

**МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ
РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**Кафедра «Информационное и математическое обеспечение
транспортных систем»**

Т. Н. ЛИТВИНОВИЧ

РЕШЕНИЕ ЗАДАЧ НА VBA

Учебно-методическое пособие по выполнению лабораторных работ

Гомель 2018

МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ
РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

Кафедра «Информационное и математическое обеспечение
транспортных систем»

Т. Н. ЛИТВИНОВИЧ

РЕШЕНИЕ ЗАДАЧ НА VBA

*Одобрено методической комиссией заочного факультета
в качестве учебно-методического пособия
по выполнению лабораторных работ*

Гомель 2018

УДК 004.438(075.8)

ББК 32.81

Л64

Рецензент – доцент кафедры «Информационные технологии» канд. техн. наук *Ю. А. Пиеничнов* (БелГУТ)

Литвинович, Т. Н.

Л64 Решение инженерных задач на VBA : учеб.-метод. пособие по выполнению лабораторных работ / Т. Н. Литвинович ; М-во трансп. и коммуникаций Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2018. – 33 с.

ISBN 978-985-554-716-8

Представлены теоретические сведения, а также рассмотрены примеры выполнения заданий, входящих в лабораторные работы.

Предназначено для выполнения лабораторных работ раздела «Программирование на VBA» по дисциплине «Информатика» студентами 1-го курса заочного факультета специальности «Промышленное и гражданское строительство».

УДК 004.438(075.8)

ББК 32.81

ISBN 978-985-554-716-8

© Литвинович Т. Н., 2018

© Оформление. БелГУТ, 2018

СОДЕРЖАНИЕ

| | |
|---|----|
| <i>Лабораторная работа № 1. MS Excel: программирование линейных и разветвляющихся алгоритмов.....</i> | 4 |
| Краткие теоретические сведения о функции ЕСЛИ..... | 4 |
| Краткие сведения из теории о программировании на VBA в MS Excel..... | 6 |
| Объектно-ориентированная технология программирования..... | 6 |
| Компоненты среды разработки..... | 7 |
| Базовые элементы языка VBA..... | 8 |
| Алфавит..... | 8 |
| Переменные и константы..... | 8 |
| Оператор присваивания..... | 9 |
| Встроенные функции..... | 11 |
| Процедуры и функции пользователя..... | 11 |
| Создание функции пользователя рабочего листа..... | 12 |
| Организация ввода и вывода данных..... | 13 |
| Программирование линейных алгоритмов..... | 15 |
| Программирование разветвляющихся алгоритмов..... | 17 |
| Порядок выполнения лабораторной работы..... | 22 |
| <i>Лабораторная работа № 2. MS Word: макросы.....</i> | 30 |
| Краткие теоретические сведения о макросах..... | 30 |
| Порядок выполнения лабораторной работы..... | 31 |
| Список литературы..... | 33 |

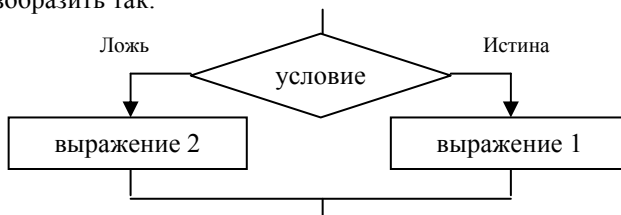
Лабораторная работа № 1

MS EXCEL: ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ И РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

Цель: научиться применять функцию ЕСЛИ для решения прикладных задач, а также расширять функциональность приложения *MS Excel* путем составления программ на *VBA* линейной и разветвляющейся структуры.

Краткие теоретические сведения о функции ЕСЛИ

Функция ЕСЛИ позволяет провести вычисления по одной формуле, если результатом проверки заданного условия является значение ИСТИНА, или по другой формуле – в противном случае. Графически работу данной функции можно изобразить так:



Формат функции ЕСЛИ:

ЕСЛИ (условие; выражение 1; выражение 2)

Здесь

условие – логическое выражение, результатом вычисления которого является одно из двух значений: ИСТИНА или ЛОЖЬ.

выражение 1, выражение 2 – арифметические, логические или строковые выражения, значения которых требуется вычислить. Кроме того, в качестве выражений могут использоваться числовые, логические или текстовые константы. Текстовая константа заключается в двойные кавычки “”.

Условия делятся на две группы: простые и составные.

Для записи простых условий используют стандартные операции-отношения, показанные в таблице 1.

Таблица 1 – Операции отношения

| Операция | Расшифровка | Пример |
|----------|------------------|------------|
| > | Больше | $x > 5$ |
| < | Меньше | $x < 4$ |
| >= | Больше или равно | $x \geq 5$ |
| <= | Меньше или равно | $x \leq 3$ |
| <> | Не равно | $x \neq y$ |
| = | Равно | $x = 2$ |

Составные условия состоят из двух и более простых, связанных между собой логическими функциями И, ИЛИ.

Формат функции И:

И (условие 1; условие 2;...;условие n)

Результатом вычисления функции И является значение ИСТИНА, если результатом проверки **всех** n условий является значение ИСТИНА, и ЛОЖЬ – в противном случае.

Формат функции ИЛИ:

ИЛИ (условие 1; условие 2;...;условие n)

Результатом вычисления функции ИЛИ является значение ИСТИНА, если результатом проверки **хотя бы одного** из n условий является значение ИСТИНА, и ЛОЖЬ – в противном случае.

Пример. Определить, поступили ли указанные в таблице абитуриенты на электротехнический факультет, если проходной балл составил 235:

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---------------|----------------|-------------|----------------|---|---|---|---|---|---|
| 1 | | | | | | | | | | | |
| 2 | | Абитуриент | Суммарный балл | Результат | Проходной балл | | | | | | |
| 3 | | Петухов А.П. | 149 | Не поступил | 235 | | | | | | |
| 4 | | Синицина К.Р. | 290 | Поступил | | | | | | | |
| 5 | | Алейник Е.П. | 278 | Поступил | | | | | | | |
| 6 | | Пирогов А.А. | 199 | Не поступил | | | | | | | |

Формула в ячейке F3: `=ЕС.ЛИ(C3>=E$3;"Поступил";"Не поступил")`

Пример. Определить, сдал ли студент зачет по предмету. Условия сдачи зачета: оценка по каждой контрольной работе не ниже 4, выполнены все лабораторные работы:

| | A | B | C | D | E | F |
|---|---|---|---------------------|----------------------|----------------------|------------|
| 1 | | | | | | |
| 2 | | Ф.И.О. | Лабораторные работы | Контрольная работа 1 | Контрольная работа 2 | Результат |
| 3 | | Петухов А.П. | сданы | 4 | 7 | зачтено |
| 4 | | Синицина К.Р. | не сданы | 6 | 5 | не зачтено |
| 5 | | Алейник Е.П. | сданы | 3 | 4 | не зачтено |
| 6 | | Пирогов А.А. | сданы | 7 | 8 | зачтено |
| 7 | | | | | | |
| 8 | | <code>=ЕС.ЛИ(И(C3="сданы";D3>=4;E3>=4);"зачтено";"не зачтено")</code> | | | | |

Пример. Вычислить значение функции $y(x) = \begin{cases} \sqrt{x}, & \text{если } x \in [3;6] \\ \sin(x) & \text{в остальных случаях} \end{cases}$

| | A | B | C | D | E | F | G | H |
|---|---|-----|-----------|--|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | x | y(x) | =ЕСЛИ(И(B 4>=3;B 4<=6);КОРЕНЬ(B 4);SIN(B 4)) | | | | |
| 4 | | 1 | 0,841471 | | | | | |
| 5 | | 2,6 | 0,5155014 | | | | | |
| 6 | | 4,2 | 2,0493902 | | | | | |
| 7 | | 5,8 | 2,4083189 | | | | | |
| 8 | | 7,4 | 0,8987081 | | | | | |
| 9 | | 9 | 0,4121185 | | | | | |

Краткие сведения из теории о программировании на VBA в MS Excel

VBA (Visual Basic for Application) – язык программирования, который применяется в приложениях пакета *MS Office* для расширения их функциональности. *VBA* относится к классу объектно-ориентированных языков программирования. Рассмотрим кратко основные принципы объектно-ориентированной технологии в программировании.

Объектно-ориентированная технология программирования

Объектно-ориентированное программирование представляет собой способ организации программного кода, при котором главными строительными блоками программы являются объекты, а логика работы программы построена на их взаимодействии. Программирование на *VBA* можно рассматривать, как управление объектами приложения. В *MS Excel*, например, объектом является рабочая книга и ее элементы.

Объект – это конкретный экземпляр класса однотипных объектов. Класс определяет общие для всех его объектов методы и свойства. Свойства – это атрибуты, присущие объектам (размер, цвет и др.). Свойства определяют внешний вид объекта. Формат программного кода, устанавливающего свойство объекта:

Объект.Свойство = Значение

Пример. Выделить ячейку D2 зеленой заливкой.

```
Public Sub fill()
Range("d2").Interior.Color = RGB(0, 255, 0)
End Sub
```

Методы – это процедуры, определяющие взаимодействие объектов с внешней средой. Методы определяют действия, выполняемые над объектом.

Формат программного кода, задающего метод объект:

Объект.Метод

Пример. Удалить формат ячейки *D2*:

```
Public Sub del()  
Range("d2").ClearFormats  
End Sub
```

В *VBA* объекты могут реагировать на события. События инициируются пользователем или системой (системные события). Самое распространенное событие вызываемое пользователем – щелчок кнопки (*Click*). Ответом объекта на произошедшее событие является выполнение программного кода, записанного в процедуре обработки события.

Объектная модель приложения *MS Excel* (сокращенная) представлена на рисунке 1.

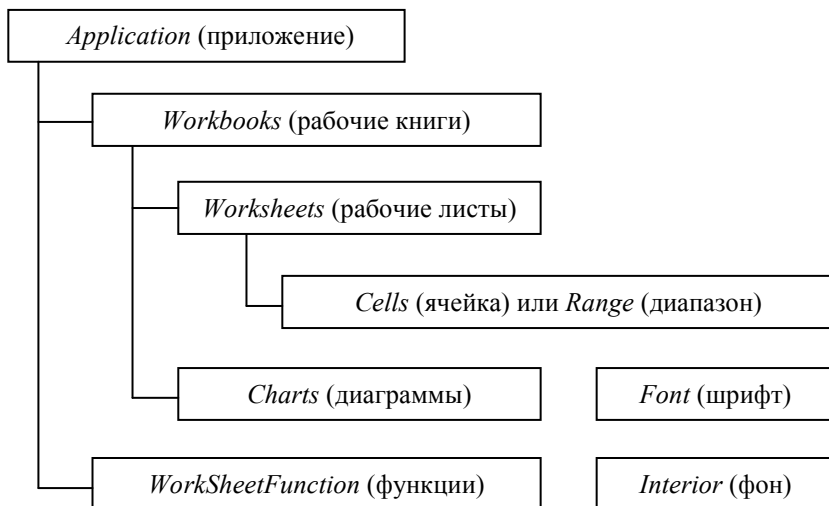


Рисунок 1 – Сокращенная объектная модель *MS Excel*

Компоненты среды разработки

Программы на *VBA* создаются с помощью среды разработки, вызываемой комбинацией клавиш **Alt+F11** или командой **Разработчик – Visual Basic**.

Внешний вид редактора настраивается командами меню *View*. Рекомендуется активизировать следующие рабочие окна: *Project Explorer (Ctrl+R)* – окно проекта, содержащее перечень объектов, составляющих текущий проект; окно настройки свойств объектов *Properties Window (F4)*; окно программного кода *Code (F7)*.

Для отладки программного кода процедур используют команды меню *Debug*. Для выполнения программного кода используется команда *Run (F5)* одноименного меню.

Базовые элементы языка *VBA*

Алфавит

VBA имеет свой набор допустимых символов, составляющих **алфавит** языка:

- прописные и строчные символы латинского и русского алфавитов;
- арабские цифры;
- знак подчеркивания;
- специальные символы (+, -, & и др.).

Алфавит языка используется для составления идентификаторов пользователя (имен констант, переменных, процедур и функций), а также для именования **зарезервированных** слов *VBA* (**встроенных** констант и переменных, типов данных, процедур и функций). Прописные и строчные буквы не различаются.

При выборе идентификаторов пользователя необходимо следовать таким правилам: максимальная длина – 255 символов; может содержать любую комбинацию букв, цифр и допустимых символов и должен начинаться с буквы; допустимые символы (“подчеркивание” не может быть первым символом; символы %, !, \$, &, @ должны быть последними); не должен совпадать с зарезервированными словами.

Переменные и константы

Переменные и константы используются для хранения значений. **Переменная** – именованная область памяти, которая используется для хранения значения, которое при выполнении программы можно изменить. Перед использованием в программе переменную требуется объявить:

Dim имя_переменной [*As* тип]

Здесь

Dim – зарезервированное слово, которое говорит о том, что будет описываться переменная;

тип – тип описываемой переменной.

Примечание – в квадратных скобках указываются необязательные параметры.

Тип данных – центральное понятие, которое существует в любом языке программирования. Тип данных определяет: диапазон значений; совокупность допустимых операций; объем выделяемой памяти и форму представления данных в ней. В *VBA* используются типы данных, представленные в таблице 2.

Таблица 2 – Типы данных VBA

| Тип | Расшифровка | Диапазон |
|----------------|---|---|
| <i>Byte</i> | Целое число | 0 .. 255 |
| <i>Integer</i> | Целое число | -32768 .. 32767 |
| <i>Long</i> | Длинное целое число | -2 147 483 648 .. 2 147 483 647 |
| <i>Single</i> | Число с плавающей запятой одинарной точности | для отрицательных чисел: -3,4E38 .. -1,4E-45; для положительных чисел: 1,4E-45 .. 3,4E38 |
| <i>Double</i> | Число с плавающей запятой двойной точности | для отрицательных чисел: -1,8E308 .. -4,94E-324; для положительных чисел: 4,94E-324 .. 1,8E308 |
| <i>Boolean</i> | Логический | <i>True</i> или <i>False</i> |
| <i>Variant</i> | Универсальный | Значения любого типа |

Константа – именованная область памяти, используемая для хранения значения, которое при выполнении программы нельзя изменить. Перед использованием в процедуре константу требуется объявить:

Const имя_константы [*As* тип] = значение

Здесь

Const – зарезервированное слово, которое говорит о том, что будет описываться константа.

Константы классифицируются в зависимости от типа данных на следующие виды:

- числовые: **6** – целые; **3.76** – вещественные;
- логические: *True* – Истина; *False* – Ложь;
- строковые: "Лабораторная № 5" – записываются в кавычках.

Оператор присваивания

Оператор присваивания осуществляет присваивание результата вычисления некоторого выражения переменной или константе и имеет следующий формат:

идентификатор = выражение

Здесь

идентификатор – имя константы или переменной;
выражение – арифметическое, логическое или строковое выражение, а также константа любого типа.

Выражение – это комбинация констант и переменных, операций, функций и круглых скобок.

Различают *числовые*, *строковые* и *логические* выражения.

Числовое выражение – это выражение, результатом вычисления которого является число.

Строковое выражение – это выражение, результатом вычисления которого является последовательность символов.

Логическое выражение – это выражение, результатом вычисления которого является одна из двух констант: *True* (Истина) или *False* (Ложь).

Все операции в *VBA* можно разделить на три основные группы:

- *арифметические*;
- *отношения*: используются при записи логических выражений;
- *конкатенации &*: используется при записи строковых выражений.

К основным арифметическим операциям относят:

- сложение – “+”;
- вычитание – “-”;
- умножение – “*”;
- деление – “/”;
- остаток от деления – “mod”;
- целочисленное деление – “\”;
- возведение в степень – “^”.

Пример. Даны две переменные: *x* и *y*. Выполнить над ними действия.

| | | | | | | | | |
|----------|----------|---|---|----|-----|------------|---|----|
| <i>x</i> | <i>y</i> | + | - | * | / | <i>mod</i> | \ | ^ |
| 7 | 2 | 9 | 5 | 14 | 3.5 | 1 | 3 | 49 |

Операции отношения: >, <, >=, <=, =, <> “не равно”.

Примеры.

1 Переменная *d* – четная: $d \text{ mod } 2 = 0$.

2 Переменная *s* – положительная: $s > 0$.

3 Переменная *r* кратна числу 7: $r \text{ mod } 7 = 0$.

Порядок выполнения операций называют **приоритетом** операций. В *VBA* установлен следующий порядок выполнения операций:

| Приоритет | Операция или оператор |
|-----------|-----------------------|
| 1 | Вызов функции, скобки |
| 2 | Возведение в степень |
| 3 | Смена знака – |
| 4 | *, / |
| 5 | \ |
| 6 | <i>Mod</i> |
| 7 | +, – |
| 8 | & |
| 9 | =, <>, <, >, <=, >= |
| 10 | <i>And</i> (И) |
| 11 | <i>Or</i> (ИЛИ) |

Встроенные функции

В *VBA* существуют встроенные и пользовательские функции. Встроенные функции делятся на категории. Рассмотрим категорию “Математические функции” (таблица 3).

Таблица 3 – Математические функции *VBA*

| Функция | Описание |
|---------------|--|
| <i>Abs(x)</i> | Возвращает абсолютное значение x |
| <i>Atn(x)</i> | Возвращает арктангенс x (радиан) |
| <i>Cos(x)</i> | Возвращает косинус x (радиан) |
| <i>Sin(x)</i> | Возвращает синус x (радиан) |
| <i>Tan(x)</i> | Возвращает тангенс x (радиан) |
| <i>Log(x)</i> | Возвращает натуральный логарифм x |
| <i>Exp(x)</i> | Возвращает число e в степени x |
| <i>Sqr(x)</i> | Возвращает корень квадратный из x |
| <i>Sgn(x)</i> | Возвращает знак числа x : -1 , если x отрицательное; 1 , если x положительное и 0 , если x равен 0 |
| <i>Rnd()</i> | Возвращает случайное вещественное число из диапазона $(0; 1)$. Используется после инициализации генератора случайных чисел (оператор <i>Randomize</i>) |
| <i>Int(x)</i> | Возвращает целую часть вещественного числа x . Округление не выполняется. Если x – отрицательное, то возвращается целая часть $+1$ |
| <i>Int(x)</i> | Возвращает целую часть вещественного числа x . Округление не выполняется. Если x – отрицательное, то возвращается целая часть $+1$ |

Процедуры и функции пользователя

Решение задач на *VBA* требует создания проекта, который состоит из следующих элементов:

- 1 Приложения, в данном случае *MS Excel*.
- 2 Среды разработки.
- 3 Модулей.

Модули состоят из процедур, в которых хранится программный код проекта. Процедуры делятся на два основных вида: подпрограммы и функции.

В отличие от подпрограмм функции всегда возвращают вычисленное значение.

Процедуры-подпрограммы имеют следующую структуру:

Sub имя ([<список формальных параметров>])

тело процедуры-подпрограммы

End Sub

Процедуры-функции имеют следующую структуру:

Function имя ([<список формальных параметров>])

тело процедуры-функции

имя = выражение

End Sub

Здесь

имя – правильный идентификатор *VBA*;

список формальных параметров – список параметров, которые передаются процедуре при вызове; параметры в списке отделяются друг от друга с помощью запятой; возможно объявление процедуры *без параметров*;

тело процедуры-подпрограммы или тело процедуры-функции – последовательность операторов, приводящих к решению поставленной задачи.

Примечание – В теле процедуры-функции, как правило, последним должен быть оператор присваивания. Полученное в результате значение выражения и является результатом, которое возвращает функция.

В *VBA*, как и в любом другом языке программирования, существует ряд правил оформления программного кода процедур, которых нужно придерживаться.

1 В процедуре выделяют описательную и исполнительную части. Описательная часть обычно содержит описание констант и переменных. Может отсутствовать. Исполнительная часть содержит операторы, приводящие к решению задачи.

2 Для ввода пояснений к программному коду процедуры используют комментарии. Для вставки комментария используется “апостроф”.

3 Длинные инструкции можно размещать в несколько строк. Для этого используется символ продолжения строки: пробел и подчеркивание.

4 Максимальная длина командной строки – 1024 символа.

5 Несколько операторов можно писать в одну строку, разделяя их символом “двоеточие”.

Создание функции пользователя рабочего листа

Функция пользователя рабочего листа описывается так же, как и любая другая пользовательская функция *VBA*. Должна храниться в модуле. Вызывается, как любая стандартная функция *MS Excel*. При использовании Мастера функций пользовательскую функцию надо искать в категории “Определенные пользователем”.

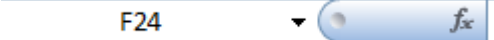
Процесс создания функции пользователя рабочего листа.

1 Загрузить среду разработки *VBA*, нажав *Alt+F11*.

2 Создать новый модуль *Insert – Module*.

3 Сформировать программный код функции.

4 Ввести на любом рабочем листе текущей книги исходные данные и выполнить расчеты, используя функцию пользователя, составленную на *VBA*. Для вызова функции использовать “Мастер функций” f_x :

. Поиск функции осуществлять в категории “Определенные пользователем”.

Организация ввода и вывода данных

В операционной системе *Windows* обмен информацией с пользователем производится посредством всплывающих диалоговых окон.

В *VBA* используются два вида диалоговых окон:

- 1) окна сообщений – служат для организации вывода информации;
- 2) окна для ввода – для организации ввода информации.

Для **ввода информации** используется функция *inputbox*, вызывающая на экран диалоговое окно, содержащее поле для ввода, и кнопки *OK* (подтверждение ввода) и *Cancel* (отмена ввода), имеющая следующий формат:

Inputbox (сообщение [, заголовок] [, умолчание])

Здесь

сообщение – текст-комментарий к строке ввода;

заголовок – текст, выводящийся в строке заголовка диалогового окна;

умолчание – значение по умолчанию.

Примечания

1 Для сохранения введенной в текстовое поле информации в переменную в программе используют оператор присваивания.

2 Для преобразования информации, введенной в текстовое поле, к соответствующему типу применяют функции преобразования формата: *Cint()* – к типу *Integer*, *Csng()* – к типу *Single*, *Cdbl()* – к типу *Double*.

Для **вывода информации**, используется процедура *msgbox*, вызывающая на экран диалоговое окно, содержащее сообщение, имеющая следующий формат:

Msgbox сообщение [, атрибуты] [, заголовок]

Здесь

сообщение – строка, содержащая выводимую информацию;

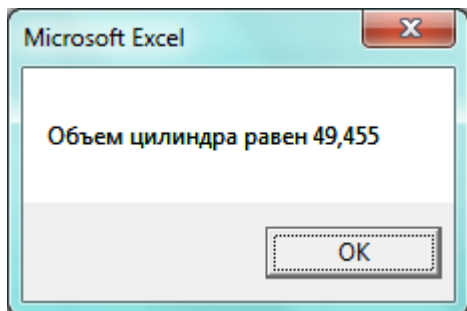
заголовок – текст, выводящийся в строке заголовка диалогового окна;

атрибуты – константы *VBA*, определяющие тип и количество отображаемых в окне кнопок, а также значок к сообщению, указывающий на его тип (информационное, предупреждающее и т. п.).

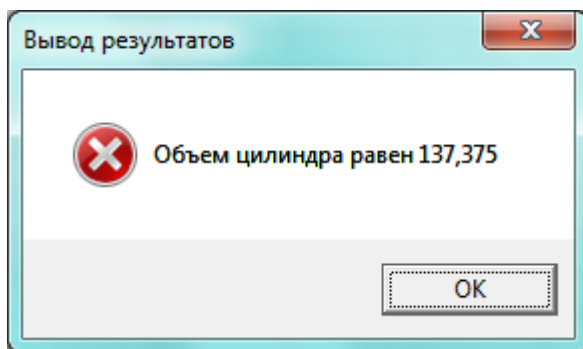
Сообщение может содержать текстовые константы, имена переменных и констант, вызовы функций, а также функцию для перевода курсора на новую строку *Chr(13)*. Все указанные составляющие соединяются в одну строку при помощи операции конкатенации *&*.

Примеры.

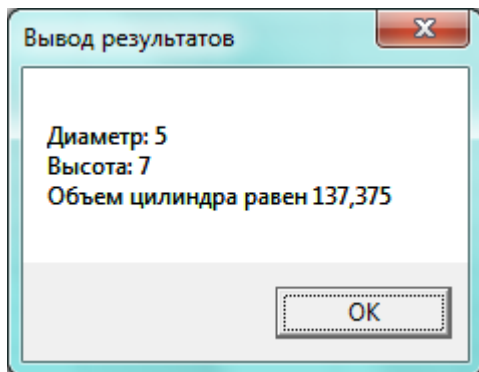
`MsgBox "Объем цилиндра равен " & V`



`MsgBox "Объем цилиндра равен " & V, vbCritical, "Вывод результатов"`



`MsgBox "Диаметр:" & d & Chr(13) & "Высота:" & h & Chr(13) & "Объем цилиндра равен " & V, , "Вывод результатов"`



Ввод и вывод данных можно также производить, читая и записывая информацию в ячейки рабочей книги. Для осуществления чтения или записи информации в ячейки рабочей книги используют:

- 1) объект *Range* ("адрес ячейки");
- 2) свойство *Cells* (номер строки, номер столбца).

Пример. Прочитать (осуществить ввод) значение переменной *a* из ячейки *d2* второго рабочего листа.

```
a=sheets(2).range("d2").value или a=sheets(2).cells(2,4)
```

Пример. Записать (осуществить вывод) значения переменной *a* в ячейку *d2* второго рабочего листа.

```
sheets(2).range("d2").value=a или sheets(2).cells(2,4)=a
```

Программирование линейных алгоритмов

Линейный алгоритм – это алгоритм, в котором действия выполняются строго друг за другом в заранее определенной последовательности.

Для программирования линейных алгоритмов на *VBA* используют инструкцию присваивания, функцию *inputbox()*, а также процедуру *msgbox*.

Задача. Вычислить объем цилиндра, если известен его диаметр *d* и высота *h*.

Решим задачу тремя способами.

1 Для ввода и вывода данных будем использовать *inputbox()* и *msgbox* соответственно.

2 Ввод и вывод данных будем осуществлять, используя возможности чтения и записи информации из ячеек рабочего листа.

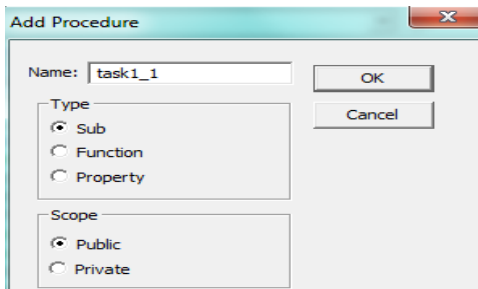
3 Составим линейную функцию пользователя рабочего листа, которую в дальнейшем будем вызывать, используя мастер функций *MS Excel*.

1-й способ решения задачи.

Запустить среду разработки, нажав **Alt+F11**.


Открыть окно программного кода модуля "Эта книга", дважды щелкнув по его имени в перечне модулей проекта.

Сформировать новую процедуру, выполнив команду **Insert – Procedure**, далее в появившемся окне дать ей имя *task1_1* и выбрать тип *Sub*:



Сформировать программный код процедуры следующего вида:

```
Public Sub task1_1()  
' Описательная часть процедуры  
Dim d As Single  
Dim h As Single  
Dim V As Single  
' Исполнительная часть процедуры  
' Ввод исходных данных  
d = CSng(InputBox("Введи диаметр", "Ввод данных"))  
h = CSng(InputBox("Введи высоту", "Ввод данных"))  
' Вычисления  
V = 3.14 * (d / 2) ^ 2 * h  
' Вывод результатов расчета  
MsgBox "Объем цилиндра равен " & V  
End Sub
```

Запустить программу, нажав кнопку  или клавишу F5 на панели инструментов.

2-й способ решения задачи.

Составить на VBA процедуру следующего вида:

```
Public Sub task1_2()  
' Описательная часть процедуры  
Dim d As Single  
Dim h As Single  
Dim V As Single  
' Исполнительная часть процедуры  
' Ввод исходных данных  
d = Sheets(1).Cells(1, 2)  
h = Sheets(1).Cells(2, 2)  
' Вычисления  
V = 3.14 * (d / 2) ^ 2 * h  
' Вывод результатов расчета  
Sheets(1).Cells(1, 1) = "Диаметр"  
Sheets(1).Cells(2, 1) = "Высота"  
Sheets(1).Cells(3, 1) = "Объем цилиндра"  
Sheets(1).Cells(3, 2) = V  
' Установка границ вокруг диапазона результатов  
Range("a1:b3").Borders.Color = black  
End Sub
```

На листе 1 текущей рабочей книги ввести любые числовые значения, соответствующие переменным d и h , в ячейки $b1$ и $b2$.


Запустить программу, нажав кнопку  или клавишу F5 на панели инструментов.

Посмотреть на результаты на листе 1. Они должны выглядеть так:

| | А | В |
|---|----------------|------|
| 1 | Диаметр | 4 |
| 2 | Высота | 5 |
| 3 | Объем цилиндра | 62,8 |

3-й способ решения задачи. Составить на *VBA* пользовательскую функцию вида.

```
Public Function V(d, h As Single)
V = 3.14 * (d / 2) ^ 2 * h
End Function
```

Перейдите на любой лист *Excel*, введите любые значения *d* и *h*, вызовите мастер функций, нажав кнопку  в строке формул, выберите категорию “Определенные пользователем” и функцию *V*, в соответствующие текстовые поля введите адреса ячеек, в которых находятся значения *d* и *h*. Оцените результат.

Программирование разветвляющихся алгоритмов

Разветвляющийся алгоритм – это алгоритм, дальнейшее выполнение действий в котором зависит от результата проверки условия. Для реализации таких алгоритмов в *VBA* применяют оператор условия *IF* и оператор выбора варианта *SELECT CASE*.

Условный оператор *IF* существует в двух базовых вариантах: сокращенном и полном форматах.

1 Сокращенная схема (однострочный формат):

If условие **Then** операторы

2 Сокращенная схема (блочный формат):

If условие **Then**

Операторы

End if

Здесь

условие – логическое выражение, результатом вычисления которого является одна из двух констант: *True* или *False*;

операторы – операторы *VBA*, которые должны выполняться при заданном условии; каждый оператор записывается с новой строки; если операторы записываются в одну строку, то разделяются двоеточием “:”.

Логические выражения делятся на две группы: простые и составные.

Для записи простых выражений используют операции отношения, рассмотренные ранее. Например, $a > 2$ или $v < 4$.

Составные логические выражения состоят из нескольких простых, связанных между собой логическими операциями *And* или *Or*.

Операция *And* (И – логическое умножение) имеет следующий формат:

(условие 1) *And* (условие 2) *And* (условие *n*)

Результатом вычисления операции *And* является значение *True*, если результатом проверки **всех** *n* условий является значение *True*, и *False* – в противном случае.

Операция *Or* (ИЛИ – логическое сложение) имеет следующий формат:

(условие 1) *Or* (условие 2) *Or* (условие *n*)

Результатом вычисления операции *Or* является значение *True*, если результатом проверки **одного из *n*** условий является значение *True*, и *False* – в противном случае.

Пример. Составить условие: переменные *a* и *b* – нечетные.

$(a \bmod 2 <> 0) \text{ And } (b \bmod 2 <> 0)$

Пример. Составить условие: одна из переменных *a* или *b* отрицательна.

$(a < 0) \text{ Or } (b < 0)$

3 Полная схема (однострочный формат):

If условие *Then* операторы 1 *Else* операторы 2

4 Полная схема (блочный формат):

If условие *Then*

операторы 1

Else

операторы 2

End if

Здесь

условие – логическое выражение, результатом вычисления которого является одна из двух констант: *True* или *False*;

операторы 1 – операторы *VBA*, которые должны выполняться, если заданное условие является истинным;

операторы 2 – операторы *VBA*, которые должны выполняться, если заданное условие является ложным.

Задача. Вычислить значение функции $y(x) = \begin{cases} \sqrt{x}, & \text{если } x \in [3; 6] \\ \sin(x) & \text{в остальных случаях} \end{cases}$

Решим задачу двумя способами.

1 Для ввода и вывода данных будем использовать *inputbox()* и *msgbox* соответственно.

2 Составим разветвляющуюся функцию пользователя рабочего листа, которую в дальнейшем будем вызывать, используя мастер функций *MS Excel*.

1-й способ решения задачи. Составить на *VBA* процедуру следующего вида:

```
Public Sub task2_1()  
' Описательная часть процедуры
```

```

Dim x As Single
Dim y As Single
' Исполнительная часть процедуры
' Ввод исходных данных
x = CSng(InputBox("Введи x", "Ввод данных"))
'Вычисления
If (x<=6) and (x>=3) then y=sqr(x) else y=sin(x)
' Вывод результатов расчета
MsgBox "При x= " & x & "y= " & y
End Sub

```

2-й способ решения задачи. Составить на *VBA* функцию пользователя следующего вида:

```

Public Function y(x As Single)
If (x<=6) and (x>=3) then
                y=sqr(x)
            else
                y=sin(x)
End if
End Function

```

Задача. Определить, поступил ли абитуриент со средним балом S на электротехнический факультет, если проходной балл составил 235.

Решим задачу двумя способами.

1 Для ввода и вывода данных будем использовать *inputbox()* и *msgbox* соответственно.

2 Составим разветвляющуюся функцию пользователя рабочего листа, которую в дальнейшем будем вызывать, используя мастер функций *MS Excel*.

1-й способ решения задачи. Составить на *VBA* процедуру следующего вида:

```

Public Sub task3_1()
' Описательная часть процедуры
Dim S As Single
Dim Rez As String
' Исполнительная часть процедуры
' Ввод исходных данных
S = CSng(InputBox("Введи средний балл", "Ввод данных"))
'Вычисления
If S>=235 then Rez="поступил" else Rez="не поступил"
' Вывод результатов расчета
MsgBox "Абитуриент " & Rez
End Sub

```

2-й способ решения задачи. Составить на *VBA* функцию пользователя следующего вида:

```

Public Function Rez(S As Single)
If S>=235 then Rez="поступил" else Rez="не поступил"
End Function

```

Оператор выбора варианта *SELECT CASE* применяется, когда задано n проверяемых условий и нужно выбрать один из вариантов решения задачи в зависимости от установленного для него условия.

Формат оператора *SELECT CASE*:

```
Select case ключ  
Case вариант 1  
    Операторы 1  
Case вариант 2  
    Операторы 1  
Case вариант  $n$   
    Операторы  $n$   
[Case else  
    Операторы else]  
End select
```

Здесь

ключ – числовое или строковое выражение, значение которого надо подобрать;

вариант 1, вариант 2, ..., вариант n – список предлагаемых вариантов для ключа, элементы которого отделяются символом “запятая”.

операторы 1, операторы 2, ..., операторы n – операторы *VBA*, которые должны выполняться при совпадении ключа с соответствующим вариантом; операторы *case else* – операторы *VBA*, которые должны выполняться, если ни один из вариантов ключу не подходит.

Список вариантов для ключа может состоять из следующих элементов:

- выражение;
- выражение 1 *to* выражение 2. Задаёт диапазон значений;
- *Is* операция сравнения выражение.

Задача. Вычислить значение функции $y(x) = \begin{cases} 3 \cdot x, & \text{если } x < -3 \\ \sqrt{x}, & \text{если } x \in [3, 6] \\ 12 \cdot x, & \text{если } x > 12 \\ \cos(x), & \text{иначе} \end{cases}$

Решим задачу двумя способами.

1 Для ввода и вывода данных будем использовать чтение (из ячейки $a2$) и запись (в ячейку $b2$) информации на лист 3 текущей рабочей книги.

2 Составим разветвляющуюся функцию пользователя рабочего листа, которую в дальнейшем будем вызывать, используя мастер функций *MS Excel*.

1-й способ решения задачи. Составить на *VBA* процедуру следующего вида:

```
Public Sub task4_1()  
    ' Описательная часть процедуры  
    Dim x As Single  
    Dim y As Single  
    ' Исполнительная часть процедуры  
    ' Ввод исходных данных  
    x = sheets(3).range("a2")  
    'Вычисления  
    Select case x  
    Case is<-3  
        y=3*x  
    Case 3 to 6  
        y=sqr(x)  
    Case is>12  
        y=12*x  
    Case else  
        y=cos(x)  
    End select  
    ' Вывод результатов расчета  
    sheets(3).range("a1")="x"  
    sheets(3).range("b1")="y"  
    sheets(3).range("b2")=y  
    ' Установка границ вокруг диапазона результатов  
    Range("a1:b2").Borders.Color = black  
End Sub
```

2-й способ решения задачи. Составить на *VBA* функцию пользователя следующего вида:

```
Public Function y(x As Single)  
    Select case x  
    Case is<-3  
        y=3*x  
    Case 3 to 6  
        y=sqr(x)  
    Case is>12  
        y=12*x  
    Case else  
        y=cos(x)  
    End select  
End Function
```

Задача. Написать программу, которая определяет тип треугольника (остроугольный, тупоугольный, прямоугольный) в зависимости от заданного угла.

Решим задачу, составив разветвляющуюся функцию пользователя рабочего листа:

```

Public Function treug(alfa As Single)
Select case alfa
Case is<90
    treug="треугольник остроугольный"
Case is>90
    treug="треугольник тупоугольный"
Case is=90
    treug="треугольник прямоугольный"
End select
End Function

```

Порядок выполнения лабораторной работы

Пример 1. Вычислить значение функции в заданных расчетных точках:

$$f(x) = \begin{cases} \sqrt{x} + e^x, & \text{если } 12 \leq x \leq 20 \\ \cos(x) & \text{в остальных случаях.} \end{cases}$$

Расчетными точками задаться самостоятельно, взяв на рассмотрение 15 – 20 значений, заданных в виде ряда по типу арифметической прогрессии.

Порядок выполнения примера 1.

1 Загрузить *MS Excel*.

2 На листе 1 задать шапку таблицы, начиная с ячейки *B3*:

| | A | B | C |
|---|---|---|------|
| 2 | | | |
| 3 | | x | f(x) |

3 Ввести ряд значений $5 \leq x \leq 27$ $dx = 1.1$.

Алгоритм ввода: в ячейку *B4* ввести первое значение на указанном интервале (число 5); в ячейку *B5* ввести расчетную формулу $=B4+1,1$ для получения второго значения элемента ряда; используя “маркер автозаполнения” получить остальные значения ряда, выполняя протяжку вниз.


| | A | B | C | D |
|---|---|-----|------|---|
| 2 | | | | |
| 3 | | x | f(x) | |
| 4 | | 5 | | |
| 5 | | 6,1 | | |
| 6 | | 7,2 | | |

4 Вычислить $f(x_1)$, введя формулу

$=\text{ЕСЛИ}(\text{И}(\text{B4} \leq 20; \text{B4} >= 12); \text{КОРЕНЬ}(\text{B4}) + \text{EXP}(\text{B4}); \text{COS}(\text{B4}))$.

5 Используя маркер автозаполнения, получить остальные значения $f(x)$.

6 Оформить полученные расчеты в виде таблицы с границами, для ввода

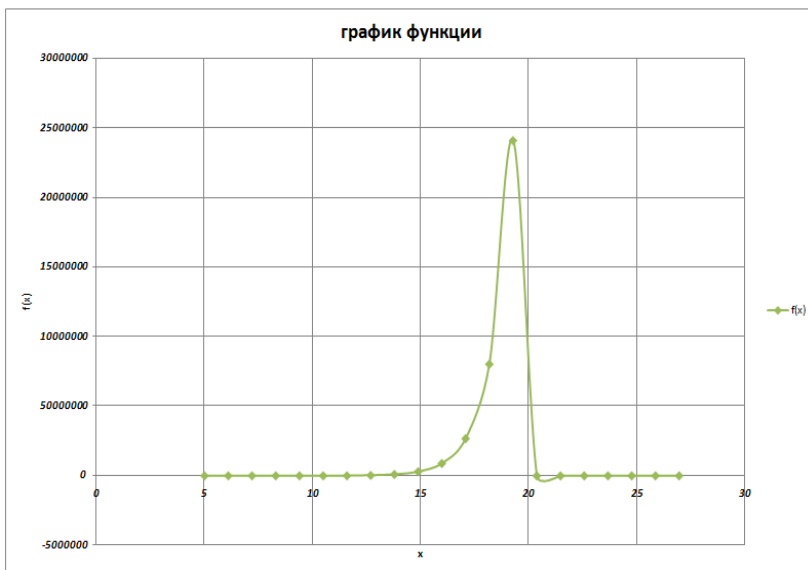
которых использовать соответствующую командную кнопку .

7 Переименовать лист 1 в “Пример 1”.

8 Построить график функции на отдельном листе.

Алгоритм построения: выделить оба столбца расчетной таблицы; выполнить команду **Точечная – гладкими кривыми и маркерами** на вкладке **Вставка**; выполнить команду **Переместить диаграмму** на вкладке **Конструктор** (выбрать – “на отдельном листе”).

9 Отформатировать и отредактировать график, используя команды вкладок **Макет** и **Формат** так, чтобы получился следующий результат:



10 Сохранить файл с именем “Лабораторная 5” в личную папку.

Индивидуальное задание 1. На новом листе с именем “Индивидуальное 1” решить задачу, используя функцию ЕСЛИ. Вычислить значения функции в заданных расчетных точках, построить график функции на отдельном листе и отформатировать на свое усмотрение. Область определения функции выбрать самостоятельно, взяв на рассмотрение 15–20 значений, заданных в виде ряда. Вариант взять из таблицы 4.

Таблица 4 – Индивидуальное задание 1

| Вариант | Задание |
|---------|--|
| 1 | $f(x) = \begin{cases} \frac{x}{e^x}, & \text{если } x > 5 \\ \sin(x) - 2 & \text{в остальных случаях} \end{cases}$ |

Окончание таблицы 4

| Вариант | Задание |
|---------|--|
| 2 | $f(x) = \begin{cases} \ln(x), & \text{если } x \in [-4;7] \\ 2^x & \text{в остальных случаях} \end{cases}$ |
| 3 | $f(x) = \begin{cases} \sin(x) - 3^x, & \text{если } x \geq 7 \\ x - 4 & \text{в остальных случаях} \end{cases}$ |
| 4 | $f(x) = \begin{cases} \sqrt{x} - 2^x, & \text{если } x \in [5;9] \\ \cos(x) - 2 & \text{в остальных случаях} \end{cases}$ |
| 5 | $f(x) = \begin{cases} \sqrt{ x } - \operatorname{tg}(x), & \text{если } x \in [-2;6] \\ \ln(x) - 2 & \text{в остальных случаях} \end{cases}$ |
| 6 | $f(x) = \begin{cases} \cos(x) + 4^x, & \text{если } x \geq 3 \\ x - \sin(x) & \text{в остальных случаях} \end{cases}$ |
| 7 | $f(x) = \begin{cases} x^3 - \sin^2(x), & \text{если } x \in [-8;8] \\ 2x & \text{в остальных случаях} \end{cases}$ |
| 8 | $f(x) = \begin{cases} \frac{2}{\ln(x-1.21)}, & \text{если } x \leq 5 \\ \sin(x+3.12) + 4x & \text{в остальных случаях} \end{cases}$ |
| 9 | $f(x) = \begin{cases} \sqrt{x} - \cos(x+1.5), & \text{если } x \in [8;14] \\ x - 5 & \text{в остальных случаях} \end{cases}$ |
| 10 | $f(x) = \begin{cases} x + e^{3-x}, & \text{если } x \in [-2;6] \\ \cos(2x) & \text{в остальных случаях} \end{cases}$ |
| 11 | $f(x) = \begin{cases} \sin(x) + x^2, & \text{если } x \geq 4 \\ x + \sin(x) & \text{в остальных случаях} \end{cases}$ |
| 12 | $f(x) = \begin{cases} 3x + \ln(x), & \text{если } x \in [1;5] \\ \cos(2x) - 2 & \text{в остальных случаях} \end{cases}$ |

Пример 2. Вычислить объемы шаров, если известны их радиусы: $5 \leq r \leq 15$, $dr = 1.4$. Для решения задачи составить линейную пользовательскую функцию рабочего листа.

Порядок выполнения примера 2.

1 Загрузить редактор *VBE*, нажав комбинацию клавиш **Alt+F11** или выполнив команду **Разработчик – Visual Basic**.

2 Создать новый модуль, выполнив команду главного меню **Insert – Module**.

3 Описать функцию для вычисления объема шара согласно образцу:

```
Public Function V(r As Single)
V = 4 / 3 * 3.14 * r ^ 3
End Function
```

4 На новом листе текущей рабочей книги ввести исходные данные для вычисления.

5 Вычислить объемы шаров, используя при расчете функцию $V(r)$. Для вызова функции использовать команду **Формулы – Вставить функцию**, далее категорию **Определенные пользователем**. Должен получиться такой результат:

| Режим значений | | | Режим формул | | |
|----------------|----------|-----------|--------------|----------|----------|
| | A | B | | A | B |
| 1 | | | 1 | | |
| 2 | | | 2 | | |
| 3 | <i>r</i> | <i>V</i> | 3 | <i>r</i> | <i>V</i> |
| 4 | 5 | 523,33333 | 4 | 5 | =V(A4) |
| 5 | 6,4 | 1097,5096 | 5 | =A4+1,4 | =V(A5) |
| 6 | 7,8 | 1986,7912 | 6 | =A5+1,4 | =V(A6) |
| 7 | 9,2 | 3260,1069 | 7 | =A6+1,4 | =V(A7) |
| 8 | 10,6 | 4986,3875 | 8 | =A7+1,4 | =V(A8) |
| 9 | 12 | 7234,56 | 9 | =A8+1,4 | =V(A9) |
| 10 | 13,4 | 10073,555 | 10 | =A9+1,4 | =V(A10) |
| 11 | 14,8 | 13572,303 | 11 | =A10+1,4 | =V(A11) |

6 Назвать лист “**Пример 2**”.

Индивидуальное задание 2. На новом листе с именем “**Индивидуальное 2**” решить задачу по своему варианту (таблица 5). Для решения задачи составить линейную пользовательскую функцию рабочего листа на *VBA*.

Таблица 5 – Индивидуальное задание 2

| Вариант | Задание |
|---------|--|
| 1 | Найти величину угловой скорости ω искусственного спутника Земли, если известно, что он вращается по круговой орбите с периодом обращения T по формуле $\omega = \frac{2\pi}{T}$ |

Продолжение таблицы 5

| Вариант | Задание |
|---------|--|
| 2 | <p>Определить величину растягивающей силы F, приложенной к крайней точке жестко закрепленного бруса длиной L, сечением A, если он удлинился на величину ΔL по формуле $F = \frac{EA\Delta L}{L}$.</p> <p>Модуль продольной упругости бруса $E = 2 \cdot 10^5$ МПа.</p> |
| 3 | <p>Камень массой m летит со скоростью v. Найти его кинетическую энергию по формуле $E_k = \frac{mv^2}{2}$</p> |
| 4 | <p>Определить потенциальную энергию камня массой m, выпущенного вертикально вверх со скоростью v_0 через время t после начала движения по формуле $E_p = 10m \left(v_0 t + \frac{gt^2}{2} \right)$</p> |
| 5 | <p>Определить период обращения искусственного спутника Земли T, если известна величина его угловой скорости ω по формуле $\omega = \frac{2\pi}{T}$</p> |
| 6 | <p>Определить величину сечения A жестко закрепленного бруса длиной L, если известно, что после того, как к нему была приложена сила F, он удлинился на величину ΔL. Использовать формулу $F = \frac{EA\Delta L}{L}$. Модуль продольной упругости бруса $E = 2 \cdot 10^5$ МПа.</p> |
| 7 | <p>Стул под действием силы F передвинули на расстояние s. Определить, какая работа была при этом совершена по формуле $A = Fs$</p> |
| 8 | <p>Определить период обращения искусственного спутника Земли T, если его орбита расположена на расстоянии h от поверхности Земли и известна величина его линейной скорости v по формуле $v = \frac{2\pi(R_z + h)}{T}$, $R_z = 6,4 \cdot 10^6$.</p> |
| 9 | <p>Определить работу электрического тока в проводнике за время t при силе тока I и напряжении U по формуле $A = IUt$.</p> |
| 10 | <p>Найти величину линейной скорости v искусственного спутника Земли, если известно, что он вращается по круговой орбите с периодом обращения T и его орбита расположена на расстоянии h от поверхности Земли по формуле $v = \frac{2\pi(R_z + h)}{T}$, $R_z = 6,4 \cdot 10^6$.</p> |

Окончание таблицы 5

| Вариант | Задание |
|---------|--|
| 11 | <p>Определить величину удлинения ΔL жестко закрепленного бруса длиной L, сечением A, если к его крайней точке приложена растягивающая сила.</p> <p>Использовать формулу $F = \frac{EA\Delta L}{L}$. Модуль продольной упругости бруса $E = 2 \cdot 10^5$ МПа.</p> |
| 12 | <p>Определить, на какую высоту h взлетел камень массой m, если известно, что через t секунд его потенциальная энергия достигнет величины E_p по формуле $E_p = mgh$.</p> |

Пример 3. Вычислить значения функции в заданных расчетных точках:

$$f(x) = \begin{cases} \sqrt{x} + e^x, & \text{если } 12 \leq x \leq 20 \\ \cos(x) & \text{в остальных случаях.} \end{cases}$$

Для решения задачи составить разветвляющуюся функцию пользователя рабочего листа на *VBA*. Для решения задачи использовать условный оператор *IF*. Область определения функции задать самостоятельно, взяв на рассмотрение 10–15 значений, заданных в виде ряда.

Порядок выполнения примера 3.

1 Вернуться в редактор *VBE*, нажав комбинацию клавиш **Alt+F11** или выполнив команду **Разработчик – Visual Basic**.

2 Создать новый модуль, выполнив команду главного меню **Insert – Module**.

3 Описать функцию, согласно образцу:

```
Public Function f(x As Single)
If (x >= 12) And (x <= 20) Then
    f = Sqr(x) + Exp(x)
Else
    f = Cos(x)
End If
End Function
```

4 На новом листе текущей рабочей книги ввести исходные данные для вычисления согласно образцу, показанному в пункте 5.

5 Провести вычисления, используя для расчета функцию $f(x)$. Для вызова функции использовать команду **Формулы – Вставить функцию**, далее – категорию **Определенные пользователем**.

Должен получиться такой результат:

| Режим значений | | | Режим формул | | |
|----------------|------|-------------|--------------|----------|---------|
| | A | B | | A | B |
| 1 | x | f(x) | 1 | x | f(x) |
| 2 | 1 | 0,54030231 | 2 | 1 | =f(A2) |
| 3 | 2,7 | -0,90407216 | 3 | =A2+1,7 | =f(A3) |
| 4 | 4,4 | -0,30733278 | 4 | =A3+1,7 | =f(A4) |
| 5 | 6,1 | 0,98326842 | 5 | =A4+1,7 | =f(A5) |
| 6 | 7,8 | 0,05395523 | 6 | =A5+1,7 | =f(A6) |
| 7 | 9,5 | -0,99717216 | 7 | =A6+1,7 | =f(A7) |
| 8 | 11,2 | 0,20300468 | 8 | =A7+1,7 | =f(A8) |
| 9 | 12,9 | 400315,63 | 9 | =A8+1,7 | =f(A9) |
| 10 | 14,6 | 2191292,53 | 10 | =A9+1,7 | =f(A10) |
| 11 | 16,3 | 11994989,4 | 11 | =A10+1,7 | =f(A11) |
| 12 | 18 | 65659973,4 | 12 | =A11+1,7 | =f(A12) |
| 13 | 19,7 | 359419495 | 13 | =A12+1,7 | =f(A13) |
| 14 | 21,4 | -0,8303009 | 14 | =A13+1,7 | =f(A14) |
| 15 | 23,1 | -0,44568966 | 15 | =A14+1,7 | =f(A15) |

6 Назвать лист “Пример 3”.

Пример 4. Вычислить значения функции в заданных расчетных точках:

$$f(x) = \begin{cases} \sqrt{x} + e^x, & \text{если } 12 \leq x \leq 20 \\ \cos(x) & \text{в остальных случаях.} \end{cases}$$

Для решения задачи составить разветвляющуюся функцию пользователя рабочего листа на *VBA*. Для решения задачи использовать условный оператор **SELECT CASE**. Область определения функции задать самостоятельно, взяв на рассмотрение 10–15 значений, заданных в виде ряда.

Порядок выполнения примера 4.

1 Вернуться в редактор *VBE*, нажав комбинацию клавиш **Alt+F11** или выполнив команду **Разработчик – Visual Basic**.

2 Создать новый модуль, выполнив команду главного меню **Insert – Module**.

3 Описать функцию согласно образцу:

```
Public Function ff(x As Single)
Select Case x
    Case 12 To 20: ff = Sqr(x) + Exp(x)
    Case Else: ff = Cos(x)
End Select
End Function
```

4 На новом листе текущей рабочей книги ввести исходные данные для вычисления согласно образцу, показанному в пункте 5.

5 Провести вычисления, используя для расчета функцию $ff(x)$. Для вызова функции использовать команду **Формулы – Вставить функцию**, далее – категорию **Определенные пользователем**. Должен получиться такой результат:

| Режим значений | | | Режим формул | | |
|----------------|------|-------------|--------------|----------|----------|
| | A | B | | A | B |
| 1 | x | $f(x)$ | 1 | x | $f(x)$ |
| 2 | 1 | 0,54030231 | 2 | 1 | =ff(A2) |
| 3 | 2,7 | -0,90407216 | 3 | =A2+1,7 | =ff(A3) |
| 4 | 4,4 | -0,30733278 | 4 | =A3+1,7 | =ff(A4) |
| 5 | 6,1 | 0,98326842 | 5 | =A4+1,7 | =ff(A5) |
| 6 | 7,8 | 0,05395523 | 6 | =A5+1,7 | =ff(A6) |
| 7 | 9,5 | -0,99717216 | 7 | =A6+1,7 | =ff(A7) |
| 8 | 11,2 | 0,20300468 | 8 | =A7+1,7 | =ff(A8) |
| 9 | 12,9 | 400315,63 | 9 | =A8+1,7 | =ff(A9) |
| 10 | 14,6 | 2191292,53 | 10 | =A9+1,7 | =ff(A10) |
| 11 | 16,3 | 11994989,4 | 11 | =A10+1,7 | =ff(A11) |
| 12 | 18 | 65659973,4 | 12 | =A11+1,7 | =ff(A12) |
| 13 | 19,7 | 359419495 | 13 | =A12+1,7 | =ff(A13) |
| 14 | 21,4 | -0,8303009 | 14 | =A13+1,7 | =ff(A14) |
| 15 | 23,1 | -0,44568966 | 15 | =A14+1,7 | =ff(A15) |

6 Назвать лист “**Пример 4**”.

Индивидуальное задание 3. На новом листе с именем “**Индивидуальное 3**” решить задачу по своему варианту (см. таблицу 4). Для решения задачи составить разветвляющуюся пользовательскую функцию рабочего листа на *VBA*, используя оператор *IF* или *SELECT CASE*.

Пример 5. Составить разветвляющуюся функцию пользователя рабочего листа с использованием оператора *SELECT CASE* для решения задачи по определению четности числа.

Порядок выполнения примера 5.

1 Вернуться в редактор *VBE*, нажав комбинацию клавиш **Alt+F11** или выполнив команду **Разработчик – Visual Basic**.

2 Создать новый модуль, выполнив команду главного меню **Insert – Module**.

3 Описать функцию согласно образцу:

```
Public Function chet(x As Integer)
Select Case x Mod 2
    Case Is = 0: chet = "четное"
    Case Else: chet = "не четное"
End Select
End Function
```

Примечание – Оператор *mod* вычисляет остаток от деления одного целого числа на другое. Обычно его применяют для определения четности, кратности, делимости.

4 На новом листе текущей рабочей книги ввести исходные данные для вычисления согласно образцу, показанному в пункте 5.

5 Провести вычисления, используя для расчета функцию *ff(x)*. Для вызова функции использовать команду **Формулы – Вставить функцию**, далее – категорию **Определенные пользователем**. Должен получиться такой результат:

| Режим значений | | | Режим формул | | |
|----------------|----|-----------|--------------|----|-----------|
| | A | B | | A | B |
| 1 | | | 1 | | |
| 2 | x | Результат | 2 | x | Результат |
| 3 | 2 | четное | 3 | 2 | =chet(A3) |
| 4 | 3 | нечетное | 4 | 3 | =chet(A4) |
| 5 | 8 | четное | 5 | 8 | =chet(A5) |
| 6 | 11 | нечетное | 6 | 11 | =chet(A6) |
| 7 | 15 | нечетное | 7 | 15 | =chet(A7) |
| 8 | 22 | четное | 8 | 22 | =chet(A8) |

6 Назвать лист “Пример 5”.

Лабораторная работа № 2

MS WORD: МАКРОСЫ

Цель: научиться работать с макросами для расширения функциональности приложения *MS Word*.

Краткие теоретические сведения о макросах

Макрос – программа на *VBA*, в которую при помощи *макрорекодера* транслируются команды, выполняемые пользователем в процессе его записи.

Область применения макросов.

1 *Ускорение обработки текста*. Например, удалить лишние пробелы между словами, выделить латинские буквы курсивом и т. п.

2 *Объединение нескольких команд* с целью одновременного их выполнения.

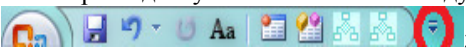
3 *Упрощение доступа к командам*.

Для записи макроса выполнить команду **Разработчик – Запись макроса**, далее в окне указать комбинацию клавиш для запуска макроса, записать команды, далее выполнить команду **Разработчик – Остановить запись**.

Для выполнения макроса:

- 1) выполнить команду **Разработчик – Макросы – Выполнить**;
- 2) использовать комбинацию клавиш;
- 3) использовать кнопку, настроенную на панель быстрого доступа.

Для настройки кнопки на панель быстрого доступа выполните команду

Другие команды, нажав кнопку ; далее в списке слева найдите категорию **Макросы**; выберите требуемый макрос и нажмите кнопку **“Добавить”**. Не выходя из этого окна, настройте значок кнопки макроса, нажав кнопку **Изменить** и осуществив свой выбор.

Порядок выполнения лабораторной работы

Пример. Запишите макрос с именем **“Пример”**, который форматирует текст таким образом: выравнивание – по ширине; отступ первой строки – 2 см; размер символов – 16 пт, начертание – курсив, цвет – зеленый, шрифт *Courier*; автоматическая расстановка переносов.

Порядок выполнения примера.



1 Загрузите *MS Word*.

2 Наберите заданный ниже текст:

Компьютеры в настоящее время используются самыми разными людьми. При этом одни пользователи с настоящим восторгом садятся каждый день к компьютеру, а другие теряются от одной мысли о необходимости работы с ним.

Повсеместное внедрение компьютеров во все сферы человеческой жизни в корне изменило подход к подготовке специалистов. Все окончившие университет специалисты должны уметь работать на компьютере и чувствовать потребность в использовании его как средства для решения возникающих в процессе их практической работы задач.

3 Запишите макрос для форматирования текста:

- выполните команду **Запись макроса** на вкладке **Разработчик**;
- в появившемся окне укажите имя макроса **“Пример”**, комбинацию клавиш для запуска (выбрать любую), доступность макроса **“Эта книга”**;
- установите первые два параметра (выравнивание – по ширине и отступ первой строки – 2 см, используя опции окна, вызываемого командной кнопкой  в группе команд **Абзац** на вкладке **Главная**;
- установите размер текста 16 пт, начертание – курсив, цвет – зеленый, шрифт *Courier*, используя опции окна, вызываемого командной копкой  в группе команд **Шрифт** на вкладке **Главная**;

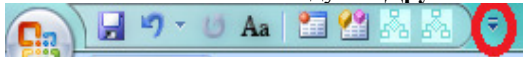
– задайте автоматическую расстановку переносов, выполнив команду **“Разметка страницы”** на вкладке **Расстановка переносов**;

– остановите запись макроса, выполнив команду **“Остановить запись”** на вкладке **Разработчик**.

4 Настройте кнопку для запуска макроса на панель быстрого доступа.

Для этого:

– выполните команду **Другие команды**, нажав кнопку



на панели быстрого доступа;

– далее в списке слева найдите категорию **Макросы**, выберите требуемый макрос и нажмите кнопку **Добавить**. Не выходя из этого окна, настройте значок кнопки макроса, нажав кнопку **Изменить** и осуществив свой выбор.

5 Выделите набранный ранее текст и нажмите только что созданную кнопку, чтобы выполнить макрос.

6 Сохраните файл, выбрав в процессе сохранения тип файла **“Документ Word с поддержкой макросов”**.

Индивидуальное задание. Записать макрос для форматирования текста с параметрами, указанными в таблице 6. Настроить кнопку для запуска макроса с панели быстрого доступа.

Таблица 6 – Индивидуальное задание

| Вариант | Задание |
|---------|--|
| 1 | Шрифт – <i>Tahoma</i> , размер – 12, выравнивание текста – по ширине, красная строка – 0,5 см, двойной междустрочный интервал, автоматическая расстановка переносов |
| 2 | Шрифт – <i>Verdana</i> , размер – 11, выравнивание текста – по ширине, красная строка – 1 см, полуторный междустрочный интервал, автоматическая расстановка переносов, начертание – курсивное |
| 3 | Шрифт – <i>Arial</i> , размер – 9, выравнивание текста – по ширине, красная строка – 1,5 см, тройной междустрочный интервал, начертание – полужирное, автоматическая расстановка переносов |
| 4 | Шрифт – <i>Bookman Old Style</i> , размер – 10, выравнивание текста – по ширине, красная строка – 0,8 см, двойной междустрочный интервал, автоматическая расстановка переносов |
| 5 | Шрифт – <i>Times New Roman</i> , размер – 13, выравнивание текста – по ширине, красная строка – 1,2 см, междустрочный интервал – множитель 1,6, автоматическая расстановка переносов, начертание – курсивное |
| 6 | Шрифт – <i>Calibri</i> , размер – 11, выравнивание текста – по ширине, красная строка – 1,8 см, одинарный междустрочный интервал, начертание – полужирное курсивное, автоматическая расстановка переносов |

Окончание таблицы 6

| Вариант | Задание |
|---------|--|
| 7 | Шрифт – <i>Tahoma</i> , размер – 9, выравнивание текста – по ширине, красная строка – 0,4 см, полуторный междустрочный интервал, автоматическая расстановка переносов |
| 8 | Шрифт – <i>Verdana</i> , размер – 12, выравнивание текста – по ширине, красная строка – 1,4 см, двойной междустрочный интервал, автоматическая расстановка переносов, начертание – полужирное |
| 9 | Шрифт – <i>Arial</i> , размер – 13, выравнивание текста – по ширине, красная строка – 2 см, одинарный междустрочный интервал, начертание – полужирное, автоматическая расстановка переносов |
| 10 | Шрифт – <i>Times New Roman</i> , размер – 12, выравнивание текста – по ширине, красная строка – 0,8 см, тройной междустрочный интервал, автоматическая расстановка переносов |
| 11 | Шрифт – <i>Bookman Old Style</i> , размер – 8, выравнивание текста – по ширине, красная строка – 2 см, двойной междустрочный интервал, автоматическая расстановка переносов, начертание – полужирное курсивное |
| 12 | Шрифт – <i>Calibri</i> , размер – 11, выравнивание текста – по ширине, красная строка – 2,5 см, одинарный междустрочный интервал, начертание – курсивное, автоматическая расстановка переносов |

СПИСОК ЛИТЕРАТУРЫ

- 1 Всемирная паутина [Электронный ресурс]. – Режим доступа: <http://itguides.ru/soft/word/kak-sozdat-makrosy-v-word.html>. – Дата доступа: 25.05.2017.
- 2 Берндт, Г. Измерение, управление и регулирование с помощью макросов VBA в Word и Excel / Г. Берндт, Б. Каинка. – М. : МК-ПРЕСС, 2008. – 246 с.
- 3 Литвинович, Т. Н. Применение VBA для решения прикладных задач : учеб.-метод. пособие по выполнению контрольной работы № 2 / Т. Н. Литвинович ; М-во трансп. и коммуникаций Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2016. – 39 с.

Учебное издание

ЛИТВИНОВИЧ Татьяна Николаевна

РЕШЕНИЕ ЗАДАЧ НА *VBA*

Учебно-методическое пособие
по выполнению лабораторных работ

Редактор *А. А. Павлюченкова*

Технический редактор *В. Н. Кучерова*

Подписано в печать 20.02.2018 г. Формат 60x84 ¹/₁₆.
Бумага офсетная. Гарнитура *Times*. Печать на ризографе.
Усл. печ. л. 2,09. Уч.-изд. л. 1,97. Тираж 100 экз.
Зак. № 607. Изд. № 3

Издатель и полиграфическое исполнение:
Белорусский государственный университет транспорта
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий
№ 1/361 от 13.06.2014.
№ 2/104 от 01.04.2014.
№ 3/1583 от 14.11.2017.
Ул. Кирова, 34, 246653, Гомель