	4,6	2,4	1,2	6,7	0,9
	$y = \begin{cases} \sqrt{ \sin(a - bx) }, \text{ если } 3.2 < x < a \\ \frac{\sqrt[3]{x} - \sqrt[4]{a}}{1,8 \ln ax - b }, \text{ если } x \ge a \end{cases}$					
7	$\begin{bmatrix} \mathbf{n}^3 \mathbf{x}^2 - a^2 b \end{bmatrix} \text{ equal } a \leq \mathbf{x} \leq b$	2,4	6,2	0,31	9,4	0,8
	$y = \begin{cases} \mathbf{n} ^{ \mathbf{x} } & \mathbf{x} = 0 \\ \mathbf{x} & \mathbf{x} = 1 \\ \mathbf{x} =$					
	$\prod_{k=1}^{n} \left(k - \sqrt{x} \right) \text{, если } x > b$					
8	$ \prod_{j=4}^{11} \frac{b-x}{\sqrt{j}} , \text{если } x \le 1,3 $	7,6	2,8	10,6	-0,2	-1,0
	$y = \begin{cases} e^{x^2 - b^2} & \text{, если } x > a \\ \arctan\left(x^2 - ba\right) \end{cases}$					
	$\left[\frac{a + b + b + b + a}{2}\right]$, если $b < x < a$					
9	$\begin{cases} \sqrt[5]{e^{x-ba}} & \text{, если } a < x \le b \\ 7 & \text{.} \end{cases}$	1,8	5,4	-1,5	7,2	1,4
	$y = \begin{cases} \sum_{k=2}^{n} \frac{\lg x}{1.1 - \sin k} , \text{если } x > b \end{cases}$					
	$\left \ln^2 \left \sin \left(x^2 - a^2 b \right) \right , \text{если } x \le 1$					

10	ſ	14	0.5	0.2	23	0.23
10	$\cos^2 \sqrt{ a^2 - b^2 x }$, если $x \ge 1.8$	1,1	0,0	•,=	2,2	0,20
	$y = \begin{cases} \ln 1, 4x^2 - a \\ x + x + x \\ x + x $					
	$\prod_{m=3}^{9} \frac{x-m}{x+m} , \text{если } x < b$					
11	$\left[x^2 - b \right]$	1,3	2,7	6,45	-0,2	-0,38
	$\left(\frac{\arctan x}{a}\right)$, если $x < a$					
	$y = \begin{cases} \sqrt{ \sin(b - ax) } & \text{, если } x \ge 3,1 \\ 7 & \hline \end{array}$					
	$\left \sum_{n=2}^{\infty} \sqrt{\left x^2 - nab\right } \text{, если } a < x \le b$					
12	$ \begin{cases} \frac{\sqrt{ x-a }}{e^{ax} + \sin^2(a-bx)} , \text{если } x \ge 5 \end{cases} $	4,8	3,2	1,7	6,8	0,53
	$y = \begin{cases} \prod_{i=4}^{11} \frac{x + \sin i}{\sqrt{i+x}} & \text{, если } x < 3 \end{cases}$					
	$\sqrt[3]{\lg(x^2-a^2)}$, если $b < x \le a$					
13	$ \int_{k=3}^{6} \frac{\sqrt{ k - \cos x }}{k + x} , \text{если } x < a $	2,6	4,9	6,3	-0,7	-0,45
	$y = \left\{ \sqrt[4]{ x^3 - a^2 } $, если $a < x \le 3,5$					
	$\left \frac{\ln ax-b^2 }{2}\right ,$ если $x \ge b$					
14	$\int \log_3 \sqrt{ x-a } \qquad , если \ b < x < a$	3,3	1,8	6,4	-0,2	-0,7
	$y = \left\{ \sum_{j=2}^{8} \frac{j^2 - x}{\sqrt{a} + j} \right\}, \text{если } x > a$					
	$e^{x-a} + b$, если $x < 1$					
15	$\left \ln \left x^3 - a^2 \right \right , если \ a \le x < b$	0,4	1,2	-0,5	2,3	0,21
	$y = \left\{ \operatorname{arctg}^2 \left(x - ab \right)^3 , \text{если } x > b \right.$					
	$\left(\prod_{t=3}^{7} \frac{x^2 + 1}{t}\right), \text{если } x \le 0,1$					
16	$\left[\ln \left \cos \left(ax - b^2 \right) \right \right], \text{если } x \ge a$	1,7	0,2	2,4	-1,3	-0,4
	$y = \begin{cases} \sum_{f=5}^{10} \frac{x-b}{a+\sqrt{f}} & \text{, если } b \ge x < a \end{cases}$					
	$\left \sqrt[4]{ x^2 - a^3 } \cdot e^{x/b} \right $, если $x < -0.32$					

17	ſ	2.8	42	17	86	03
17	1,3tg $\sqrt{ x^2-a }$, если $x > b$	2,0	1,2	1,7	0,0	0,5
	$y = \left\{ \ln \left x - a^2 \right \right\}, \text{если } x < 2$					
	$\prod_{p=3}^{7} \frac{\sqrt{p + \cos x}}{1 + \lg x} , если a \le x \le b$					
18		0,9	2,6	4,3	-1,2	-0,3
	$\operatorname{ctg}^{3}\left(x^{2} - ba\right), \operatorname{если} a < x \leq 2$					
	$y = \left\{ \begin{array}{c} \sqrt{ x^{*} - a^{*} } \cdot e^{x - a^{*}} \\ 9 \end{array} \right\}, \text{ если } x > b$					
	$\sum_{w=4} \lg \frac{w}{b+x} , \text{если } x \le a$					
19	$\left[\operatorname{arcctg}^3 \sqrt{\left x^2 - ab\right }\right]$, если $x \ge b$	-0,9	2,6	-3,1	4,2	0,6
	$y = \begin{cases} \prod_{k=3}^{9} \frac{x + \sin^2 k}{k - a} & \text{, если } x < -2 \end{cases}$					
	$\left \ln \left x^2 - ab^3 \right \qquad , если \ a \le x < b $					
20	$\int_{-\infty}^{7} \sqrt{e^{\nu - x} + b} \qquad , ecли 11 < x < a$	13,7	8,2	15	6,8	-0,8
	$y = \begin{cases} v=2\\ \ln^2 x^2 - ab^2 \\ x^2 - ab^2 \end{cases}, если x \le b$					
	$\left \frac{x^2 + \log_3 a}{b} + \sqrt{ x }\right , \text{если } x \ge a$					
21	$\int \cos^3(x+a)^2$	-2,4	1,5	3	-3	-0,4
	a, если $x < a$					
	$y = \begin{cases} \frac{\sqrt{ x-ab }}{\ln x^2 - a^2 } , \text{если } x \ge 2 \end{cases}$					
	$\sum_{d=5}^{12} \frac{b - d \sin^3 x}{d - x} , \text{если } -1.5 < x \le b$					
22	$\ln \sin^3(ax-b)$	1,8	7,4	10,9	0,2	-0,75
	$1,5 + \cos x$, если $x > 9$					
	$y = \left\{ \sqrt{\left x^2 - e^{a - bx} \right } \text{, если } a < x \le b \right.$					
	$\prod_{k=2}^{7} \frac{\sqrt{\left x^2 + \lg k\right }}{k+a} , если \ x \le a$					
23	$\left[\cos^2 \sqrt{\left x^2 - a^2\right } , если \ a \ge x > 5\right]$	8,6	3,8	1,2	14,1	1,1
	$y = \left\{ \sum_{i=3}^{7} \log_3 \left \frac{a}{x - \sqrt{i}} \right , \text{если } x > 11 \right.$					
	$\left[\operatorname{arcctg}^{3}\left(x-ab\right)^{2}\right]$, если $x \leq b$					

24	$\left\{\prod_{j=4}^{10} e^{x-j}\right\}$, если <i>x</i> < <i>a</i>	-1,4	-0,7	-0,3	-2,2	-0,2
	$y = \left\{ \sin \sqrt[3]{ bx + a } \right\}$,если <i>х</i> ≥ <i>b</i>					
	$\frac{\left \lg^2\right x+a^2b }{a+x}$,если −1 ≤ <i>x</i> < <i>b</i>					
25	$\int_{c=7}^{13} \arcsin \frac{x}{c+a}$, если $2 \ge x < a$	3,4	1,6	5,2	1,2	-0,3
	$y = \left\{ \ln^2 bx + a \right\}$,если <i>x</i> ≤ <i>b</i>					
	$\left[\frac{1}{\sqrt[3]{x^2 - \sqrt{a}}}\right]$,если <i>x</i> ≥ 4					
26	$\int_{-\infty}^{3} \sqrt{\cos^2(x-ba)}$,если <i>х</i> ≥ 6	2,9	4,9	-2,9	8,2	0,7
	$y = \left\{ \prod_{z=3}^{9} \frac{\sqrt{z+x^2}}{a} \right\}$, если <i>x</i> < 2					
	$\left \operatorname{arcctg} \sqrt{\left x^2 - b^2 \right } \right $,если <i>a</i> ≤ <i>x</i> < <i>b</i>					
27	$\int_{4\cos^2(\sqrt{ \mathbf{r}-\mathbf{q} })}$		2,7	5,4	9,8	0,2	-0,35
	$v = \sqrt{\frac{ x - a }{ x - a \cdot \lg 2b }}$	$e \cos a < r < 41$					
	$\int \frac{d^{2} f}{dt} = \int \frac{d^{2} f$, com a < x = 1,1					
	$\left \sum_{f=2}^{\infty} \sqrt{\left e^{j-x}-ab\right }\right $, если $x \leq a$					
28	$\int_{d=1}^{7} \frac{d^2}{x + \ln d}$,если <i>x</i> ≤ <i>b</i>	5,7	3,4	2,1	7,2	0,5
	$y = \begin{cases} \frac{e^{x-a}}{\cos^2(x-b)+a} \end{cases}$, если <i>b</i> < <i>x</i> < 5,1					
	$\frac{\operatorname{arcctg}^{3}\left(ax+b\right)}{x+b}$,если <i>х</i> ≥ 6					
29			3,3	6,8	1,9	12,1	0,85
	$\frac{\lg^2(bx-a)^3}{\lfloor x^2 - a - a^{x-b} \rfloor}$	_, если <i>x</i> ≤ 2,5					
	$y = \begin{cases} \frac{3}{\sqrt{x + b \cdot \cos ax}} \end{cases}$, если <i>x > b</i>					
	$\sum_{k=5}^{12} \frac{\sqrt{k} + x}{\ln k}$,если <i>a</i> < <i>x</i> ≤ 6					
30	b · ar	$\operatorname{ctg}^2(x-b)^3$, если $x \le a$	0,3	4,2	6,2	-1,9	-0,65
	$y = \left\{ \prod_{n=2}^{10} \frac{10}{n} \right\}$	$\frac{\sin^2 n + \cos x}{b - n} , \text{если } x > 5,5$					
	$\ln \sqrt{1}$	$\overline{x^2 - a^2 b^3} \qquad , \text{если } 1, 4 < x \le b$					

Лабораторная работа№16 Использование циклов For...Next, Do...Loop

Задача. Составьте программу для решения задачи.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений и пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Сформировать отчет средствами VBA (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). Отчет должен содержать: условие задачи; формулу расчета с обозначениями и подставленными вместо них числами; полученный результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Варианты задания.

1. Дано целое неотрицательное число. Определите количество нулей при его десятичной записи. Например, для *n* = 103062 ответ 2.

2. Дано натуральное число. Замените все цифры 7 в данном числе цифрами 8 и удалите из записи числа все единицы. Например, для 175718 новое число равно 8588.

3. Выведите на экран числа Фибоначчи от 1 до *n*. Каждое число последовательности из чисел Фибоначчи, начиная с третьего, получается сложением двух предыдущих чисел. Например: 1, 1, 2, 3, 5, 8, 13, 21 ...

4. Вычислите значение дроби, если а и *n* заданы



5. Вычислите значение дроби при заданном значении х:



6. Подсчитайте сумму всех нечетных чисел от 101 до 301.

7. Дано натуральное число *n*. Разложите его на простые множители.

8. Создайте программу, помогающую вкладчику узнать, сколько лет нужно ожидать суммы *S*, если начальный вклад равен *V* при *P* процентах прироста вклада за год.

9. Существуют числа, обладающие следующими свойствами:

- число делится на все свои цифры;

- число, полученное из данного записью цифр в обратном порядке, тоже делится на все свои цифры.

Примером такого числа является 216. Составьте программу нахождения всех трехзначных чисел, обладающих этими свойствами, при этом числа с одинаковыми первой и последней цифрами не рассматривайте.

10. Дано натуральное число *n* < 99. Получите все комбинации выплаты суммы *n* с помощью монет достоинством 1, 5, 10 и 50 коп.

11. В заданном натуральном числе поменяйте порядок цифр на обратный и сравните полученное число с исходным.

12. Для заданного числа найдите все его делители.

13. Натуральное число называют совершенным, если оно равно сумме всех своих делителей, включая 1. Например, 6=1+2+3. Найдите совершенные числа, меньшие заданного числа *N*.

14. Плотность воздуха убывает с высотой по закону $p = p_0 e^{-hz}$, где p— плотность на высоте h метров, $p_0 = 1$, 29 кг/м³, $z = mg/RT = 1.25 \cdot 10^{-4}$. Напечатайте таблицу зависимости плотности от высоты для значений от 0 до 1000 м через каждые 100 м.

15. Одноклеточное животное амеба каждые 3 часа делится на 2 клетки. Определите, сколько клеток будет через 3,6,9,...,24 часа, если первоначально была одна амеба.

16. Гражданин 1 марта открыл счет в банке, вложив 1000 руб. Каждый месяц размер вклада увеличивается на 2 % от имеющейся суммы. Определите:

а) прирост суммы вклада за первый, второй, ..., десятый месяц;

б) сумму вклада через три, четыре, ..., двенадцать месяцев;

в) через сколько месяцев размер вклада превысит 1200 руб.

17. Вычислите приближенно площадь одной арки синусоиды.

18. Известно сопротивление каждого из элементов электрической цепи. Определите общее сопротивление цепи, если:

а) все элементы соединены последовательно;

б) все элементы соединены параллельно.

19. Ежемесячная стипендия студента составляет A руб., а расходы на проживание превышают стипендию и составляют B руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3 %. Составьте программу расчета суммы денег, которую необходимо единовременно попросить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

20. Вычислите количество точек с целочисленными координатами, находящихся в круге радиусом R > 0.

21. Напечатайте в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр (операции деления и нахождения остатка от деления не использовать).

22. Задача Ал-Хорезми (780-850 гг.). Разложите число 10 на 2 слагаемых, сумма квадратов которых равна 58.
23. Определите пары натуральных чисел a < 100 и b < 100, произведение которых в 10 раз больше их суммы. Сколько таких пар?

24. Индийский математик С. Рамануджан обратил внимание на то, что число 1729 можно представить в виде суммы кубов двух чисел двумя способами. Найдите эти числа.

25. Выведите на экран все четырехразрядные числа, в записи которых нет одинаковых цифр.

26. В некоторой стране используются денежные купюры достоинством в 1, 2, 4, 8, 16, 32 и 64. Дано натуральное число *n*. Как наименьшим количеством таких денежных купюр можно выплатить сумму *n* (указать количество купюр каждого достоинства, используемых для выплаты)? Предполагается, что имеется достаточно большое количество купюр всех достоинств.

27. В водоеме 100 т рыбы. Каждый год рыболовецкая бригада вылавливает 15т. Воспроизводство рыбы 5 % в год. Для сохранения воспроизводства необходимо прекращать лов, когда в водоеме ее остается менее 5 т. Определите через сколько лет лов рыбы должен быть прекращен.

28. Дано натуральное число. Определите, какая цифра встречается в нем чаще, 0 или 9.

29. Дано натуральное число. Если в нем есть цифры 2 и 5, то определите, какая из них расположена левее. Если одна или обе эти цифры встречаются в числе несколько раз, то необходимо рассмотреть самые левые из одинаковых цифр.

30. Дано натуральное число. Определите, какая цифра встречается в нем чаще, 0 или 9.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего служат операторы ветвления в программе?

2. Какие значения могут принимать логические выражения?

3. Какие символы используются при записи логических отношений?

4. Назначение логических операций.

5. Порядок вычисления логических выражений.

6. Какой вычислительный процесс называется разветвляющимся?

7. Какие операторы служат для организации разветвляющихся процессов?

8. Чем блочный синтаксис оператора If отличается от однострочного?

9. В каких случаях удобнее использовать оператор множественного выбора (Select Case) вместо оператора ветвления (If...End If)?

10. Какой вычислительный процесс называется циклическим?

11. Какие операторы предназначены для организации циклов?

12. Какие операторы следует использовать, если количество повторений цикла известно заранее?

13. Как работает цикл For. . . Next?

14. Как организовать накопление суммы или произведения?

15. Как организовать цикл Do с постусловием (с предусловием)?

16. В чем отличие цикла с предусловием от цикла с постусловием?

17. Чем цикл Do с условием While отличается от цикла Do с условием Until?

18. Как организовать досрочный выход из цикла?

Лабораторные работы № 17-19

Понятие массива

Массив - набор последовательно индексируемых элементов, имеющих одинаковый внутренний тип данных. Примерами таких данных могут быть результаты экспериментов, цифры статистической отчетности, табличные значения той или иной функции, список фамилий студентов в группе, их оценок по какому-либо предмету и т.п.

Использование массивов в программах позволяет существенно уменьшить размер программы и время разработки за счет использования циклической обработки данных. Например, необходимо хранить все фамилии 25 студентов группы. Для этого можно просто определить и использовать 25 различных переменных типа String. Но, вместо этого, можно объявить один массив, содержащий 25 однотипных элементов.

Другой пример- поле для игры в «Морской бой», которое можно представить в виде двумерного массива из логических переменных, причем, если в точке с координатами (x, y) есть корабль, то значение переменной равно *True* (истина), а если нет, то *False* (ложь).

Таблица значений определенной функции (например, $y=x^2$) тоже представляет собой двумерный массив, но уже числового вида:

x	0	1	2	3	4
y	0	1	4	9	16

Каждый элемент массива характеризуется порядковым номером (индексом), который однозначно определяет положение элемента в массиве. Индексы используются для организации доступа к элементам массива. Количество индексов (размерностей) определяет порядок массива, который в языке Visual Basic не может быть больше 60.

Одномерный массив можно представить в виде строки или столбца соответствующих элементов, значение каждого из которых определяется его порядковым номером (индексом). Одномерный массив называется вектором.

Вектор, элементы которого записаны в строку, называется вектором-строкой. Например:

$$A = [a_1, a_2, a_3, \dots, a_i, \dots, a_n].$$

Вектор, элементы которого записаны в столбец, называется вектором-столбцом. Например:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}.$$

Прямоугольная таблица, составленная из элементов и имеющая n строк и m столбцов, называется двумерным массивом или матрицей размера $n \ge m$. Элементы матрицы обозначаются в виде a_{ij} , где i - но-мер строки, j - номер столбца, на пересечении которых находится этот элемент. Например:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{bmatrix}$$

Так как каждый элемент массива имеет определяющий его уникальный номер (индекс), то изменение одного элемента массива не приводит к изменению других элементов.

Массив представляет собой одну переменную с множеством ячеек памяти для хранения значений, тогда как обычная переменная имеет только одну ячейку, в которой может храниться только одно значение.

Например, для записи денежных трат на каждый день календарного года можно описать один массив с 365 элементами, вместо того, чтобы описывать 365 переменных. При необходимости сослаться на все элементы массива можно ссылаться на массив как целое, но возможны также и ссылки на его отдельные элементы, т.к. каждый элемент массива содержит одно значение.

Массивы подразделяются на статические и динамические.

Статические массивы - это массивы, у которых количество элементов и размерность не изменяются в процессе выполнения программы. Границы статического массива устанавливаются на этапе разработки.

Динамические массивы могут изменять свою размерность в ходе выполнения программы в соответствии с конкретными условиями.

Часто перед выполнением программы количество элементов массива или его размерность неизвестны. Поэтому в процессе выполнения программы возникает необходимость изменять размер массива. В этом случае первоначально массив следует объявить как динамический. А что значит объявить массив? Рассмотрим это далее.

Объявление массивов

Перед использованием в программе массивы должны быть объявлены (описаны). Это требование обязательно как для массивов, которые вводятся в качестве исходных данных, так и для тех, которые формируются в самой программе в результате каких-либо вычислений.

Цель описания массива состоит в том, чтобы предупредить вычислительную систему об ожидаемом типе и количестве элементов массива, под которые система резервирует необходимое число ячеек оперативной памяти, а также о размерности массива, т.е. о числе индексов каждого элемента.

Описание статических массивов так же, как и переменных, осуществляется оператором **Dim**, в которой указывается имя массива и тип содержащихся в нем данных. Кроме того, в отличие от описания переменных, указывается размерность (индексы) массива и количество элементов входящих в каждую из размерностей (индексов). Вме-

149

сто ключевого слова **Dim**, можно использовать также **Public**. Правила применения этих операторов такие же, как и для переменных.

Синтаксис оператора Dim:

Dim ИмяМассива (Индексы) [As Тип]

где **ИмяМассива** - имя массива, удовлетворяющее стандартным правилам именования переменных.

Тип –тип данных переменной (поддерживаются типы Byte, Boolean, Integer, Long, Currency, Single, Double, Decimal (в настоящее время не поддерживается), Date, String (для строк переменной длины), String * длина (для строк фиксированной длины), Object, Variant).

Индексы - размерности массива. Данный параметр имеет следующий синтаксис:

[Нижний То] Верхний [,[Нижний То] Верхний] ...

Т.е. количество элементов в индексе указывается диапазоном от нижнего до верхнего индекса.

Если нижний индекс не указывается, то счет элементов начинается с номера по умолчанию.

Когда нижний индекс не задан явно, нижняя граница массива определяется оператором **Option Base**. В Visual Basic по умолчанию индексирование всегда начинается с нуля, т.е. индекс 0 обозначает первый элемент массива, индекс 1 — второй и т.д. Оператор **Option Base** позволяет задать индексацию массива с 1. Этот оператор должен находиться вне процедур и функций «контейнера» (формы, модуля). Допустимыми значениями для **Option Base** являются только 0 и 1.

Оператор **Dim** должен помещаться в программе раньше любых строк программы, выполняющих те или иные действия с элементами соответствующих массивов.

Некоторые возможные способы объявления массива сведены в табл. 4.

Таблица 4

Способ объявле-	Пример	Количество элементов	
ния массива	объявления масси-	В массиве	
	ва		
Option Base 0 (или	Dim $x(3)$ as Integer	Массив состоит из 4 элементов:	
отсутствует) и мас-		x(0), x(1), x(2), x(3)	
сив задается верх-	Dim $x(2,2)$ as Integer	Массив состоит из 9 элементов	
ним индексом		(3 строки, 3 столбца):	
		x(0,0) $x(0,1)$ $x(0,2)$	
		x(1,0) $x(1,1)$ $x(1,2)$	
		x(2,0) x(2,1) x(2,2)	
Option Base 1, Mac-	Dim $x(3)$ as Integer	Массив состоит из 3 элементов:	
сив задается верх-		x(1), x(2), x(3)	
ним индексом	Dim $x(2,2)$ as Integer	Массив состоит из 4 элементо	
		(2 строки, 2 столбца):	
		x(1,1) x(1,2)	
		x(2,1) x(2,2)	
Option Base 0 (или	Dim x(1 to 3) as Integer	Массив состоит из 3 элементов:	
отсутствует),		x(1), x(2), x(3)	
Option Base 1,		Массив состоит из 4 элементов :	
массив задается	Dim $x(1 \text{ to } 2, 1 \text{ to } 2)$ as	x(1,1) $x(1,2)$	
диапазоном индек-	Integer	x(2,1) x(2,2)	
СОВ			
Для многомерных	Dim $x(1 \text{ to } 2, 2)$ as In-	Массив состоит из 6 элементов	
массивов:	teger	(2 строки, 3 столбца):	
Option Base 0 (или		x(1,0) $x(1,1)$ $x(1,2)$	
отсутствует), массив		x(2,0) $x(2,1)$ $x(2,2)$	

может задаваться	Dim $x(3, 1 \text{ to } 2)$ as In-	Массив состоит из 8 элементов		
как верхним индек-	teger	(4 строки, 2 столбца):		
сом так и диапазо-		x(0,1) $x(0,2)$		
ном индексов		x(1,1) $x(1,2)$		
		x(2,1) $x(2,2)$		
		x(3,1) x(3,2)		
Для многомерных	Dim $x(1 \text{ to } 2, 2)$ as In-	Массив состоит из 4 элементов		
массивов:	teger	(2 строки, 2 столбца):		
Option Base 1, mac-		x(1,1) x(1,2)		
сив может задавать-		x(2,1) x(2,2)		
ся и верхним индек-	Dim $x(3, 1 \text{ to } 2)$ as In-	Массив состоит из 6 элементов		
сом и диапазоном	teger	(3 строки, 2 столбца):		
индексов		x(1,1) $x(1,2)$		
		x(2,1) x(2,2)		
		x(3,1) $x(3,2)$		

Динамический массив создается в два этапа. Сначала массив определяется с помощью операторов Static, Dim, Private, или Public с пустыми скобками, как показано в следующем примере.

```
Dim sngArray() As Single
```

Затем в программе следует вычислить необходимый размер массива в некоторой переменной, например N, и изменить размер динамического массива с помощью оператора **ReDim**.

Синтаксис оператора ReDim:

ReDim [Preserve] ИмяМассива(Индексы) [As Тип]

ИмяМассива, **Тип** – параметры, аналогичные описанным выше в операторе **Dim** при объявлении **статического массива**;

Preserve - параметр (необязательный), это ключевое слово, используемое для сохранения данных в существующем массиве

В отличие от оператора **Dim**, оператор **ReDim** используется только в процедурах. При этом тип данных указывать не обязательно, особенно если он уже определен оператором **Dim**. Оператор **ReDim** можно использовать для изменения числа элементов или размерности массива. Однако нельзя объявить массив с данными одного типа, а затем использовать **ReDim** для приведения массива к другому типу, за исключением случая, когда массив содержит переменные типа Variant. Только в этом случае допустимо приведение массива к явно-му типу данных.

Таким образом, размерность массива можно при необходимости изменить. Однако при этом возникает опасность потерять его содержимое, так как после изменения размерности элементам массива присваиваются значения по умолчанию.

Пример. Пусть у нас задан массив:

2	8	6	3	7	5	6
---	---	---	---	---	---	---

Dim a() as Integer 'Объявляем массив ReDim a(1 to 8) 'Изменяем размерность массива

В результате получим массив:

Количество элементов массива стало равно 8, однако все элементы массива стали равны нулю.

В VBA предусмотрена возможность изменять размерность массива без потери содержимого. Для этого следует использовать **ReDim** вместе с ключевым словом **Preserve**. При этом следует учитывать, что для многомерных массивов можно изменять только «последнее» измерение. Например, если массив имеет только одну размерность, то можно изменить ее, так как она является единственной и «последней» размерностью. Если же массив имеет две или более размерностей, то для сохранения содержимого массива, возможно изменение значения только «последней» из них. Не допускается изменение числа размерностей.

Проиллюстрируем изложенное фрагментом программы

```
Dim a() As Single
ReDim a (10, 10)
ReDim Preserve a(10, 15) 'правильно
ReDim Preserve a(15, 15) 'ошибка
```

В четвертой строке допущена ошибка, т.к. изменяется и первая, и вторая размерности.

При изменении размерности можно изменять и верхнюю, и нижнюю границы индекса. Но если при этом используется ключевое слово **Preserve**, то разрешено изменять только верхнюю границу, что видно из примера:

```
Dim a()As Single
'Переопределение массивов
ReDim a(1 To 10)
ReDim Preserve a(1 To 15) 'правильно
ReDim Preserve a(10 To 15) 'ошибка
```

Оператор **ReDim** можно применять для изменения динамического массива неоднократно. Однако следует запомнить, что если уменьшить размер массива, данные из удаленных элементов будут потеряны.

Оператор **ReDim Preserve** увеличивает размер массива, сохраняя при этом его содержимое. В следующем примере показывается, как можно увеличить массив varArray на 10 элементов без уничтожения текущих значений элементов массива.

ReDim Preserve varArray(UBound(varArray) + 10)

Функция **UBound** (будет описана далее) возвращает значение типа Long, содержащее максимально доступный индекс указанной размерности массива.

В следующем примере оператор **ReDim** используется для первоначального и повторного выделения памяти для динамических массивов.

```
Option Base 1
Dim MyArray() As Integer ' Объявляет динамический массив.
Redim MyArray(5) ' выделяется память для размещения
' 5 элементов.
For I = 1 To 5 ' Выполняет цикл 5 раз.
MyArray(I) = I ' Инициализирует массив.
Next I
```

Теперь нам нужно изменить размер массива и его элементы. Это осуществляется с помощью следующего оператора

```
Redim MyArray(10) ' Увеличивает размер массива до 10 элементов.
For I = 1 To 10 ' Выполняет цикл 10 раз.
MyArray(I) = I ' Инициализирует массив.
Next I
```

Далее изменив размер массива, сохраним существующие элементы.

```
Redim Preserve MyArray(15) 'Увеличивает размер массива до 15 эле-
ментов.
```

Операторы и функции для работы с массивами.

Функция Array

Возвращает значение типа Variant, содержащее массив.

Синтаксис функции Array

Array(списокАргументов)

СписокАргументов представляет разделенный запятыми список значений, присваиваемых элементам массива, содержащегося внутри значения типа Variant. Если аргументы не указываются, создается массив нулевой длины. Для определения элемента массива записывается имя переменной, за которым следуют скобки, содержащие номера индексов нужного элемента. В следующем примере первый оператор создает переменную *A* типа Variant. Второй оператор присваивает целый массив переменной *A*. Последний оператор показывает, как присвоить переменной *B* значение второго элемента массива.

```
Dim A As Variant 'объявляется переменная типа Variant
A = Array(10,20,30) 'массив в скобках присваивает переменной A
'Переменной В присваивается значение элемента массива A(2)
B = A(2)
```

Нижняя граница индексов массива, созданного с помощью функции Array, всегда равняется нулю. В отличие от массивов других типов, она не определяется нижней границей, заданной в операторе Option Base.

Примечание. Значение типа Variant, не описанное как массив, все равно может содержать массив. Переменная типа Variant может содержать массив любого типа, за исключением строк фиксированной длины и определяемых пользователем типов.

В следующем примере функция **Array** используется для возвращения значения типа Variant, содержащего массив.

```
Option Base 1
Dim MyWeek, MyDay
MyWeek = Array("Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Вс")
MyDay = MyWeek(2) ' МуDay содержит "Вт".
MyDay = MyWeek(4) ' МуDay содержит "Чт".
```

Функция LBound

Возвращает значение типа Long, содержащее минимально доступный индекс указанной размерности массива.

Синтаксис функции LBound

LBound (имяМассива[, размерность])

Синтаксис функции LBound содержит следующие элементы:

имяМассива – имя переменной массива, удовлетворяющее стандартным правилам именования переменных.

размерность (необязательный), Variant (Long) – целое число, указывающее размерность, нижнюю границу которой возвращает функция. Для первой размерности следует указать 1, для второй 2 и т.д. Если аргумент размерность опущен, подразумевается значение 1.

В следующем примере LBound возвращает значения, для массива:

```
Dim Lower
Dim A(1 To 100, 0 To 3, -3 To 4)
Dim AnyArray(10)
X = LBound(A, 1) 'Возвращается 1
X = LBound(A, 2) 'Возвращается 0
X = LBound(A, 3) 'Возвращается -3
'Возвращает 0 или 1 (в зависимости от оператора Option Base).
Lower = Lbound(AnyArray)
```

По умолчанию нижняя граница любой размерности имеет значение 0 или 1, в зависимости от значения задаваемого в операторе **Option Base**. Нижней границей в массивах, созданных с помощью функции **Array**, всегда является нуль; вне зависимости от значения задаваемого в операторе **Option Base**.

Массивы, размерности которых задаются с помощью предложения **То** в операторе **Dim, Private, Public, ReDim** или **Static**, могут иметь в качестве нижней границы индекса любое целое значение.

Функция UBound

Возвращает значение типа Long, содержащее максимально доступный индекс указанной размерности массива.

Синтаксис функции UBound

UBound (имяМассива[, размерность])

Синтаксис функции **UBound** содержит следующие элементы:

имяМассива – имя переменной массива, удовлетворяющее стандартным правилам именования переменных.

размерность – необязательный, Variant (Long) - целое число, указывающее размерность, верхнюю границу которой возвращает функция. Для первой размерности используется значение 1, для второй 2 и т.д. Если аргумент размерность опущен, подразумевается значение 1.

Функция **UBound** вместе с функцией **LBound** используется для определения размеров массива.

В следующем примере функция **UBound** используется для определения максимально доступного индекса указанной размерности массива.

```
Dim Upper
Dim MyArray(1 To 10, 5 To 15, 10 To 20) ' Объявляет переменные-
массивы.
Dim AnyArray(10)
Upper = UBound(MyArray, 1) ' Возвращает 10.
Upper = UBound(MyArray, 3) ' Возвращает 20.
Upper = UBound(AnyArray) ' Возвращает 10.
```

Функция IsArray

Возвращает значение типа Boolean, показывающее, является ли переменная массивом.

Синтаксис функции IsArray

IsArray (имяПеременной)

Функция **IsArray** возвращает *True*, если переменная содержит массив; в противном случае возвращается *False*. Функцию **IsArray**

используют для проверки значений типа Variant, содержащих массивы.

В следующем примере функция **IsArray** используется для проверки, является ли переменная массивом.

'Следующая строка описывает массив. Dim MyArray(1 To 5) As Integer, YourArray, MyCheck YourArray = Array(1, 2, 3)' Вызывает функцию Array. MyCheck = IsArray(MyArray)' Возвращает True. MyCheck = IsArray(YourArray) ' Возвращает True.

Оператор Erase

Повторно инициализирует элементы массивов фиксированной длины и освобождает память, отведенную для динамического массива.

Синтаксис оператора Erase

Erase списокМассивов

где списокМассивов представляет имена одной или нескольких очищаемых переменных массивов, разделенных запятой.

Важно знать, имеет ли массив фиксированный размер (обычный массив) или является динамическим, поскольку оператор Erase выполняется по-разному, в зависимости от типа массива. Оператор Erase не освобождает память для массивов фиксированной длины. В табл. 5 показано как оператор Erase действует на элементы массивов фиксированной длины:

Таблица 5

Тип массива	Действие Erase на массив фиксированной длины		
Массив чисел	Присваивает каждому элементу значение 0.		

Массив строк (пере-	Присваивает каждому элементу значение пустой				
менной длины)	строки ("").				
Массив строк (фикси-	Присваивает каждому элементу значение нуля.				
рованной длины)					
Maccив типа Variant	Присваивает каждому элементу значение Empty.				
Массив типа, определя-	Присваивает каждому элементу значение, как от-				
емого пользователем	дельной переменной.				
Массив объектов	Присваивает каждому элементу специальное значе-				
	ние Nothing.				

Для динамических массивов оператор Erase освобождает используемую память. Для того, чтобы из программы вновь стала возможна ссылка на динамический массив, необходимо переопределить размерности переменной массива с помощью оператора **ReDim**.

В следующем **примере** оператор **Erase** используется для повторной инициализации элементов массивов фиксированной длины и освобождения памяти, отведенной для динамического массива.

```
' Объявление переменных-массивов.
Dim NumArray(10) As Integer 'Массив значений типа Integer.
Dim StrVarArray(10) As String 'Массив строк переменной длины.
Dim StrFixArray(10) As String * 10 'Массив строк
                           'фиксированной длины.
Dim VarArray(10) As Variant 'Массив типа Variant.
Dim DynamicArray() As Integer 'Динамический массив.
ReDim DynamicArray(10) 'Занимает память.
Erase NumArray
                 'Каждый элемент становится равным 0.
Erase StrVarArray 'Каждый элемент становится пустой
           ' строкой ("").
Erase StrFixArray 'Каждый элемент становится равным 0.
Erase VarArray
                 'Каждый элемент становится пустым.
Erase DynamicArray 'Освобождает память, используемую массивом.
```

Ввод-вывод одномерных массивов.

Массив в программе определяется поэлементно, т. е. каждому элементу массива присваивается определенное значение. Обращение к конкретному элементу массива осуществляется по его индексу. Например:

```
Dim B(l To 4) As Single
B(l) = 2
B(2) = 7.5
B(3) = 6
B(4) = 11
```

Этот способ определения можно использовать, если количество элементов массива невелико. Однако часто массив содержит десятки или даже сотни элементов, и такой способ становится неприемлемым из-за большого количества операторов присваивания. Поэтому для ввода больших массивов следует организовать цикл, в котором индекс должен изменяться от нижней до верхней границы.

Для ввода значений элементов массива в диалоговом режиме можно воспользоваться функцией **InputBox**.

Пример. Организовать ввод одномерного массива с помощью функции **InputBox** и циклической структуры. Ввод массива закончить, когда в диалоговое окно **InputBox** ничего не будет введено.

```
Sub DemoDinArray()
   объявляем массив без указания размерности
   Dim Massiv() As Integer
   Dim i As Integer 'Объявляем счетчик
   Dim Num 'объявляем переменную Num
   і = 0 'Начальное значение счетчика і
   'организуем бесконечный цикл,
   'т. к. количество вводимых данных заранее неизвестно
   Do
    i = i + 1
    в диалоговом окне вводим значения для элементов массива
   Num = InputBox("Введите элемент массива А("
            & і & ")", "Ввод элементов массива")
      If Len(Num) = 0 Then Exit Do 'выходим из цикла если в окно
                       'InputBox ничего не ввели
    'изменяем размера массива с сохранением элементов
    ReDim Preserve Massiv(i)
     вводим данные в і-й элемент массива
    Massiv(i) = Num
  Loop
```

Рассмотренный способ ввода имеет существенный недостаток: при каждом запуске программы необходимо вновь вводить значения элементов массива. Поэтому при отладке программы или в учебных целях элементы массива можно задавать случайным образом при помощи функции **Rnd** и оператора **Randomize**.

Оператор Randomize

Инициализирует датчик случайных чисел.

Синтаксис оператора Randomize

Randomize [число]

Необязательный аргумент **число** представляет значение типа Variant или любое допустимое числовое выражение.

Оператор **Randomize** использует аргумент **число** для инициализации датчика случайных чисел функции **Rnd**, передавая ему новое опорное число. Если аргумент **число** опущен, в качестве нового опорного числа используется значение, возвращаемое системным таймером.

Функция Rnd

Возвращает значение типа Single, содержащее случайное число. Синтаксис функции **Rnd**

Rnd[(число)]

Необязательный аргумент **число** представляет значение типа Single или любое допустимое числовое выражение. Варианты аргумента **число** и возвращаемые значения функцией **Rnd** сведены в табл. 7.

Таблица 7

Число	Rnd возвращает
Меньше	Каждый раз одно и то же число, используя аргумент число в ка-
нуля	честве опорного числа.
Больше ну-	Следующее случайное число в последовательности.
ля	
Равно нулю	Случайное число, возвращенное при предыдущем вызове этой
	функции.
Не указано	Следующее случайное число в последовательности.

Функция **Rnd** возвращает значение, меньшее 1 и большее или равное нулю.

Перед вызовом функции **Rnd** для инициализации генератора случайных чисел значением, возвращаемым системным таймером можно использовать оператор **Randomize** без аргумента.

Ниже приведен программный код формулы, предназначенной для получения случайных целых чисел в заданном диапазоне:

```
CInt((верхняяГраница - нижняяГраница + 1) * Rnd + нижняяГраница)
```

Здесь верхняяГраница соответствует максимальному числу в диапазоне, а нижняяГраница - минимальному числу в диапазоне.

В следующем примере оператор **Randomize** используется для инициализации генератора случайных чисел. Поскольку числовой аргумент оператора **Randomize** опущен, в качестве нового опорного числа используется значение, возвращаемое функцией **Timer**.

```
Dim x as Integer
Randomize ' Инициализирует генератор случайных чисел.
x = CInt((6 * Rnd) + 1)'Возвращает случайное число от 1 до 6.
```

Еще один **пример** показывает, как создается вектор X(50), значения элементов которого расположены в диапазоне от -10 до 25 случайным образом.

```
Sub massiv2()
Dim X(1 to 50) As Single
Dim i As Integer
'Включить генератор случайных чисел
Randomize
For i = 1 To 50
X(i) = -10 + (25 - (-10)) * Rnd
Next i
End Sub
```

Ввод массива можно осуществить с помощью элементов управления формы, таких как List (список), ComboBox (комбинированное поле), которые представляют собой списки элементов. Также для ввода массива можно использовать файлы данных. Кроме того, в приложения Word и Excel ввод массива можно осуществить с помощью таблиц.

Пример. Создать одномерный массив из элементов, находящихся в таблице документа Word.

Допустим, в документе Word имеется таблицы, состоящие из одной колонки или из одной строки со значениями:



Покажем, как данные из таких таблиц можно присвоить одномерному массиву.

В редакторе VBA вставим модуль, внутри которого поместим процедуру:

```
Sub ReturnCellContentsToArray()
    Dim intCells As Integer 'Объявляем число ячеек
                          'Объявляем объект – ячейка
    Dim celTable As Cell
    Dim sngCells() As Single 'Объявляем динамический массив
    Dim intCount As Integer 'индексы массива
    'Объявление диапазона внутри ячейки,
    ' содержимое этого диапазона помещается в массив
    Dim rngText As Range
    'Считывание данных происходит, если в документе
    существует хотя бы одна таблица
    If ActiveDocument.Tables.Count >= 1 Then
      'Работа внутри диапазона таблицы
     With ActiveDocument.Tables(1).Range
        определение количества ячеек в таблице
       intCells = .Cells.Count
        'переопределение массива
       ReDim sngCells(intCells)
        intCount = 1 'начальный индекс
        'Для каждой ячейки в коллекции ячеек таблицы
       For Each celTable In .Cells
          Установка диапазона внутри каждой из ячеек
          Set rngText = celTable.Range
          'Т.к. в диапазоне внутри ячейки содержится
          скрытый символ уменьшаем данный диапазон
          'на один символ
          rngText.MoveEnd Unit:=wdCharacter, Count:=-1
          'присваиваем массиву содержимое диапазона
          внутри ячейки без скрытого символа
          sngCells(intCount) = CSng(rngText)
          увеличиваем индекс на единицу
          intCount = intCount + 1
       Next celTable
     End With
    End If
```

End Sub

С помощью этого кода можно также считывать данные и из таблицы, которая состоит из нескольких строк и нескольких столбцов. В этом случае данные считываются в одномерный массив из таблицы слева – направо и сверху – вниз.

При создании программ часто возникает задача записи в массив значений функции на определенном интервале. В этом случае следует воспользоваться динамическим массивом.

Например, создадим программу для записи в массив значений функции $y(x) = e^{2-x} + \sqrt{x}$ при изменении *x* от 3 до 7 с шагом 0.25.

```
Sub massiv3()

'Объявление динамического массива

Option Base 1

Dim y() As Single

'Индекс определяемого элемента

Dim i As Integer

Dim X As Single

For X = 3 To 7 Step 0.25

' Перейти к следующему элементу

i = i + 1

'Изменить размер массива

ReDim Preserve y(i)

y(i) = Exp(2 - X) + Sqr(X)

Next X

End Sub
```

Для вывода значений элементов одномерного массива необходимо организовать цикл по индексу массива, а в цикле вывести значение соответствующего элемента на печать.

В следующем примере элементы массива записываются в переменную str_msg, а затем выводятся в диалоговом окне **MsgBox**. Предполагается, что выводимый массив уже существует.

```
str_msg = ""
For j = 1 To i - 1
str_msg = str_msg & X(j) & ", "
Next j
'вызываем стандартное диалоговое окно с кнопкой ОК и помещаем
надпись
MsgBox "Введено: " & str_msg, , "Вывод массива"
```

Вместо диалогового окна массив можно вывести в объекты Label (метка), TextBox (текстовое поле), List (список), ComboBox (комбинированное поле) на форме, а также в документ Word.

Ввод-вывод двумерных массивов.

Каждый элемент двумерного массива (матрицы) характеризуют 2 индекса, первый из которых обозначает номер строки, а второй - номер столбца, на пересечении которых располагается данный элемент. Поэтому, в отличие от одномерных массивов, для ввода-вывода матриц нужно использовать двойные вложенные циклы, в одном из которых параметром является номер строки, а во втором - номер столбца. В остальном же, правила организации ввода-вывода двумерных массивов не отличаются от уже описанных правил ввода-вывода для одномерных массивов.

В общем случае для VBA во всех приложений Microsoft Office и приложениях других сторонних разработчиков со встроенным VBA организовать ввод двумерных массивов можно как показано в следующем примере.

Пример. Организовать ввод в диалоговом режиме целочисленной матрицы *Y*(5,4).

Внешний цикл организовывается по строкам матрицы, внутренний - по столбцам.

```
Sub DvumMassv()

Dim Y() As Integer

ReDim Y(1 To 5, 1 To 4) As Integer

Dim i As Integer, j As Integer

For i = 1 To 5 ' Цикл по строкам

For j = 1 To 4 'Цикл по столбцам

'Ввод элемента матрицы

Y(i, j) = Val(InputBox("Введите Y(" & i _

& "," & j & ")", "Ввод матрицы Y"))

'Печать элемента матрицы

Next j
```

```
'Перейти на следующую строку
Next i
End Sub
```

Для приложений Word и Excel данные можно считывать из таблиц. **Пример**. Создать двумерный массив, в качестве элементов которого, данные берутся из таблицы документа Word.

Допустим, в документе Word имеется таблица со значениями:

54	8	13	46	79
13	46	89	5	6
3	54	9	3	1
3	48	9	7	2

В редакторе VBA вставим модуль, внутри которого поместим процедуру:

```
Sub Massiv6()
 Dim Massiv() As Single 'объявляем массив
 Dim tblOne As Table 'объявляем таблицу tblOne
 Dim celTable As Cell 'объявляем ячейку
 Dim rngTable As Range 'объявляем диапазон
                       'Объявляем кол-во строк
 Dim n As Integer
 Dim m As Integer
                        Объявляем кол-во столбцов
  создаем новую таблицу и устанавливаем на нее ссылку
 Set tblOne = ActiveDocument.Tables(1)
  'присваиваем переменной n общее количество строк
 n = tblOne.Rows.Count
  'присваиваем переменной m общее количество столбцов
 m = tblOne.Columns.Count
  'переопределяем массив
 ReDim Massiv(1 To n, 1 To m)
  'открываем цикл для считывания данных из таблицы tblOne
 For x = 1 То п 'строка
   Y = 1 'столбец
    'для каждой ячейки в коллекции ячеек, которая
    'находится в строке x, в таблице tblOne
   For Each celTable In tblOne.Rows(x).Cells
        устанавливаем диапазон ячейки
       Set rngTable = celTable.Range
        уменьшаем диапазон ячейки на один символ
       rngTable.MoveEnd Unit:=wdCharacter, Count:=-1
```

```
'текст ячейки присваиваем массиву
Massiv(x, Y) = rngTable.Text
Y = Y + 1
Next celTable
Next x
End Sub
```

Операции с массивами

Суммирование и перемножение элементов массивов

Нахождение суммы или произведения всех или некоторых элементов массива является одной из основных операций при работе с массивами.

Пример. Найти сумму элементов одномерного массива A(15).

Чтобы просуммировать все элементы массива, необходимо переменной, в которой будет храниться значение суммы, присвоить нулевое значение, а затем в цикле нарастить ее величину, последовательно прибавляя к ней значение каждого очередного элемента массива, начиная с первого и заканчивая последним.

```
Dim A(1 To 15) As Single, i As Integer, S As Single
'Ввод массива
• • •
S = 0 'Начальное значение суммы
For i = 1 To 15
S = S + A(i) 'Накапливание суммы
Next i
```

Если требуется вычислить произведение, то в приведенном фрагменте достаточно заменить начальное значение переменной на 1, а знак «+» на «*».

Пример. Вычислить произведение элементов, расположенных ниже главной диагонали квадратной матрицы *B*(5,5).

	1	2	3	4	5
1	54	8	13	46	79
2	13	46	89	5	6

3	3	54	9	3	1
4	3	48	9	7	2
5	4	8	9	6	6

Главной диагональю квадратной матрицы называется диагональ, проходящая через левый верхний и правый нижний угол, т. е. совокупность элементов с одинаковыми индексами. Соответственно элементы, расположенные на диагонали, проходящей через правый верхний и левый нижний угол, составляют побочную диагональ матрицы.

Для перебора элементов матрицы ниже главной диагонали организуем двойной цикл. Во внешнем цикле будем изменять номер строки. Как видно из таблицы индекс *i* (номер строки) для элементов ниже главной диагонали изменяется от 2 до 5. Во внутреннем цикле будем изменять номер столбца. Как видно, что для второй строки индекс *j* изменяется от 1 до 1, для третьей - от 1 до 2, для четвертой - от 1 до 3, для пятой - от 1 до 4, т. е. в общем случае - от 1 до (i-1).

```
Sub Massiv7()

Dim B(l To 5, 1 To 5) As Single

Dim i As Integer, j As Integer

Dim P As Single 'Произведение

'Ввод матрицы

'------

'Начальное значение произведения

P = 1

For i = 2 To 5

For j = 1 To i - 1

'Накопление произведения

P = P * B(i, j)

Next j

Next i

End Sub
```

Выбор элементов из массива

Задачи этого типа могут формулироваться по-разному, но в общем случае сводятся к выбору из данного массива элементов с заданными свойствами. Например: в заданном числовом массиве подсчитать количество положительных, отрицательных или нулевых элементов; в массиве целых чисел подсчитать количество четных или нечетных элементов; в таблице функции выбрать значения по модулю большие (или меньшие) заданного; найти сумму или произведение элементов из определенного интервала и т. п.

Пример. Сформировать вектор *Y*, состоящий из элементов кратных трём целочисленного вектора *X*(35).

Для решения поставленной задачи необходимо в цикле для каждого элемента вектора X проверить, делится ли значение элемента на 3 без остатка. Если это условие выполняется, то добавить значение элемента в вектор Y. Так как нельзя заранее точно определить количество элементов в векторе Y, то следует описать его как динамический.

```
Sub Massiv8()
 Dim Y() As Integer
 Dim n As Integer 'Количество элементов в векторе Y
 Dim X(35) As Integer
 Dim i As Integer
 'Ввод вектора Х
  ·-----
 'Формирование вектора У
 n = 0
 For i = 1 To 35
   If X(i) \mod 3 = 0 Then n = n + 1
     ReDim Preserve Y(n)
     Y(n) = X(i)
   End If
 Next
  'Вывод массива
  !_____
End Sub
```

Пример. Подсчитать количество элементов матрицы *A*(6, 8), значения которых лежат в диапазоне от -30 до 50.

```
Sub Massiv9()
 Dim A(l To 6, 1 To 8) As Single
 Dim i As Integer, i As Integer
 Dim n As Integer 'Искомая величина
  'Ввод матрицы А
  !_____
  'Подсчет количества элементов по условию задачи
 п = 0
 For i = 1 To 6
   For j = 1 To 8
     If A(i, j) \ge -30 And A(i, j) \le 50 Then
       n = n + 1
     End If
   Next j
 Next i
End Sub
```

Поиск наибольших или наименьших элементов в массивах

Алгоритм поиска экстремальных значений в массиве состоит в следующем. Например, требуется определить максимальный элемент массива X. Принимаем первоначально за наибольший первый элемент массива X, присваивая его значение переменной max, т. е. $max = X_1$. Затем сравниваем с этой переменной каждый очередной элемент, начиная со 2-го. Если очередной элемент меньше, $X_i < max$, переходим к проверке следующего элемента. Если же $X_i > max$, присваиваем переменной max значение *i*-того элемента массива X, т. е. $max = X_i$ и продолжаем сравнивать следующие элементы массива X уже с этим новым значением max. Если встретится число еще большее, снова производим присваивание и т. д. Последнее значение max после окончания просмотра всех элементов массива X и будет наибольшим.

Пример. В векторе *Y*(40) определить наибольший элемент и его номер.

```
Sub Massiv10()
```

```
Dim Y(1 To 40) As Single
 Dim Max As Single 'Максимальный элемент
 Dim i As Integer
 Dim k As Integer 'Номер max элемента
  'Ввод массива
  ·_____
  'Поиск максимального элемента
 Max = Y(1) 'Начальное значение максимума
 к = 1 'Начальное значение номера тах элемента
 For i = 2 To 40
    If Y(i) > Max Then
     Max = Y(i) 'Новое значение максимума k = i
     'Новый номер max элемента
   End If
 Next i
End Sub
```

Если требуется определить наименьший элемент массива, то в условии **If** знак > нужно заменить на знак <.

Сортировка массивов

Сортировкой массива (или ранжирование) называется расположение элементов массива в порядке возрастания или убывания их значений (размещение элементов по рангу). Общий метод решения задачи ранжирования состоит в просмотре массива, сравнении друг с другом каждой пары рядом стоящих элементов и перемене их местами, если они стоят «не по рангу».

Для решения задачи ранжирования можно использовать метод «пузырька». Суть метода состоит в следующем. Например, необходимо упорядочить массив X по возрастанию. Согласно методу «пузырька», последовательно просматривается массив, сравнивая каждый *i*-й его элемент со следующим *i*+1- м и проверяя их на условие $X_i < X_i + 1$. При этом все пары соседних элементов, удовлетворяющие этому условию, т.е. стоящие «по рангу», пропускаются, а пары, не удовлетворяющие ему, т.е. стоящие не «по рангу», переставляются местами. В заранее заданной переменной, например *Flag*, запоминается, были ли перестановки за весь цикл просмотра.

В результате выполнения одного цикла просмотра самый больший по значению элемент перемещается в конец массива. Отсюда и название метода, т.е. элемент массива, подобно пузырьку воздуха, «всплывает» наверх.

Если в результате выполнения цикла просмотра, перестановок не произошло, то массив уже упорядочен. Если перестановки были, то следует повторить цикл просмотра, исключив последний элемент. Перестановки продолжаются до тех пор, пока в результате цикла просмотра не произойдет перестановок, т.е. переменная *Flag* примет значение *False*.

Сравнение пар чисел X_i и X_i +1 следует вести при изменении индекса от i=1 до i=n-1, где n – переменное количество просматриваемых элементов. Последнее из этих значений i будет соответствовать сравнению предпоследнего элемента массива с последними, ибо последний элемент сравнивать уже не с чем.

Для программной реализации перестановки двух элементов используется следующий алгоритм обмена значений двух переменных A и B: значение переменной A запоминается во вспомогательной переменной C, затем переменной A присваивается значение переменной B, а переменной B - значение переменной C.

Пример. Ранжировать в порядке возрастания вектор A(20).

```
Sub Massiv11()
Rem Coртировка массива по возрастанию
Dim A(1 To 20) As Single
Dim C As Single 'Вспомогательная переменная
Dim Flag As Boolean 'Флаг перестановок
Dim i As Integer, n As Integer
'Ввод массива
'Включить генератор случайных чисел
Randomize
For i = 1 To 20
A(i) = Rnd
```

```
'MsgBox A(i)
    Next i
     'Ранжирование массива
    n = 20 'Количество элементов в массиве
    Do
      Flag = False 'Перестановок не было
      n = n - 1 'Индекс последнего элемента при сравнении
       'Цикл просмотра
      For i = 1 To n
         'Проверка расположения элементов по рангу
         If A(i) > A(i + 1) Then
           'Перестановка
           C = A(i)
           A(i) = A(i + 1)
          A(i + 1) = C
           Flag = True 'Произошла перестановка
        End If
      Next i
    Loop While Flag 'Перейти к началу цикла,
     'если были перестановки
     'Вывод отсортированного массива
    For j = 1 To 20
      str msg = str msg & Chr(13) & A(j) & ", "
    Next
    вызываем стандартное диалоговое окно с кнопкой ОК и помещаем
надпись
    MsgBox "Введено: " & str_msg, , "Вывод ранее введенного массива"
  End Sub
```

Для ранжирования массива по убыванию в приведенном фрагменте программы достаточно в операторе **If** знак > поменять на знак <.

Кроме сортировки методом «пузырька» существует другие методы сортировки массивов, отличающиеся скоростью работы алгоритма.

Характеристики числовых массивов

При работе с числовыми массивами часто возникают задачи определения следующих величин.

Длина вектора

$$L = \sqrt{x_1^2 + x_2^2 + \ldots + x_i^2 + \ldots + x_n^2} = \sqrt{\sum_{i=1}^n x_i^2}.$$

Среднее арифметическое

$$x_{cp} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Дисперсия

$$d = \frac{n}{n-1} \left(\frac{1}{n} \sum_{i=1}^{n} x_i^2 - x_{cp}^2 \right).$$

Среднеквадратическое отклонение

$$\sigma = \sqrt{d}$$
.

Среднее квадратическое

$$S_{\kappa} = \sqrt{\frac{1}{n} \left(x_1^2 + x_2^2 + \ldots + x_i^2 + \ldots + x_n^2 \right)} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}.$$

Среднее геометрическое

$$S_{ge} = \sqrt[n]{x_1 \cdot x_2 \cdot \ldots \cdot x_i \cdot \ldots \cdot x_n} = \sqrt[n]{\prod_{i=1}^n x_i}.$$

Среднее гармоническое

$$S_{ga} = \frac{1}{n} \left(\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_i} + \dots + \frac{1}{x_n} \right) = \frac{1}{n} \sum_{i=1}^n \frac{1}{x_i}$$

Вычисление этих величин сводится в конечном итоге к вычислению соответствующих сумм и произведения.

Транспонирование матриц

Матрица *B*, у которой строки заменены столбцами по отношению к исходной матрице *A*, т.е. $B_{ij} = A_{ji}$, называется транспонированной.

Например:

$$A = \begin{bmatrix} 3 & 8 & 6 & 2 & 7 \\ 1 & 9 & 4 & 8 & 5 \\ 3 & 7 & 2 & 1 & 6 \end{bmatrix}, \quad B = A^{T} = \begin{bmatrix} 3 & 1 & 3 \\ 8 & 9 & 7 \\ 6 & 4 & 2 \\ 2 & 8 & 1 \\ 7 & 5 & 6 \end{bmatrix}.$$

Чтобы транспонировать матрицу, необходимо организовать двойной цикл по строкам и столбцам исходной матрицы *A* и в цикле элементам транспонированной матрицы B присвоить значения элементов матрицы A, причем номер строки элемента матрицы B должен быть равен номеру столбца элемента матрицы A, а номер столбца элемента матрицы A, а номер столбца элемента матрицы A. Количество строк транспонированной матрицы B должно быть равно количеству столбцов исходной матрицы A, а количество строк транспонированной матрицы A, а количество столбцов матрицы B должно быть равно количеству столбцов исходной матрицы A. Далее приведен пример программы транспонирования матрицы.

```
Sub Massiv12()
Dim A(3, 5) As Single, B(5, 3) As Single
Dim i As Integer, j As Integer
'BBOД МАТРИЦЫ A
'------
' Транспонирование матрицы A
For i = 1 To 3
For j = 1 To 5
B(j, i) = A(i, j)
Next j
Next i
End Sub
```

Сложение и вычитание матриц

Суммой матриц A(n,m) и B(n,m) является матрица C(n,m), элементы которой равны суммам соответствующих элементов матриц A и B, т. е.

 $C_{ij} = A_{ij} + B_{ij} \, .$

Разностью матриц A(n,m) и B(n,m) является матрица D(n,m), элементы которой равны разностям соответствующих элементов матриц A и B, т. е.

 $D_{ij} = A_{ij} - B_{ij} \, .$

Следует учесть, что суммировать и вычитать можно матрицы только одного размера.

Пример. Найти сумму и разность матриц *A* и *B* размером 4×7.

```
Sub Massiv13()
 Dim A(l To 4, 1 To 7) As Single
 Dim B(l To 4, 1 To 7) As Single
  'Ввод массивов
  ·_____
 Dim C(l To 4, 1 To 7) As Single 'Сумма
 Dim D(l To 4, 1 To 7) As Single 'Разность
 Dim i As Integer, ј As Integer 'Ввод матриц А и В
 'Вычисление суммы и разности матриц
 For i = 1 To 4
   For j = 1 To 7
     C(i, j) = A(i, j) + B(i, j)
     D(i, j) = A(i, j) - B(i, j)
   Next j
 Next i
  'Вывод полученных массивов
  !_____
End Sub
```

Скалярное произведение векторов

Скалярным произведением векторов A(n) и B(n) называется число, равное сумме попарных произведений их элементов:

$$P = A \cdot B = A_1 \cdot B_1 + A_2 \cdot B_2 + \ldots + A_i B_i + \ldots + A_n B_n = \sum_{i=1}^n A_i \cdot B_i.$$

Скалярно умножать можно вектор на вектор с равным количеством элементов.

Пример. Найти скалярное произведение векторов X(15) и Y(15).

```
Sub Massiv14()
Dim X(l To 15) As Single, Y(l To 15) As Single
Dim i As Integer, P As Single
'Ввод векторов X и Y
'-----
```

```
'Вычисление скалярного произведения векторов

P = 0

For i = 1 To 15

P = P + X(i) * Y(i)

Next i

End Sub
```

Умножение матрицы на вектор

Произведением матрицы A(n, m) на B(m) является вектор C(n), элементы которого C; получены путем скалярного произведения *i*-й строки матрицы A на вектор B, т.е.

$$C_i = \sum_{j=1}^m A_{ij} \cdot B_j$$

Произведение матриц *A* на вектор *B* возможно только в том случае, когда число столбцов матрицы *A* равно количеству элементов вектора *B*.

Например, в результате умножения матрицы A(4, 3) на вектор B(3) получим вектор C(4):

$$C = A \cdot B = \begin{vmatrix} 2 & 5 & 4 \\ 6 & 3 & 8 \\ 5 & 7 & 2 \\ 3 & 9 & 1 \end{vmatrix} \cdot \begin{vmatrix} 3 \\ 2 \\ 4 \end{vmatrix} = \begin{vmatrix} 2 \cdot 3 + 5 \cdot 2 + 4 \cdot 4 \\ 6 \cdot 3 + 3 \cdot 2 + 8 \cdot 4 \\ 5 \cdot 3 + 7 \cdot 2 + 2 \cdot 4 \\ 3 \cdot 3 + 9 \cdot 2 + 1 \cdot 4 \end{vmatrix} = \begin{vmatrix} 32 \\ 56 \\ 37 \\ 31 \end{vmatrix}$$

Для вычисления произведения матрицы на вектор в программе необходимо организовать двойной цикл: внешний по строкам матрицы A и элементам вектора C (i = 1...n), внутренний по столбцам матрицы A и элементам вектора B (i = 1...m).

Пример. Умножить матрицу A(5, 7) на вектор B(7).

```
Sub Massiv15()
Option Base 1
Dim A(5, 7) As Single, B(7) As Single
Dim C(5) As Single 'Результат умножения
Dim i As Integer, j As Integer
```

```
'Ввод матрицы и вектора
'-----
'Вычисление произведения
For i = 1 To 5
C(i) = 0
For j = 1 To 7
C(i) = C(i) + A(i, j) * B(j)
Next j
Next i
End Sub
```

Перемножение матриц

Произведением матрицы A(n, 1) на матрицу B(1, m) называется матрица C(n, m), элементы которой C_{ij} получены путем скалярного произведения *i*-й строки матрицы A на *j*-й столбец матрицы B, т.е.

$$C_{ij} = \sum_{k=1}^{1} A_{ik} \cdot B_{kj} \, .$$

Произведение матриц A и B в указанном порядке возможно в том случае, когда число столбцов матрицы A равно числу строк матрицы B. К произведению матриц в общем случае не применим переместительный закон, т.е. $AB \neq BA$.

Например, в результате перемножения матриц A(2, 2) и B(2, 3) получим матрицу C(2, 3):

 $C = A \cdot B = \begin{vmatrix} 3 & 5 \\ 4 & 2 \end{vmatrix} \cdot \begin{vmatrix} 1 & 8 & 5 \\ 3 & 6 & 7 \end{vmatrix} = \begin{vmatrix} 3 \cdot 1 + 5 \cdot 3 & 3 \cdot 8 + 5 \cdot 6 & 3 \cdot 5 + 5 \cdot 7 \\ 4 \cdot 1 + 2 \cdot 1 & 4 \cdot 8 + 2 \cdot 6 & 4 \cdot 5 + 2 \cdot 7 \end{vmatrix} = \begin{vmatrix} 18 & 54 & 50 \\ 10 & 44 & 34 \end{vmatrix}.$

Как следует из данного примера и общей формулы, матрица - произведение *C* будет содержать *n* строк и *m* столбцов, для формирования которых потребуется двойной цикл. Так как каждый элемент матрицы *C* в свою очередь представляет собой сумму нескольких слагаемых - произведений, для нахождения которой также требуется цикл, то для вычисления произведения двух матриц требуется тройной цикл по индексам i(1...n), j(1...m), k(1...l).
Пример. Вычислить произведение матриц X(4, 6) и Y(6, 5).

```
Sub Massiv16()
 Dim X(1 To 4, 1 To 6) As Single
 Dim Y(1 To 6, 1 To 5) As Single
 Dim Z(1 To 4, 1 To 5) As Single 'Результат
 Dim i As Integer, j As Integer, k As Integer
  'Ввод матриц Х и Ү
  !_____
  !_____
  'Вычисление произведения матриц X и Y
 For i = 1 To 4
   For j = 1 To 5
     Z(i, j) = 0
       For k = 1 To 6
         Z(i, j) = Z(i, j) + X(i, k) * Y(k, j)
       Next k
   Next j
 Next i
End Sub
```

Лабораторная работа №17 Одномерные массивы

Составьте программу, согласно варианту задания.

Создайте форму с элементами управления, если это необходимо.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений и пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Заполнение исходного массива можно осуществить любым из перечисленных выше способов, отдавая предпочтение способу заполнения массива случайными числами или с помощью таблицы в документе Word. Другие способы используйте, если первые два способа не подходят по условию задачи. Сформируйте отчет средствами VBA (обязательно!). Отчет должен содержать:

- условие задачи;

- исходные массивы;

- полученные массивы и результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

1. Заполните массив, применив для его заполнения следующее выражение: $a[i] = \frac{xi^2}{i+x}$.

2. В одномерном массиве целых чисел удалите *k*-й элемент массива.

3. В одномерном массиве целых чисел удалите элемент, равный заданному числу, если он есть. Если таких элементов несколько, то удалите последний элемент из найденных.

4. Вставите на *k*-ю позицию массива целых чисел элемент, равный наименьшему элементу массива.

5. Даны три одномерных числовых массива *A*, *B*, *C*. Сформируйте массив *K*, элементы которого вычисляются по формуле $K[i] = \frac{A[i] + B[i]}{1 + C[i]}$.

6. Заданны два одномерных массива размеров *m* и *n* и число k(k < m). Не создавая дополнительный массив, включите второй массив в первый между *k* -м и (*k*+1)-м его элементами.

7. Заданный массив *А* сдвиньте циклически на *п* элементов вправо. При циклическом сдвиге вправо «выталкиваемые» элементы с конца массива заполняют освобождающиеся места в начале массива. Например, при сдвиге вправо на 3 разряда массива *A*(1,2,3,4,5,6,7) получаем массив *A*(5,6,7,1,2,3,4).

8. Заполните массив:

а) двадцатью первыми натуральными числами, делящимися нацело на 13 или на 17 и большими 300;

б) тридцатью первыми простыми числами.

9. Дан массив. Составьте программу:

а) извлечения квадратного корня из любого элемента массива;

б) расчета среднего арифметического двух любых элементов массива.

10. Дан массив целых чисел. Выясните:

а) является ли *n*-й элемент массива положительным числом;

б) является ли *k*-й элемент массива четным числом;

в) какой элемент массива больше, *k*-й или *n*-й.

11. В массиве хранится информация о сопротивлении каждого из 20 элементов электрической цепи. Определите общее сопротивление цепи, если:

а) все элементы соединены последовательно;

б) все элементы соединены параллельно.

12. Дан массив целых чисел. Выясните:

а) верно ли, что сумма элементов массива есть четное число;

б) верно ли, что сумма квадратов элементов массива есть пятизначное число;

в) верно ли, что сумма элементов массива неотрицательна.

13. В массив A[n] занесены натуральные числа. Найдите сумму тех элементов, которые кратны данному значению *К*.

14. В целочисленной последовательности есть нулевые элементы. Создайте массив из номеров этих элементов.

15. Дана последовательность натуральных чисел $a_1, a_2, ..., a_n$. Создайте массив из четных чисел этой последовательности. Если таких чисел нет, то выведите сообщение об этом.

16. Дана последовательность вещественных чисел $a_1, a_2, ..., a_n$. Замените все ее члены, большие данного числа Z, этим числом. Подсчитайте количество замен.

183

17. Даны вещественные числа $a_1, a_2, ..., a_n$. Поменяйте местами наибольший и наименьший элементы.

18. В заданном одномерном массиве поменяйте местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах.

19. Определите, сколько процентов от всего количества элементов последовательности целых чисел составляют нечетные элементы.

20. Дан вектор $\{z_i\}, i = 1, ..., 50$. Вычислите длину этого вектора: $L = \sqrt{z_1^2 + z_2^2 + ... + z_{50}^2}$.

21. Удалите элемент массива целых чисел, удовлетворяющий условию: остаток от деления на 2 равен 3. Если таких элементов нет, выдайте сообщение «Элементы для удаления не найдены».

22. Удалите элемент массива целых чисел *С*(25), кратный 5. Если таких элементов нет, выдайте сообщение «Элементы для удаления не найдены». Если таких элементов несколько, удалите последний из найденных.

23. Удалите элемент одномерного массива целых чисел, который не делится на 6. Если таких элементов нет, выдайте сообщение «Элементы для удаления не найдены», иначе удалите первый из найденных.

24. В одномерный массив целых чисел на место минимального элемента вставьте максимальный элемент.

25. Удалите из массива:

а) все отрицательные элементы;

б) все элементы, большие данного числа п;

26. Вставьте число, заданное переменной *а* в массив целых чисел после всех элементов, в которых есть цифра 5. Каков должен быть максимальный размер исходного массива?

27. Каждый солнечный день улитка, сидящая на дереве, поднимается вверх на 2 см, а каждый пасмурный день опускается вниз на 1 см. В начале наблюдения улитка находилась на расстоянии *A* см от земли на 5-метровом дереве. Имеется 30-элементный массив, содержащий сведения о том, был ли соответствующий день наблюдения пасмурным или солнечным. Напишите программу, определяющую местоположение улитки к концу 30-го дня наблюдения.

28. Заданы два массива. Создайте один массив, в котором вначале идут четные элементы, затем нечетные.

29. Задан массив, содержащий несколько нулевых элементов. Сожмите его, исключив эти элементы.

30. Дан массив из *n* четырехзначных натуральных чисел. Выведите на экран только те, у которых сумма первых двух цифр равна сумме двух последних.

Лабораторная работа №18 Двумерные массивы

Составьте программу, согласно варианту задания.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений и пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Заполнение исходного массива можно осуществить любым из перечисленных выше способов, отдавая предпочтение способу заполнения массива случайными числами или с помощью таблицы в документе Word. Другие способы используйте, если первые два способа не подходят по условию задачи.

Сформируйте отчет средствами VBA (обязательно!). Отчет должен содержать:

- условие задачи;

- исходные массивы;

- полученные массивы и результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке). 1. Вычислите сумму и число положительных элементов матрицы *A*[*n*, *n*], находящихся над главной диагональю.

2. Задана квадратная матрица. Поменяйте местами строку с максимальным элементом на главной диагонали со строкой с заданным номером.

3. Дана матрица *B*[*n*, *m*]. Найдите в каждой строке матрицы максимальный и минимальный элементы и поменяйте их местами с первым и последним элементом строки соответственно.

4. Определите, является ли заданная целая квадратная матрица *n*-го порядка симметричной (относительно главной диагонали).

5. Задана матрица размером $_{n \times m}$. Найдите максимальный по модулю элемент матрицы. Переставьте строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении *k*-й строки и *k*-го столбца.

6. Дана квадратная матрица *A*[*n*, *n*]. Запишите на место отрицательных элементов матрицы нули, а на место положительных — единицы.

7. Заданы матрица порядка *n* и число *k*. Разделите элементы *k*-й строки на диагональный элемент, расположенный в этой строке.

8. Найдите наибольший и наименьший элементы прямоугольной матрицы и поменяйте их местами.

9. Дана прямоугольная матрица. Найдите строку с наибольшей и наименьшей суммой элементов. Выведите на печать найденные строки и суммы их элементов.

10. Дана матрица *A*(5,5), состоящая из случайных чисел. Составьте одномерный массив *S*, в котором будут содержаться значения суммы элементов каждого столбца матрицы.

11. Образуйте два одномерных массива путем перезаписи в них элементов из заданного целочисленного двухмерного массива размером $n \times n$, при этом в один из формируемых массивов перепишите все

элементы, стоящие выше главной диагонали, а другой — ниже главной диагонали, в порядке, указанном на рис. 9.

12. Найдите суммы элементов двухмерного массива вещественных чисел, расположенных на линиях, параллельных главной диагонали, и выше нее.

13. Найдите номер строки и столбца двухмерного массива для максимального элемента этого массива.



Рис. 9.

14. Дан двухмерный массив. Составьте программу, которая переставляет две любые строки массива.

15. Дан двухмерный массив. Составьте программу, которая переставляет два любых столбца массива.

16. Дан двухмерный массив из четного числа строк. Строки верхней половины массива поменяйте местами со строками нижней половины.

17. Дан двухмерный массив из четного числа строк. Поменяйте местами первую строку со второй, третью — с четвертой и т. д.

18. Дан двухмерный массив. Удалите строку и столбец, на пересечении которых расположен наименьший по модулю элемент массива.

19. Задайте двумерный массив размером 5*5. Выведите массив в виде таблицы. Найдите среднее значение *Sr* элементов второго столбца. Найдите число элементов матрицы, больших *Sr*.

20. Задайте квадратную матрицу размером 5*5. Сформируйте одномерный массив из второй строки матрицы. Сосчитайте число положительных элементов одномерного массива. 21. Дан двухмерный массив из четного числа столбцов. Поменяйте местами первый столбец со вторым, третий — с четвертым и т. д.

22. Дан двухмерный массив из четного числа строк. Поменяйте местами его строки следующим способом: первую строку поменяйте с последней, вторую — с предпоследней и т. д.

23. Поменяйте местами первую строку и строку, в которой находится первый нулевой элемент. Принять, что нулевые элементы в массиве есть, и что массив просматривается слева направо и сверху вниз.

24. Дан двухмерный массив. Удалите из него *k*-ю строку и *s*-й столбец.

25. Дан двухмерный массив. Удалите из него первую из строк, сумма элементов которых не превышает заданное число *n*.

26. Дан двухмерный массив. Удалите из него два столбца, один из которых расположен после *k*-го столбца, а второй — перед *k*-м столбцом.

27. Дан двухмерный массив. Удалите из него все столбцы, номер которых кратен 3.

28. Дан двухмерный массив. Вставьте в него строку из заданной последовательности чисел перед строкой с номером *s*.

29. Дан двухмерный массив целых чисел. Вставьте в него строку из чисел 100 после каждой строки, сумма элементов которой больше заданного числа.

30. Дан двухмерный массив. Переставьте первую строку на место последней, при этом вторую, третью,..., последнюю строки поднять.

Лабораторная работа №19 Матрицы

Составьте программу, согласно варианту задания.

Создайте форму с элементами управления, если это необходимо.

Программа должна содержать форму с текстовыми полями (Text-Box) для ввода величин, метками (Label) для вывода вычисленных значений и пояснительных надписей, рисунками (Image) для отображения формул, кнопками (Command Button) для выполнения расчета, формирования отчета, выхода из программы.

Заполнение исходного массива можно осуществить любым из перечисленных выше способов, отдавая предпочтение способу заполнения массива случайными числами или с помощью таблицы в документе Word. Другие способы используйте, если первые два способа не подходят по условию задачи.

Значения *n* и *m* выбрать из табл. 8 в соответствии с номером варианта по списку. Значения элементов матрицы задать произвольно.

Сформируйте отчет средствами VBA (обязательно!). Отчет должен содержать:

- условие задачи;

- исходные массивы;

- полученные массивы и результат.

В программе предусмотреть защиту от неверного ввода данных (по желанию, оценивается дополнительным 0,5 балла к результирующей оценке).

Таблица 8

Варианты заданий

$\mathcal{N}_{\underline{o}}$	n	т	Задание
1.	4	6	Сформировать вектор из максимальных элементов столбцов матри-
			цы и найти его длину
2.	7	3	Сформировать вектор из сумм индексов минимальных элементов
			строк матрицы и найти его среднее гармоническое
3.	3	8	Сформировать вектор из сумм элементов столбцов матрицы и от-
			сортировать его по убыванию
4.	5	4	Сформировать вектор из произведений элементов строк матрицы и
			найти его среднее гармоническое
5.	5	5	Транспонировать матрицу и найти среднее арифметическое элемен-
			тов побочной диагонали транспонированной матрицы

6.	6	4	Сформировать вектор из минимальных элементов строк матрицы и
			найти его среднее геометрическое
7.	4	5	Сформировать вектор из средних значений элементов столбцов мат-
			рицы и найти его максимальный элемент
8.	8	3	Сформировать вектор из сумм индексов максимальных элементов
			строк матрицы и найти его дисперсию
9.	3	7	Сформировать вектор из средних геометрических столбцов матрицы
			и отсортировать его по возрастанию
10.	4	4	Сформировать вектор из элементов главной и побочной диагоналей
			матрицы и найти их скалярное произведение
11.	6	3	Сформировать вектор из произведений индексов минимальных эле-
			ментов строк матрицы и найти его среднее геометрическое
12.	3	5	Сформировать вектор из произведений столбцов матрицы и найти
			индекс его минимального элемента
13.	7	3	Сформировать вектор из сумм элементов строк матрицы и найти его
			среднее квадратическое
14.	5	5	Сформировать вектор из элементов, расположенных выше главной
			диагонали матрицы, и найти его длину
15.	6	4	Сформировать вектор из произведений элементов строк матрицы и
			отсортировать его по убыванию
16.	3	7	Сформировать вектор из произведений элементов строк матрицы и
			найти его среднее геометрическое
17.	4	5	Сформировать вектор из минимальных элементов столбцов матри-
			цы и найти его среднее гармоническое
18.	4	4	Транспонировать матрицу и найти среднее геометрическое элемен-
			тов побочной диагонали транспонированной матрицы
19.	3	8	Сформировать вектор из средних значений элементов столбцов мат-
			рицы и найти его минимальный элемент
20.	6	4	Сформировать вектор из сумм индексов минимальных элементов
			строк матрицы и найти его дисперсию
21.	3	7	Сформировать вектор из средних арифметических столбцов матри-
			цы и отсортировать его по возрастанию.
22.	5	5	Сформировать вектора из элементов главной и побочной диагона-
			лей матрицы и найти их сумму.
23.	4	6	Сформировать вектор из произведений индексов максимальных
			элементов столбцов матрицы и найти его длину

24.	7	3	Сформировать вектор из произведений строк матрицы и найти ин-
			декс его максимального элемента
25.	5	4	Сформировать вектор из сумм элементов строк матрицы и найти его
			среднее квадратическое отклонение
26.	3	6	Сформировать вектор из произведений элементов строк матрицы и
			отсортировать его по возрастанию
27.	4	4	Сформировать вектор из элементов, расположенных выше побочной
			диагонали матрицы, и найти его длину
28.	6	4	Сформировать вектор из максимальных элементов строк матрицы и
			найти его дисперсию
29.	3	7	Сформировать вектор из сумм индексов максимальных элементов
			строк матрицы и найти его среднее геометрическое
30	5	4	Сформировать вектор из сумм элементов строк матрицы и отсорти-
			ровать его по убыванию

Контрольные вопросы

1. Дать определение массива.

2. Какими параметрами характеризуются массивы?

3. В чем состоит цель описания массивов в программе?

4. Как описать статический массив?

5. Как начать нумерацию элементов массива с единицы?

6. Чем динамические массивы отличаются от статических?

7. Как изменить границы динамического массива без сохранения данных? С сохранением данных?

8. Как программно изменить размер массива управляющих элементов?

9. Как организовать ввод значений элементов массива?

10. Как вывести на печать вектор с заданным количеством элементов в строке?

11. Как организовать ввод матрицы в диалоговом режиме?

12. Как определить нижнюю/верхнюю границу размерности массива?

13. Как организовать накопление суммы или произведения элементов массива?

14. Как найти наибольший или наименьший элемент в массиве?

- 15. Как отсортировать массив в порядке возрастания или убывания?
- 16. Перечислите характеристики массивов.
- 17. Как транспонировать матрицу?
- 18. Как вычислить скалярное произведение векторов?
- 19. Назовите условие перемножения двух матриц.

Заключение.

Прочитав предложенное автором пособие и выполнив упражнения, вы на практике ознакомились с организацией управления ходом программой и обработкой массивов данных. Данные разделы рассмотрены достаточно полно для понимания, но это не значит, что рассмотрены все аспекты тем. Автор пытался создать толчок для дальнейшего самостоятельного изучения и использования возможностей VBA.

Библиографический список

1. Минеев С.П. VBA: Создание простейших программ, создание отчетов в документе Word. С. СамГТУ. 2004.

2. Перри Г., Хеттиева С. Visual Basic 6. Освой самостоятельно за 24 часа.: Пер. с англ.- М.: ЗАО "Издательство БИНОМ", 1999.-384 с: ил.

3. Райтингер М., Мич Г. Visual Basic 6: полное руководство.: Пер. с англ. - К.: Издательская группа BHV, 1999.- 784 с: ил.

4. Гарнаев Ю.А. Visual Basic 6.0: разработка приложений. - СПб: БХВ -Санкт-Петербург, 2000.-448 с: ил.

5. Сайлер Б., Споттс Д. Использование Visual Basic 6. Специальное издание.: Пер. с англ. - М.; СПб.; К.: Издательский дом "Вильяме", 2001. - 832 с: ил.

6. Браун С. Visual Basic 6: учебный курс: Пер. с англ. - СПб: ЗАО "Издательство Питер", 1999.- 576 с: ил.

7. Гуденко. Д, Петроченко Д. Сборник задач по программированию. СПб.: Издательский дом «Питер», 2003.-480 с. ил.