

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

УТВЕРЖДАЮ

Директор ИПР

«\_\_\_\_» \_\_\_\_\_ 2013 г.

## **ИНФОРМАТИКА**

Методические указания и индивидуальные задания  
для студентов ИПР заочной формы обучения, обучающихся по направлениям  
022000 «Экология и природопользование»  
131000 «Нефтегазовое дело»

*Составитель* **О.В. Шефер**

<b>Семестр</b>	<b>2</b>
Кредиты	4
Лекции, часов	8
Лабораторные занятия, часов	6
Практические занятия, часов	4
Индивидуальные задания	1
Самостоятельная работа, часов	108
Формы контроля	зачет

Издательство  
Томского политехнического университета  
2013

УДК 004.09

Информатика: метод. указ. и индивид. задания для студентов ИПР заочной формы обучения, обучающихся по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело» / сост. О.В. Шефер; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2013.– 60 с.

Методические указания и индивидуальные задания рассмотрены и рекомендованы к изданию методическим семинаром Кафедры информатики и проектирования систем « 27 » 08 \_\_\_\_\_ 2013 года, протокол № 1.

Зав. кафедрой ИПС,  
доктор техн. наук \_\_\_\_\_ М. А. Сонькин

#### **Аннотация**

Методические указания и индивидуальные задания по дисциплине «Информатика» предназначены для студентов ИПР заочной формы обучения, обучающихся по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело». Данная дисциплина изучается в одном семестре.

Приведено содержание основных тем дисциплины, указаны перечень лабораторных работ и контрольные вопросы. Приведены варианты заданий для индивидуальной домашней работы. Даны методические указания по выполнению индивидуальной домашней работы.

## 1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Методические указания по изучению дисциплины «Информатика» для студентов ИПР заочной формы обучения, обучающихся по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело» (бакалавр), составлены на основе требований ФГОС ВПО РФ, утвержденных Приказом Минобрнауки № 56 от 30.06.99 и Образовательного стандарта ТПУ и рекомендаций ИДО ТПУ.

Целью настоящей дисциплины является изучение основных компьютерных технологий для использования их в профессиональной деятельности. Задачами настоящей дисциплины является обучение студентов так, что в результате они будут (цитата из ФГОС):

### **знать:**

- основные сведения о дискретных структурах, используемых в персональных компьютерах; один из языков программирования, структуру локальных и глобальных компьютерных сетей;
- основные алгоритмы типовых численных методов решения математических задач;

### **уметь:**

- работать в качестве пользователя персонального компьютера, использовать внешние носители информации для обмена данными между машинами, создавать резервные копии, архивы данных и программ;
- использовать языки и системы программирования, работать с программными средствами общего назначения;
- использовать основные приемы обработки экспериментальных данных;

### **владеть:**

- методами поиска и обмена информацией в глобальных и локальных компьютерных сетях, техническими и программными средствами защиты информации при работе с компьютерными системами, включая приемы антивирусной защиты.

Дисциплина Информатика относится к циклу Б2 – математической и естественнонаучной подготовки. Для ее освоения требуются знания обязательного минимума содержания среднего (полного) образования по информатике, основы курса «Высшая математика» (пререквизиты). Кореквизиты – «Высшая математика», «Физика», «Экология». Знание содержания дисциплины Информатика необходимо для освоения дисциплин, относящихся к профессиональному циклу Б.3.

## **2. СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОГО РАЗДЕЛА ДИСЦИПЛИНЫ**

### **1. Введение в теорию информации и информационные технологии**

Обзор информационных процессов и задач применительно к проблемам экологии и природопользования, нефтегазового дела, технологии геологической разведки. Основные понятия и определения информатики. Взаимосвязь между данными, информацией и знаниями. Определение и основные характеристики процессов получения, переработки, передачи, хранения и использования данных. Основные операции и законы Булевой алгебры логики, дискретное представление чисел, разрядность. Системы счисления. Типы данных.

#### **Методические указания**

Необходимо освоить общие понятия и структуру информатики, основу информационных технологий, иметь представление о свойствах информации, общей характеристике процессов сбора, передачи, обработки информации, а также о системе кодирования в вычислительной технике. Знать основные логические операции.

#### **Вопросы и задания для самоконтроля**

1. Назовите позиционные системы счисления.
2. В чем измеряется количество информации?
3. В чем состоят логические основы ЭВМ?
4. Какое количество различных двоичных чисел  $N$  можно записать с помощью  $I$  двоичных разрядов?
5. Назовите основные цвета модели *СМУК*.
6. Укажите кодировку синего цвета модели *RGB*.

### **2. Аппаратное и программное обеспечение ПК**

Устройство персонального компьютера, основные сведения о дискретных структурах, используемых в персональных компьютерах. Типы и характеристики процессоров, характеристики запоминающих устройств и носителей данных, характеристики линий связи, типы и характеристики мониторов, прочих внешних устройств. Рабочие станции, серверы, кластеры и мэйнфреймы. Особенности современных операционных систем. Классификация программного обеспечения (ПО). Работа в *Microsoft Office*.

## Методические указания

Иметь представление о технических средствах реализации информационных процессов. Изучить архитектуру ПК. Знать основные аппаратные средства ПК и что включает в себя программное обеспечение ПК. Иметь представление о широко используемых операционных системах. Освоить основные операции, выполняемые в приложениях *Office*.

### Вопросы и задания для самоконтроля

1. Каковы особенности классической архитектуры (фон Неймана) ПК?
2. Назовите устройства ввода и вывода ПК.
3. В чем отличия ВЗУ и ПЗУ?
4. Что такое *NTFS*?
5. Назовите основные проводные и беспроводные средства связи персональных сетей.
6. Что такое каталог (*directory*) и каково его назначение в файловой системе?
7. Укажите базовые приемы работы в текстовом редакторе *Microsoft Word*.
8. Какие типы представления адресов ячеек *Excel* вы знаете?
9. В чем особенность абсолютных и относительных адресов ячеек, используемых в формуле при расчетах в *Excel*?
10. Как провести сортировку элементов столбца по возрастанию и по убыванию?

## 3. Основы алгоритмизации. Язык программирования

Основные свойства алгоритмов. Формы представления алгоритмов. Блок–схемы линейных, разветвляющихся и циклических алгоритмов. Сравнительные характеристики языков программирования. Встроенный в приложения *Windows* язык *VBA* Программы, модули, проекты. Элементы управления *VBA*. Объекты, методы и свойства *VBA*. Обмен информацией между листами *Excel* и программой на *VBA*. Константы, переменные, типы *VBA*. Области видимости уровня процедуры, модуля, проекта. Встроенные функции *Excel* и *VBA*. Оконный ввод-вывод. Статические и динамические массивы, логические операторы *VBA*. Логические операции в *Excel* и *VBA*. Условные операторы. Программирование разветвляющихся процессов. Циклы. Программирование циклических процессов *VBA*. Вычисление сумм, произведений, сложных алгебраических и логических выражений.

### **Методические указания**

Освоить основы алгоритмизации. Используя основные алгоритмические структуры (линейная структура, ветвление, цикл), освоить методику разработки алгоритмов. Изучить основы языка *Visual Basic for Applications (VBA)*. Освоить методику программирования с использованием средств *VBA*. Освоить приемы реализации программы с использованием основных управляющих структур.

#### **Вопросы и задания для самоконтроля**

1. Назовите основные этапы компьютерного решения задач.
2. В каких случаях используется управляющая структура *VBA: If - Then - Else*?
3. Для организации какого процесса используется управляющая структура *VBA: For - Next*?
4. Какие действия можно реализовать с помощью управляющей структуры *VBA: While - Wend*?

### **4. Локальные и глобальные компьютерные сети**

Компьютерные сети. Обобщенная структура компьютерной сети. Основные программные и аппаратные компоненты сети. Топология физических связей. Классификация вычислительных сетей. Глобальная сеть *Internet*. Архитектура составной сети. Протоколы. Типы адресов. Служба *WWW*. Понятие информационной безопасности. Угрозы информационной безопасности. Уровни информационной безопасности.

### **Методические указания**

Знать назначение компьютерных сетей. Иметь представление о топологии физических связей компьютерных сетей и классификации вычислительных сетей. Иметь представление о программном сетевом обеспечении и об основном сетевом оборудовании. Знать основные службы Интернет.

#### **Вопросы и задания для самоконтроля**

1. Назовите наиболее часто встречающиеся топологии КС.
2. Какие три типа адреса используются в стеке *TCP / IP*?
3. Что такое символьные доменные имена?
4. Для чего используются *IP* – адреса?
5. Как функционирует электронная почта?
6. Каково назначение службы *FTP*?

### **3. СОДЕРЖАНИЕ ПРАКТИЧЕСКОГО РАЗДЕЛА ДИСЦИПЛИНЫ**

### 3.1. Перечень лабораторных работ

#### Лабораторная работа №1

**Цель работы:** получить простейшие навыки в реализации программ в VBA.

#### *Задания*

Лабораторная работа состоит из 11 заданий (см. приложение 2). Каждое задание - программный код, который нужно набрать в VBA, протестировать и получить результат, уметь ответить на вопрос, что реализовано той или иной строкой программного кода, что означает каждое «слово» программы.

**Замечание.** Перед выполнением лабораторной работы изучите «Методические указания. Введение в VBA» (см. ниже)

## 4. ИНДИВИДУАЛЬНЫЕ ДОМАШНИЕ ЗАДАНИЯ

### (контрольная работа)

#### 4.1. Общие методические указания

Методические указания по изучению дисциплины «Информатика» для студентов ИПР заочной формы обучения, обучающихся по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело» (бакалавр), составлены на основе требований ФГОС ВПО РФ, утвержденных Приказом Минобразования № 56 от 30.06.99 и Образовательного стандарта ТПУ и рекомендаций ИДО ТПУ.

В соответствии с учебным планом для студентов ИПР заочной формы обучения, обучающихся по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело» предусмотрено выполнение одного индивидуального домашнего задания, которое должно содержать решенные задачи. Выполнение этого задания необходимо для закрепления теоретических знаний и приобретения практических навыков программирования и обработки экспериментальных данных.

Курс «Информатика» предполагает знакомство со всеми основными компьютерными технологиями и изучение программирования на языке VBA. Выпускная квалификационная работа студентов по направлениям 022000 «Экология и природопользование», 131000 «Нефтегазовое дело» также содержит элементы этих компьютерных технологий.

Индивидуальное домашнее задание включает решение 3-х типовых задач на широко распространенных программных средствах и ответы на теоретические контрольные вопросы.

Номер варианта, выполняемого задания определяется по двум последним цифрам номера зачетной книжки. Если указанное число превышает «25», то из него следует вычесть 25, полученный результат будет определять номер варианта задания.

#### **4.2. Варианты домашних заданий и методические указания**

Варианты ИДЗ представлены в приложении.

**Задача 1. Вычислить значения аналитического выражения (линейный алгоритм).**

**Цель работы:** составить программу для вычисления значения функции  $y(x)$  при заданном значении аргумента  $x$ ; вывести значения аргумента и функции.

##### **Порядок работы**

1. Изучите методические указания к лабораторной работе по VBA (см. ниже).

2. Средствами VBA согласно вашему варианту вычислите значения выражения (см. приложение).

3. По выполненной работе составить отчет в электронной форме (и распечатать), куда включить титульный лист, на последующих страницах поместить содержание задания и результаты выполненного задания. Для иллюстрации выполненного задания используйте скриншот.

**Пример.** Вычислить значение выражения

$$y(x) = \frac{\sqrt{a \cdot b \cdot c}}{2.4} - \frac{0.7 \cdot a \cdot b \cdot c \cdot x}{(a + b)^2} - e^{a+b}$$

при  $c=1.5$ ,  $d=0.6$ ,  $x=0.64$ .

**Решение.**

Программа и результат расчетов представлены на рис. 1.



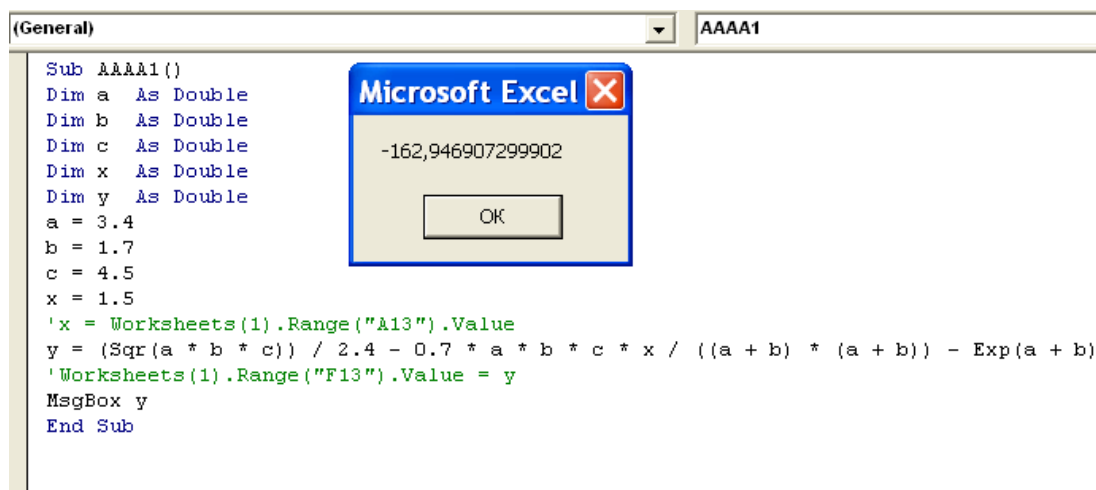


Рис. 1. Листинг программы для расчета значения выражения с результатом расчета

**Замечание:** первый символ строки:

```
'x=Worksheets(1).Range("A13").Value
```

есть апостроф ('). После символа ' (см. на клавиатуре клавишу с буквой Э) записываются комментарии, не участвующие в работе программы.

**Задача 2. Вычислить значения функции, заданной различными аналитическими выражениями, при заданных значениях аргумента (разветвляющийся алгоритм).**

**Цель работы:** составить программу для вычисления значения функции  $u(x)$  при различных условиях.

**Порядок работы**

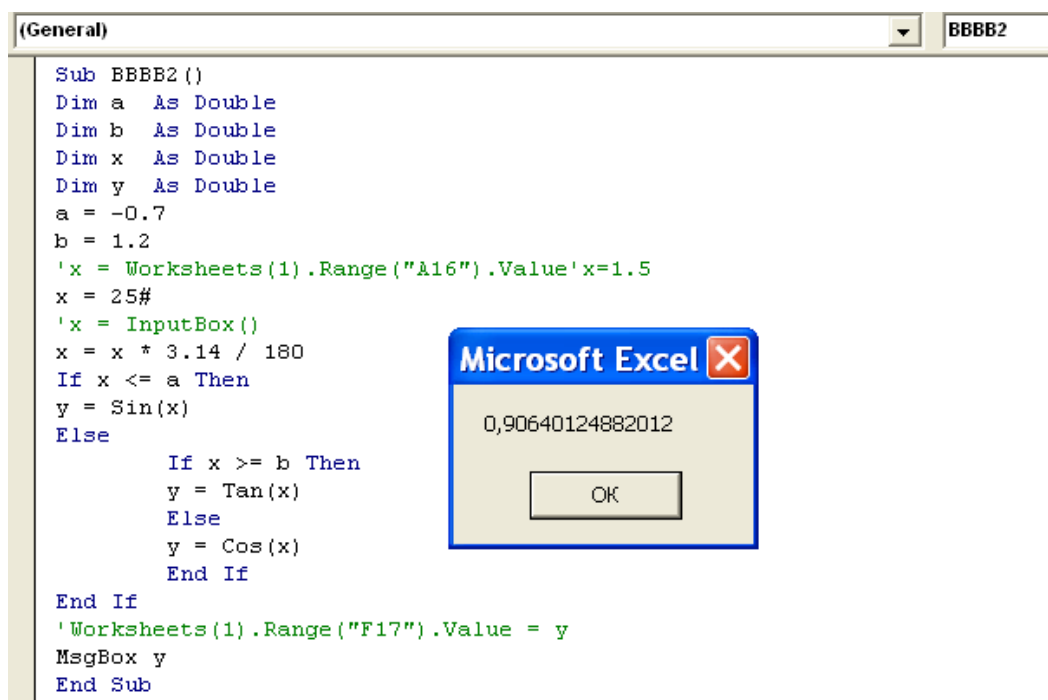
1. Изучите методические указания к лабораторной работе по VBA (см. ниже).
2. Средствами VBA согласно вашему варианту вычислите значения выражения при трех различных значениях  $x$  из разных промежутков (см. приложение).
3. По выполненной работе составить отчет в электронной форме (и распечатать), куда включить титульный лист, на последующих страницах поместить содержание задания и результаты выполненного задания. Для иллюстрации выполненного задания используйте скриншот.

**Пример.** Вычислим значение функции:

$$y(x) = \begin{cases} \sin x, & x \leq a, \\ \cos x, & a < x < b, \\ \operatorname{tg} x, & x \geq b, \end{cases}$$

$a = -0.7, b = 1.7.$

Указанное выражение может быть запрограммировано в виде кода, представленного на рис. 2 или рис. 3.



The image shows a screenshot of the Microsoft Excel VBA editor. The window title is "Microsoft Excel" and the active sheet is "BBBB2". The VBA code in the editor is as follows:

```
Sub BBBB2 ()
  Dim a As Double
  Dim b As Double
  Dim x As Double
  Dim y As Double
  a = -0.7
  b = 1.2
  'x = Worksheets(1).Range("A16").Value 'x=1.5
  x = 25#
  'x = InputBox()
  x = x * 3.14 / 180
  If x <= a Then
    y = Sin(x)
  Else
    If x >= b Then
      y = Tan(x)
    Else
      y = Cos(x)
    End If
  End If
  'Worksheets(1).Range("F17").Value = y
  MsgBox y
End Sub
```

Overlaid on the code editor is a message box titled "Microsoft Excel" with the text "0,90640124882012" and an "OK" button.

Рис. 2. Листинг программы для расчета значения функции  $y(x)$  при различных условиях с результатом расчета (вариант 1)

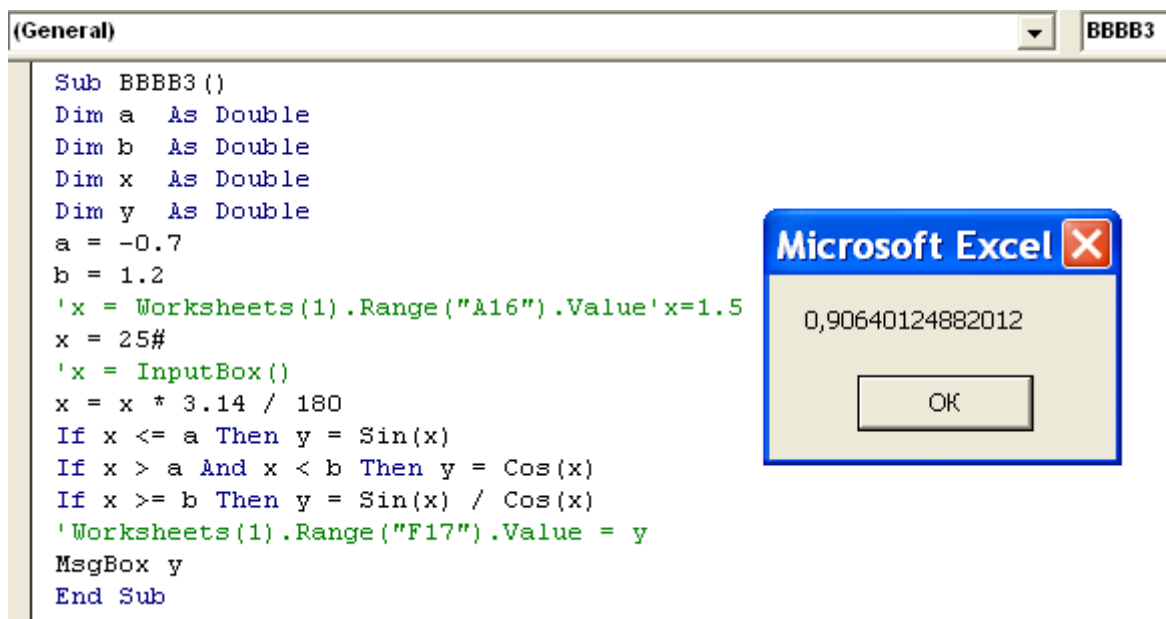


Рис. 3. Листинг программы для расчета значения функции  $y(x)$  при различных условиях с результатом расчета (вариант 2)

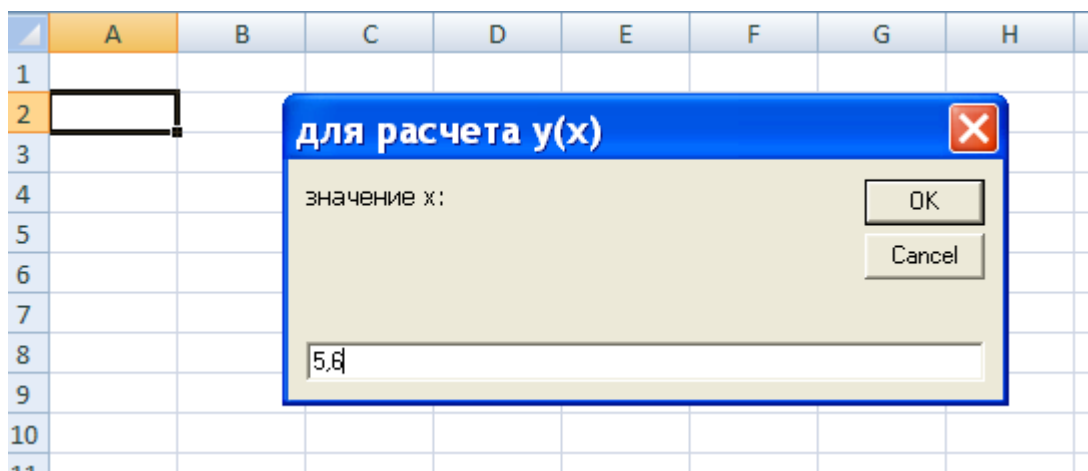


Рис. 4. Результат выполнения оператора ввода  $x = \text{InputBox}(\text{"значение x"}, \text{"для расчета } y(x)\text{"})$

**Задача 3. Вычислить таблицы значений функции, заданной аналитически, при известных начальном и конечном значениях аргумента и шаге его изменения.**

**Цель работы:** составить программу для табулирования функции  $y(x)$  с использованием цикла посредством оператора *while – wend u for – next*.

## Порядок работы

1. Изучите методические указания к лабораторной работе по *VBA* (Введение в *VBA*)

2. Средствами *VBA* согласно вашему варианту выполните задание лабораторной работы «Табулирование функции средствами *VBA*» (см. приложение).

**Задание.** Вычислить таблицы значений функции, заданной аналитически, при известных начальном и конечном значениях аргумента и шаге его изменения (циклический алгоритм). Результаты значений  $x$  и  $y(x)$  вывести на лист *Excel*. Это задание выполнить двумя способами, используя управляющую структуру *while – wend* (первая программа), используя управляющую структуру *for – next* (вторая программа).

**Примечание.** Реализация задачи вторым способом является не обязательным дополнительным заданием.

3. По выполненной работе составить краткий отчет в электронной форме, куда включить титульный лист, на последующих страницах поместить содержание заданий и результаты выполненных заданий. Для иллюстрации выполненного задания используйте скриншот.

**Пример.** Вычислить и вывести таблицу значений функции  $y(x)=e^x$  при  $x$ , изменяющемся от  $x_0=1.5$  до  $x_k=4.7$  с шагом  $dx=0.5$  на лист *Excel*. При написании кода использовать управляющую структуру *while – wend*.

Листинг программы, использующий управляющую структуру *while – wend*, представлен на рис. 5 (пример первой программы). Примеры программ, использующие управляющую структуру *for – next* см. методические указания ниже.

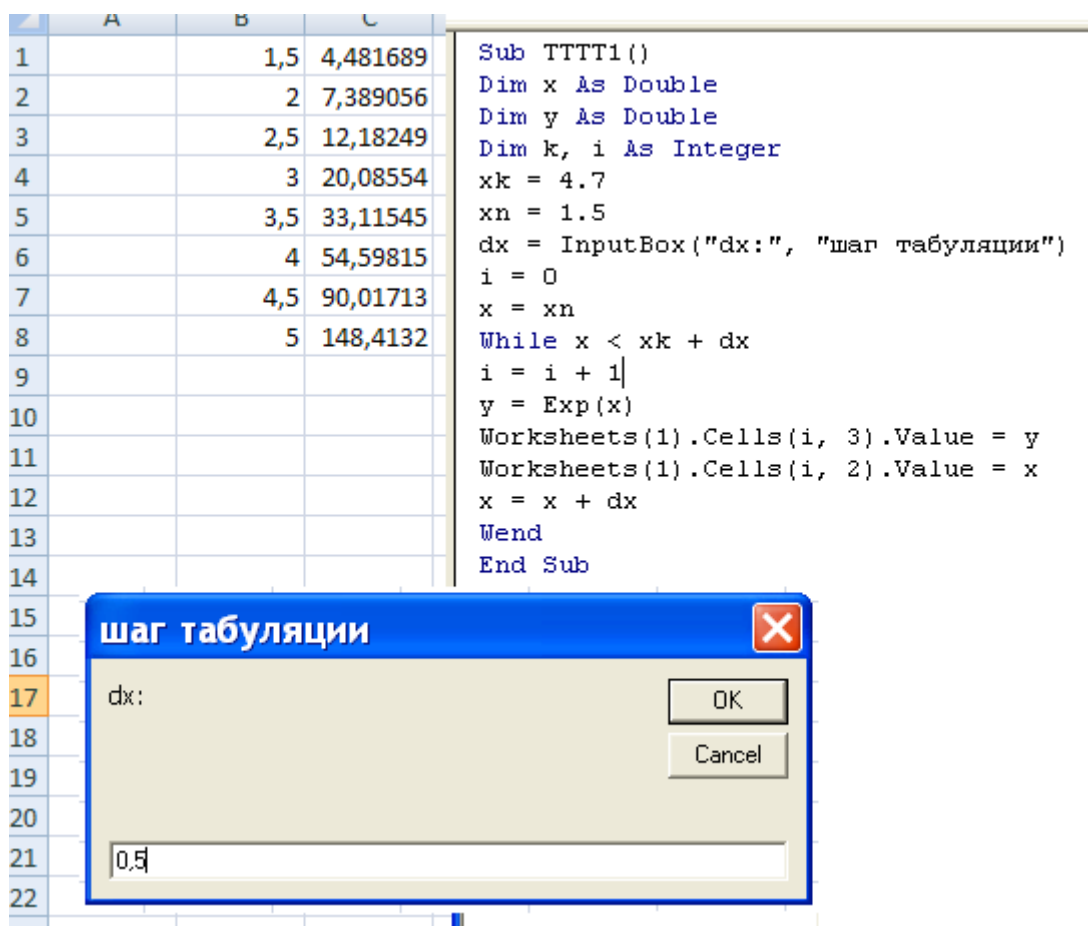


Рис. 5. Результат решения примера на тему «Табулирование функции»

### Методические указания. Введение в VBA

#### Создание приложений на языке *Visual Basic for Applications (VBA)*

*Visual Basic for Applications (VBA)* — среда визуального объектно-ориентированного программирования для создания прикладных программ в среде *Microsoft Office*.

С помощью VBA:

- создаются объекты управления графического интерфейса пользователя;
- задаются и изменяются свойства объектов;
- подключается соответствующий программный код.

Для написания программного кода используется редактор *Visual Basic Editor (VBE)*.

Если *Microsoft Excel* версия 2010, то в приложении 3 представляется информация «Путь к обращению «Редактор *Visual Basic*»».

Для написания программного кода используется редактор *Visual Basic Editor (VBE)*, версия *Microsoft Excel 2003*. Начало работы с редактором *VBE: Microsoft Excel /Сервис/ Макрос/ Редактор Visual Basic* (рис. 6).

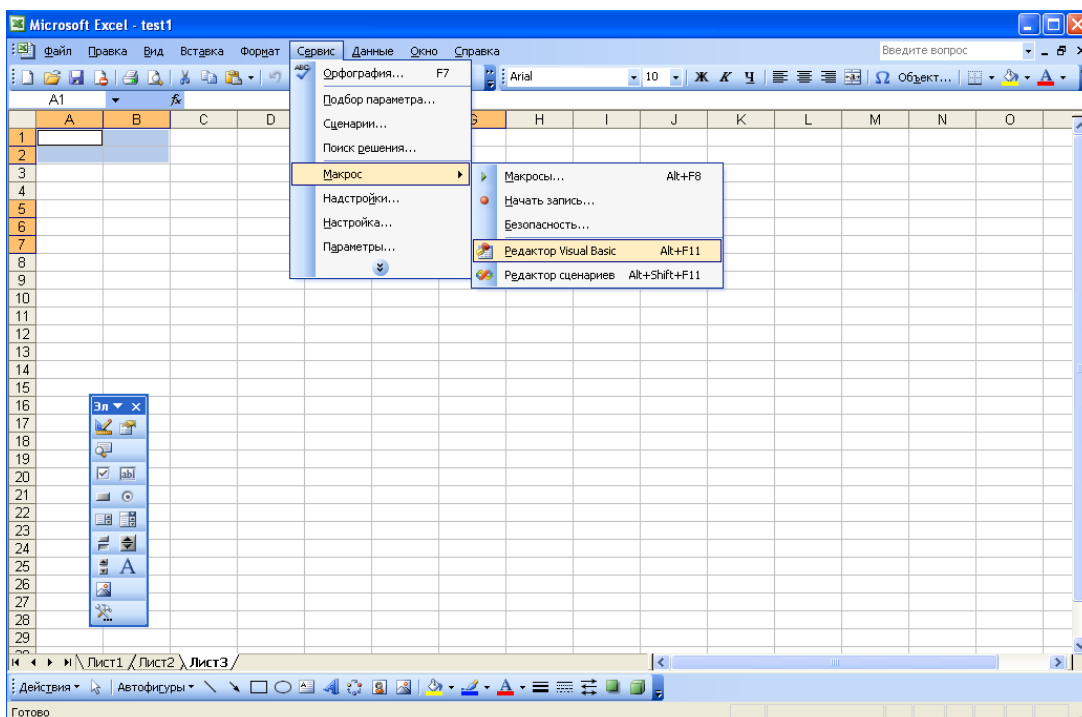


Рис. 6. Начало работы с редактором *VBE* в среде *Microsoft Excel*

Создание программного модуля в составе рабочей книги: *Insert / Module* (рис. 7).

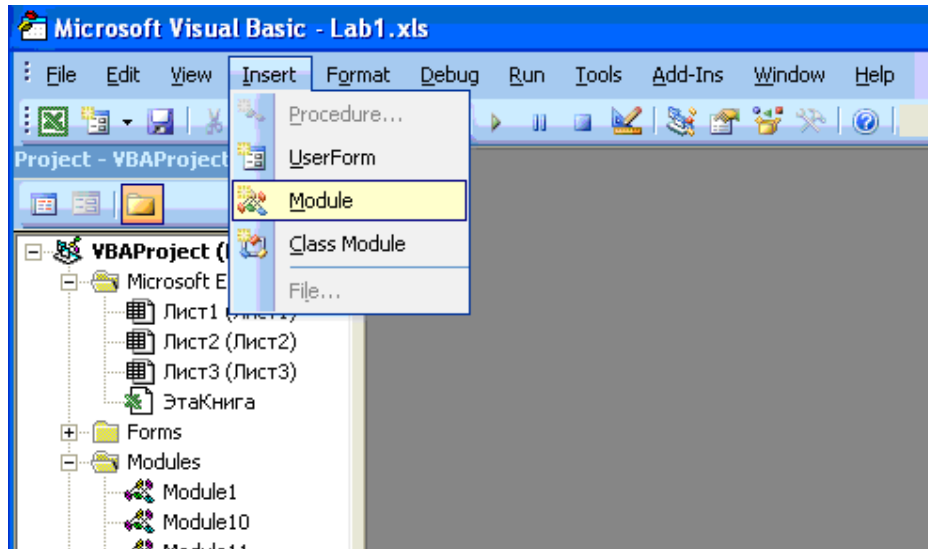


Рис. 7. Создание программного модуля в среде *Microsoft Excel*  
 При этом создается новый рабочий лист со стандартным именем *Module* и порядковым номером программного модуля в рабочей книге.

### Среда разработки программ

Основными пунктами меню, необходимыми для запуска и отладки программы являются:

- Текст (код) программы набирают в окне редактора.
- *Run/Run Sub* – запуск программы;
- *Run/Break* – приостановка выполнения программы;
- *Run/Reset* – прерывание выполнения программы;
- *Debug/Step Into* – выполнить очередной оператор программы;
- *Debug/Step Over* – выполнить оператор программы без входа в процедуры;
- *Debug/Toggle BreakPoint* – установить точку прерывания;
- *Debug/Clear All BreakPoints* – очистить все точки прерывания.

### Интерфейс редактора *VBE*

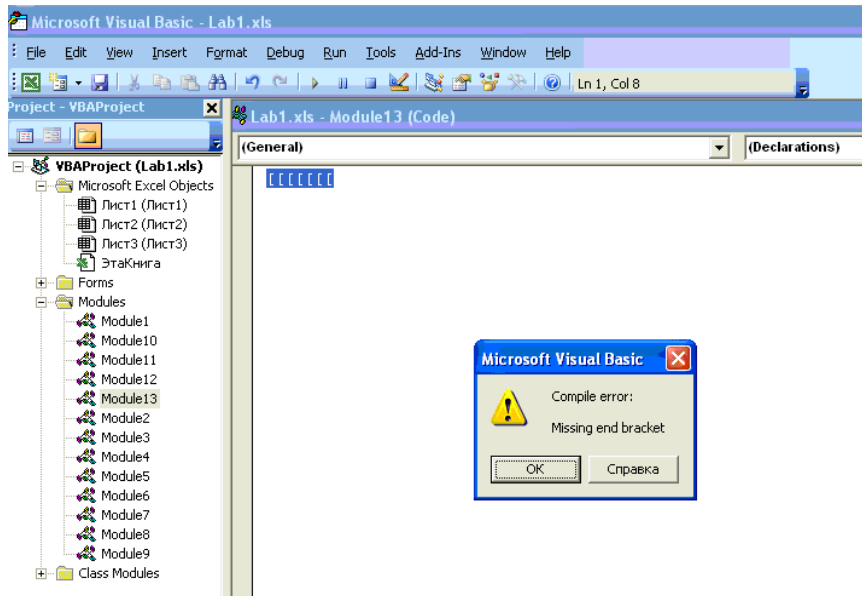


Рис. 8. Интерфейс редактора VBE в среде *Microsoft Excel*, иллюстрирующий наличие ошибки

Если в программе обнаружены ошибки, их необходимо исправить, написать правильно код программы.

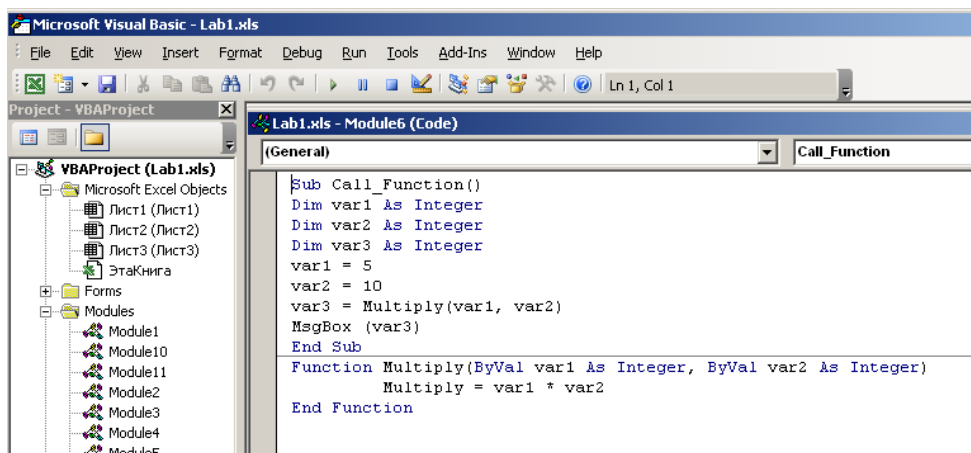


Рис. 9. Интерфейс редактора VBE в среде *Microsoft Excel* с правильным написанием программного кода

Используя главное меню, выполняем последовательность действий: **Debug/ Compile VBAProject**. Чтобы отправить на счет, выполняем последовательность действий: **Run / Run Sub/UserForm**



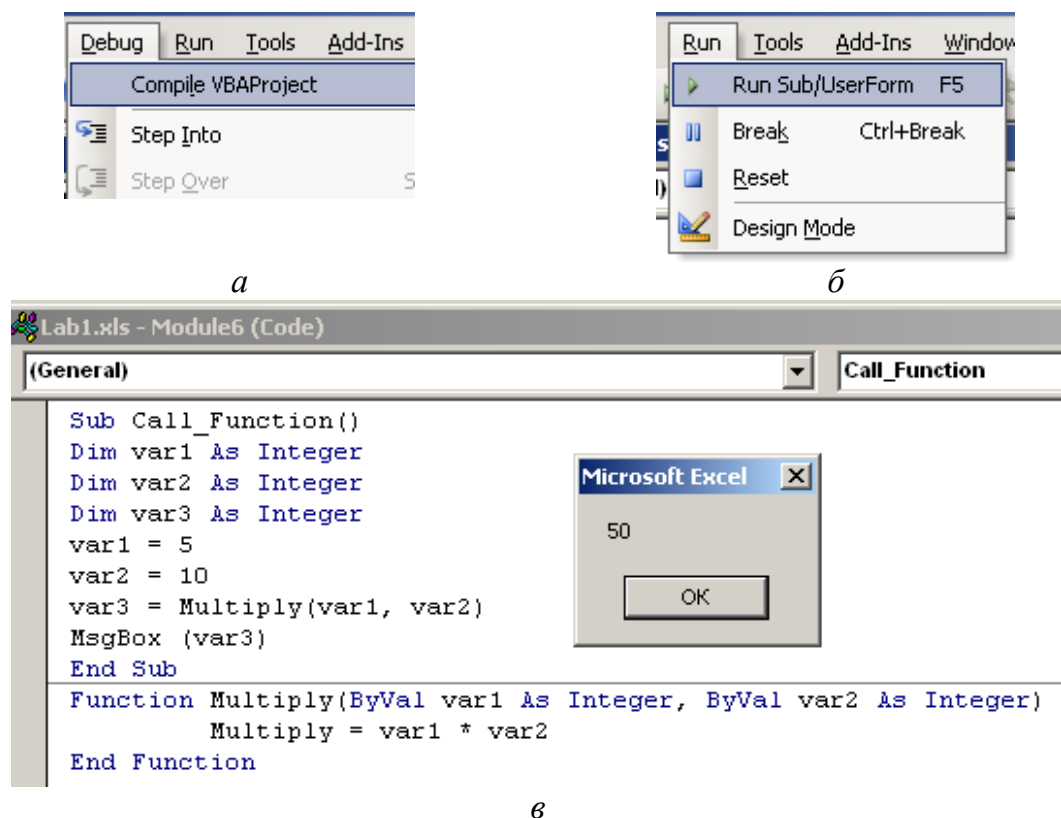


Рис. 10. Рекомендуемая последовательность действий готовой программы для запуска на счет: *а* – шаг 1, *б* – шаг 2, *в*– программа с результатом расчетов

Появится окно со списком наименований программ, выбираем нужную, затем нажимаем кнопку «**Run**». Получаем результат, например, такой как иллюстрируется на рис. 18 *в*.

### Структура программного кода

Программа на языке *VBA* имеет модульную структуру, в составе которого вложенные модули, содержащие одну или более вложенных процедур. Каждая переменная имеет сферу действия (уровень видимости). В *VBA* переменные имеют три уровня видимости:

- уровень процедуры;
- уровень модуля;
- уровень проекта (общий уровень)

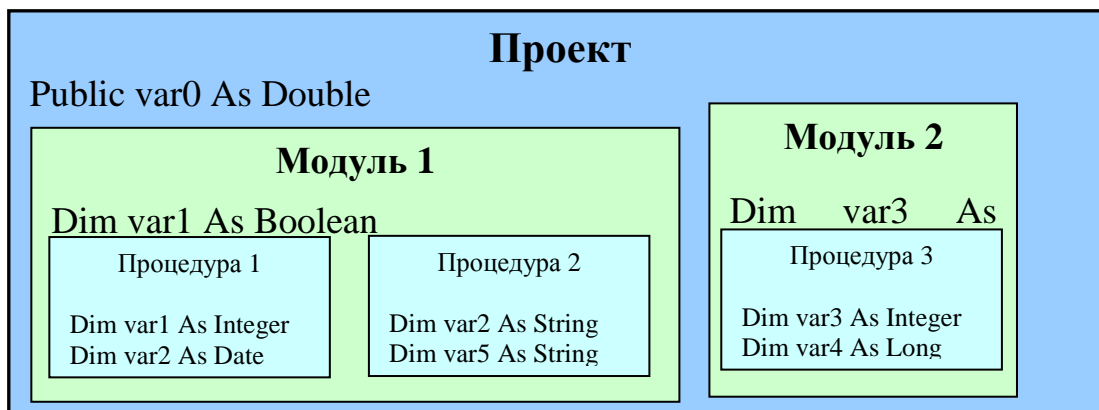


Рис. 11. Пример общей структуры программы

### Процедуры и функции

Программные модули *VBA* состоят из одной или более подпрограмм (процедур или функций). Процедура – минимальный модуль в составе прикладной программы на языке *VBA*. Процедуры имеют стандартное оформление:

```
Sub <имя_процедуры> (аргументы)
    тело процедуры (операторы)
End Sub
```

Оператор **Sub** - объявление процедуры, задается имя, указывается состав аргументов, передаваемых при вызове процедуры из программы. Каждому оператору **Sub** обязательно соответствует **End Sub**.

Функция – отличается от процедуры следующим:

- начинается ключевым словом **Function** и заканчивается ключевыми словами **End Function**;
- функцию можно вызвать из формулы, введенной в ячейку;
- функция может возвращать значение в вызывающую программу или формулу.

Вызов процедуры общего назначения выполняется по имени:

```
<Имя>( <Фактические аргументы>)
```

При вызове процедуры фактические аргументы подставляются на место формальных и управление выполнением передается процедуре. Аргументы могут быть входными, выходными или модифицируемыми. Через входные аргументы процедура получает данные при обращении к ней. Выходные аргументы возвращают результаты выполнения процедуры. Модифицируемые аргументы являются одновременно входными и выходными.

Функция общего назначения построена также как процедура, однако, результат работы функции передается (возвращается) через ее имя. Поэтому, как и в математике, обращения к функциям можно использовать внутри арифметических и логических выражений.

### Область видимости процедур и функций

У процедур и функций два уровня видимости:

- уровень модуля;
- уровень проекта.

Служебные слова *Private* и *Public* задают область видимости процедур и функций. *Private* делает объект доступным только внутри данного модуля. *Public* делает объект доступным из другого модуля. Для того чтобы перевести процедуру или функцию на уровень модуля, необходимо объявить ее с ключевым словом *Private*. Такие программы не могут запускаться сами по себе. Их можно только вызвать из других программ. Рассмотрим пример.

```
Sub Proc()  
    var1=GetRand  
    ModuleLevelProcedure var1  
End Sub
```

```
Private Sub ModuleLevelProcedure(ByVal var1)  
    MsgBox var1*GetRand  
End Sub
```

```
Private Function GetRand()  
    GetRand=Int(100*Rnd())  
End Function
```

Сначала процедура *Proc()* вызывает функцию *GetRand* для генерации случайного целого числа в диапазоне от 0 до 99, а затем передает его в процедуру *ModuleLevelProcedure*. Там этот аргумент умножается на результат еще одного обращения к функции *GetRand*. Их произведение выводится в информационном окне.

### Переменные

Переменная – поименованная область в памяти компьютера во время выполнения программы. Переменная предназначена для хранения и изменения значений во время выполнения программы. Переменная требует явного объявления своего имени. Имя образуется из алфавитно-

цифровых символов и знака подчеркивания `_`. Имя всегда начинается с буквы и представляет собой непрерывную последовательность символов, но не более 254; пробелы в имени не допускаются.

Итак, в языке *Visual Basic* действуют следующие соглашения на имена процедур, переменных и констант:

- должны начинаться с буквы;
- могут включать буквы, цифры и символы подчеркивания;
- не должны включать знаки препинания или пробелы;
- не должны совпадать с ключевыми словами языка *Visual Basic*.

### Описание переменных

Модуль, тело процедуры или функции обычно начинаются с раздела описаний. Он содержит определения переменных и констант, которые используются в модуле и процедурах. С помощью переменных в процедуры передаются аргументы, в ходе выполнения процедур сохраняются рабочие промежуточные значения, осуществляется обмен данными между процедурами. Переменные существуют только внутри модулей, процедур или функций. Каждая переменная имеет имя. Основной инструкцией для явного описания переменных является инструкция ***Dim***. При определении переменной для нее указывается тип данных.

Формат инструкции описания переменной:

***Dim*** <Имя переменной> [***As*** <Тип данных>]

Следующая инструкция создает переменную *x* и указывает для нее текстовый (строковый) тип данных ***String: Dim x As String***. Если разместить данную инструкцию внутри процедуры, то переменная *x* может быть использована только внутри этой процедуры. Если поместить данную инструкцию в раздел описаний модуля, то переменная *x* будет доступна для любых процедур в данном модуле, но недоступна для процедур в других модулях. Для того чтобы сделать данную переменную доступной для всех процедур в базе данных, следует описать ее как общую с помощью инструкции ***Public: Public x As String***

Основными типами данных, используемыми при описании переменных, являются:

***Integer*** – целое число (2 байта) (например, ***Dim X As Integer, Y As Integer, Z As Integer***);

- ***Long*** – длинное целое число (4 байта);
- ***Single*** – десятичное число одинарной точности (4 байта);
- ***Double*** – десятичное число двойной точности (8 байтов);
- ***Currency*** – десятичное число с фиксированной точкой (8 байтов);

- **String** – строка текста (до 65400 символов);
- **Byte** – целое от 0 до 255 (1 байт);
- **Boolean** – логическое значение **True** или **False** (2 байта);
- **Date** – дата и время (8 байтов);
- **Object** – экземпляр класса (4 байта);
- **Variant** – любой из перечисленных выше типов (16 байтов + 1 байт/символ). Указание типа данных в инструкции описания не является обязательным. Если тип данных не указан, по умолчанию переменная получит тип **Variant**.

## Массивы

### Объявление массива

- Локальный массив:

**Dim** <имя массива> (размерность) **As** <тип элементов массива>

- Глобальный массив:

**Public** <имя массива> (размерность) **As** <тип элементов массива>

### Примеры объявления массива

Одномерный массив, состоящий из 10 элементов –

```
Dim NumberArray(10) As Integer
```

2-х мерный массив, состоящий из  $10 \times 20 = 200$  элементов –

```
Dim TableArray(10,20) As String
```

3-х мерный массив –

```
Dim BigArray(5,50,100) As Variant
```

Нумерация элементов массива задается инструкцией **Option Base** в начале модуля: **Option Base 0** или **Option Base 1**. По умолчанию нумерация элементов массива начинается с 0.

### Использование массива

Пример использования массива типа **Integer**:

```
Option Base 1
```

---

```
Sub UsingArray()  
    Dim Vals(3) As Integer  
    Vals(1)=Int(100*Rnd())  
    Vals(2)=Int(100*Rnd())  
    Vals(3)=Int(100*Rnd())
```

```

    MsgBox "Lottery numbers: " & Vals(1) & ", "
& Vals(2) & ", " & Vals(3)
End Sub

```

Имеются ряд специальных функций для работы с массивами.

### Примеры функций для работы с массивами

1. Функция *Array* – позволяет создавать массив в ходе выполнения программы, без предварительного описания.

Пример:

```

Sub CreateArray()
    DataCA=Array("Kate",43,#4/15/1962#)
    MsgBox DataCA(0) & ",age" & DataCA(1) & ", born"
    & DataCA(2)
End Sub

```

2. Функция *Erase* – используется для удаления данных, хранимых в элементах массива. Если массив фиксированного размера – очищается содержимое массива, а память, выделенная массиву, остается за ним.

Пример:

#### Option Base 1

---

```

Sub EraseArray()
    Dim EA(2) As Integer
    EA(1) = Int(100 * Rnd())
    EA(2) = Int(100 * Rnd())
    MsgBox "Lottery numbers: " & EA(1) & ", " &
EA(2)
    Erase EA
    MsgBox "Lottery numbers: " & EA(1) & ", " &
EA(2)
End Sub

```

3. Функция *IsArray* – позволяет проверить, является ли переменная массивом. Функция имеет один аргумент и возвращает *True*, если переменная является массивом, и *False*, если переменная не массив. Функция полезна, если необходимо проверить, возвращает ли вызываемая функция массив или обычную переменную.

Пример:

```
Sub IsArrayFunction()  
    Dim Arr(2) As Integer  
    Dim ArrayBool As Boolean  
    ArrayBool = IsArray(Arr)  
    If ArrayBool = True Then  
        MsgBox "Arr is an array."  
    End If  
End Sub
```

4. Функции *LBound* и *UBound* – определяют верхнюю и нижнюю границы индексов элементов массива.

Пример:

```
Sub LBoundAndUBound()  
    Dim Data(4 To 15) As Integer  
    MsgBox "The lower bound is " & LBound(Data) &  
    ". "  
    MsgBox "The upper bound is " & UBound(Data) &  
    ". "  
End Sub
```

Перед вызовом функций *LBound* и *UBound* рекомендуется проверить переменную функцией *IsArray*.

### Передача данных при вызове подпрограммы

Передача аргументов из одной программы в другую осуществляется двумя способами:

- по ссылке (*by reference*);

- по значению (*by value*).

Выбор способа передачи – с помощью ключевых слов *ByRef* и *ByVal*:

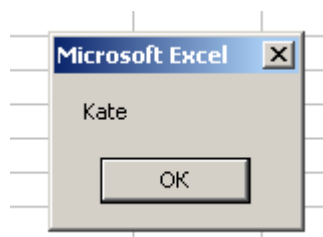
по ссылке – передается сама переменная (имя переменной), поэтому ее значение в подпрограмме можно изменить;

по значению – передается только значение переменной; изменить это значение в вызванной подпрограмме нельзя.

При отсутствии в описании ключевых слов передача осуществляется по ссылке.

### Пример передачи переменной по ссылке

```
Sub PassArgumentByReference ()
    Dim Username As String
    Username = "Mike"
    ChangeName Username
    MsgBox Username
End Sub
```




---

```
Sub ChangeName (ByRef Username)
    Username = "Kate"
End Sub
```

В процедуре *PassArgumentByReference()* переменная *Username* передается в процедуру *ChangeName* по ссылке. Это означает, что в *ChangeName* используется имя переменной *Username*, а именно – переменной *Username* присваивается новое значение, которое затем возвращается в процедуру.

### Пример передачи переменной по значению

```
Sub PassArgumentByVal ()
    Dim Username As String
    Username="Mike"
    ChangeName Username

    MsgBox Username
End Sub
```

*Mike*

---

```
Sub ChangeName (ByVal Username)
    Username="Kate"
```



**End Sub**

### Операторы, используемые в выражениях

Таблица 1

Оператор	Пример использования	Описание
+	$a + b$	Сложение двух чисел
-	$a - b$	Вычитание
*	$a * b$	Умножение
/	$a / b$	Деление
\	$a \setminus b$	Целочисленное деление
Mod	$a \text{ Mod } b$	Возвращается остаток от деления
^	$a ^ b$	Возведение в степень
And	выражение1 And выражение2	Логическое “И”
=	выражение1 = выражение2	Оператор эквивалентности
>	выражение1 > выражение2	Оператор сравнения “больше”
>=	выражение1 >= выражение2	Оператор сравнения “больше или равно”
<>	выражение1 <> выражение2	Оператор сравнения “не равно”
<	выражение1 < выражение2	Оператор сравнения “меньше”
<=	выражение1 <= выражение2	Оператор сравнения “меньше или равно”

### Основные математические функции VBA

Таблица 2

Функция	Пример использования	Описание	Тип возвращаемого результата
Abs	Abs(x)	Модуль числа x	Совпадает с

			ТИПОМ числа x
Atn	Atn(x)	Арктангенс числа x	Double
Cos	Cos(x)	Косинус числа x	Double
Exp	Exp(x)	Экспоненциальная функция (в степени x)	Double
Fix	Fix(x)	Возвращает целую часть числа x. Если x - отрицательное, то возвращаемое значение округляется в <u>большую</u> сторону (напр., если x=-1.5, возвращается -1).	Integer
Int	Int(x)	Возвращает целую часть числа x. Если x - отрицательное, то возвращаемое значение округляется в <u>меньшую</u> сторону (напр., если x=-1.5, возвращается -2).	Integer
Log	Log(x)	Натуральный логарифм числа x	Double
Rnd	Rnd или Rnd(x)	Генерируется случайное число. Если аргумент не указан – генерируется число в диапазоне от 0 до 1	Single
Sgn	Sgn(x)	Знак числа (1 0 -1)	Integer
Sin	Sin(x)	Синус числа x	Double
Sqr	Sqr(x)	Квадратный корень из x	Double
Tan	Tan(x)	Тангенс числа x	Double

**Дополнение к основным математическим функциям VBA:**

- $\text{Log}_n(x) = \text{Log}(x) / \text{Log}(n)$
- Для перевода x из градусов в радианы:  $x = x * \pi / 180$

- **IsNumeric** () – встроенная функция VBA – возвращает *True*, если ее аргумент является числом (строка), и *False* – в противном случае.
- **Val** () – встроенная функция VBA - преобразует переданную ей строку в число.
- **InputBox** () – выдает запрос на ввод значения; возвращает строковое значение.

### Примеры ввода и вывода значений

#### Ввод:

- `x=5.7` (оператор присваивания);
- `x = Worksheets(1).Range("A1").Value` (значение считывается из ячейки A1);
- `x = InputBox("Enter Number: ", "Calculate Factorial")` (создается окно ввода значений переменной *x* и указываются соответствующие комментарии).

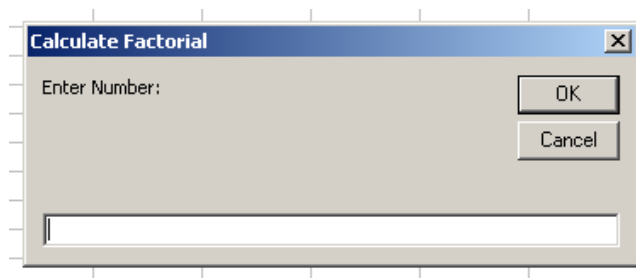


Рис. 12. Окно ввода значений

#### Вывод:

- `Cells(1,2).Value=5` (присваивает ячейке B1 текущего рабочего листа активной рабочей книги значение 5);
- `Worksheets(1).Range("A1:B2").Value = 10` (присваивание блоку ячеек значения, равного 10);
- `Range("D15").Value="Test"` (присваивание ячейке текущего рабочего листа активной рабочей книги значения *Test*);
- `MsgBox (x)` (создание окна сообщений, пример результата см. рис. 13).

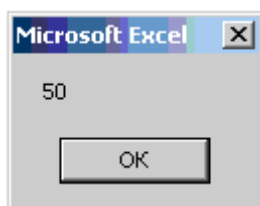


Рис. 13. Результат выполнения оператора вывода MsgBox (x)

### Примеры на использование обращений к функции и процедуре

Пример 1:

```

Sub Call_Function()
    Dim var1 As Integer
    Dim var2 As Integer
    Dim var3 As Integer
    var1 = 5
    var2 = 10
    var3 = Multiply(var1,var2)
    MsgBox (var3)
End Sub

```

---

```

Function Multiply(ByVal var1 As Integer, ByVal
var2 As Integer)
    Multiply = var1 * var2
End Function

```

### Значение строк программы (пример 1)

Таблица 3

<i>Строка программы</i>	<i>Назначение строки программы</i>
<b>Sub</b> Call_Function()	начало процедуры с указанием имени процедуры
<b>Dim</b> var1 <b>As Integer</b> <b>Dim</b> var2 <b>As Integer</b> <b>Dim</b> var3 <b>As Integer</b>	блок описания (представляется переменная с именем <i>var1</i> и указывается тип переменной <i>var1</i> как целочисленный)
var1 = 5 var2 = 10	блок ввода (присваивается

	переменной <i>var1</i> значение 5)
<code>var3 = Multiply(var1, var2)</code>	переменной по имени <i>var3</i> присваивается значение. Причем это значение появляется в результате вызова функции <i>Multiply</i> и передачи ей значений переменных <i>var1</i> , <i>var2</i> , равных соответственно 5 и 10
<code>MsgBox (var3)</code>	оператор вывода (вызывается окно сообщений, где представляется значение переменной <i>var3</i> )
<code>End Sub</code>	конец процедуры
<hr/>	граница между процедурой и функцией одного модуля
<code>Function Multiply(ByVal var1 As Integer, ByVal var2 As Integer)</code>	начало функции с указанием имени функции ( <i>Multiply</i> ), в скобках указывается способ передачи значений переменных ( <i>ByVal</i> – по значению, имя переменной – <i>var1</i> , тип переменной – <i>As Integer</i> ; подобным образом, через запятую, указывается информацию о другой переменной – <i>var2</i> )
<code>Multiply = var1 * var2</code>	Результат умножения значений переменных <i>var1</i> на <i>var2</i> является значением функции. Внимание: имя функции

	и имя объекта, куда записывается расчетное значение функции – одинаковое.
<b>End Function</b>	Конец функции

Пример 2:

```

Sub DDDD ()
Dim x As Double
Dim y As Double
Dim z As Double
x = 1.1
y = 2.2
MsgBox (x)
MsgBox (y)
TTTT x, y

```

x=1.1

y=2.2

*вызываем процедуру по имени TTTT и передаем значения переменных x и y*

```

MsgBox (y)
MsgBox (x)
z= FFF(x)
MsgBox (z)
End Sub

```

y=3.3

x=1.1

---

**Function** FFF(**ByVal** x **As** Double)

FFF = 1/x^2

$$FFF = \frac{1}{x^2} \xrightarrow{x=1.1} \frac{1}{1.1^2}$$

**End Function**

---

**Sub** TTTT(**ByVal** x **As** Double, **ByRef** y **As** Double)

*процедуре TTTT передаются значения x=1.1 «по значению» (способные не изменяться при выходе из процедуры), y=2.2 «по ссылке» (способные изменяться при выходе из процедуры)*

```

MsgBox (y)
y=x+y

```

y=2.2

y=1.1+2.2=3.3

MsgBox (y)

y=3.3

x=5.5

MsgBox (x)

x=5.5

End Sub

## Управляющие структуры

Управляющие структуры предназначены для изменения порядка выполнения инструкций.

**If - Then - Else** – выполняет группу инструкций, если соблюдено некоторое условие;

**For - Next** – выполняет группу инструкций заданное число раз;

**While - Wend** – выполняет группу инструкций, пока соблюдается некоторое условие;

**Do - Loop** – выполняет группу инструкций, пока соблюдается или не соблюдается некоторое условие;

**Select Case** – в зависимости от значения некоторой переменной или результата проверки условия выполняет одну или несколько возможных групп инструкций;

**For - Each - Next** – выполняет действия над каждым объектом семейства или элементом массива.

## Примеры использования некоторых управляющих структур

### Управляющая инструкция **If - Then - Else**

Условная инструкция **If - Then - Else** изменяет порядок выполнения инструкций в зависимости от результатов проверки некоторого условия.

Пример.

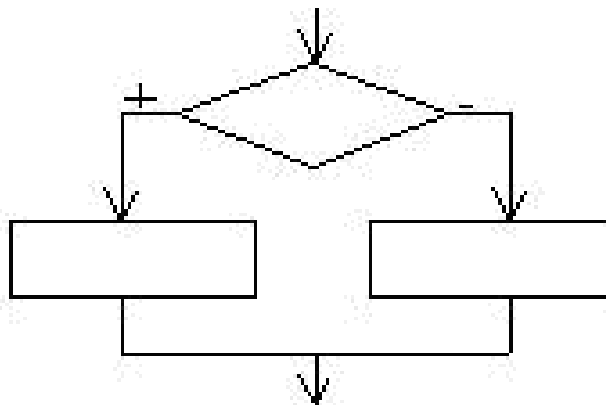
$$y = \begin{cases} \cos(x), & x < 7, \\ e^x, & x \geq 7. \end{cases}$$

### Решение (способ 1).

```

Sub AAAAA()
  Dim x As Double
  Dim y As Double
  x=1.5
  y=0
  If x >= 7 Then
y=exp(x)
  Else y=cos(x)
  End If
End Sub

```

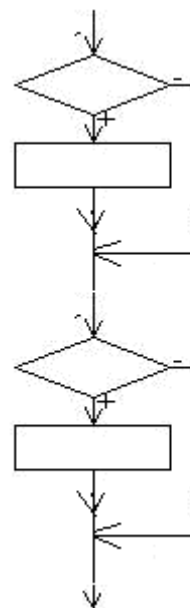


### Решение (способ 2).

```

Sub AAAAA()
  Dim x As Double
  Dim y As Double
  x=1.5
  If x < 7 Then y=cos(x)
  If x >= 7 Then y=exp(x)
  MsgBox y
End Sub

```



### Управляющая инструкция *For - Next*

Управляющая инструкция *For - Next* позволяет выполнять несколько команд заданное число раз.

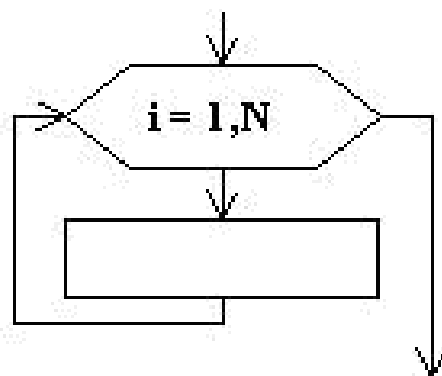
Пример.

$$\sum_{i=1}^5 V_i = S, \quad V_i = i, \quad S = ?$$



### Option Base 1

```
Sub BBBB()  
    Dim V(5) As DOUBLE  
    Dim S As DOUBLE  
    Dim i As Integer  
    Dim n As Integer  
    n=5  
    S = 0  
    For i = 1 To n Step 1  
        V(i) =i  
        S = S + V(i)  
        MsgBox V(i)  
    Next  
    MsgBox S  
End Sub
```



*Step* – любое целое число, определяющее шаг приращения счетчика.

**Пример.** Дан массив  $B$ , содержащий 15 элементов. Указанный массив сформирован на основе встроенной функции  $Rnd()$ . Вывести массив  $B$  и элементы массива  $B$ , которые стоят на четных местах и превышают по абсолютному значению 2.5. Записать в массив  $T$  найденные элементы. Вывести массив  $T$ .

**Решение.** Рабочая область листа и программный код показаны на рис. 14 и 15.

```

(General) CommandButton1_Click

Private Sub CommandButton1_Click()
Worksheets(1).Range("A1:C15").ClearContents
Dim arrb(15) As Double
Dim arrt(15) As Double
Dim k, i As Integer
k = 0
For Count = 1 To 15 Step 1
Randomize
arrb(Count) = 10 * Rnd()
Worksheets(1).Cells(Count, 1).Value = arrb(Count)
If (Count Mod 2 = 0) And (Abs(arrb(Count) > 2.5)) Then
Worksheets(1).Cells(Count, 2).Value = arrb(Count)
k = k + 1
arrt(k) = arrb(Count)
End If
Next
For i = 1 To k Step 1
Worksheets(1).Cells(i, 3).Value = arrt(i)
Next
End Sub

```

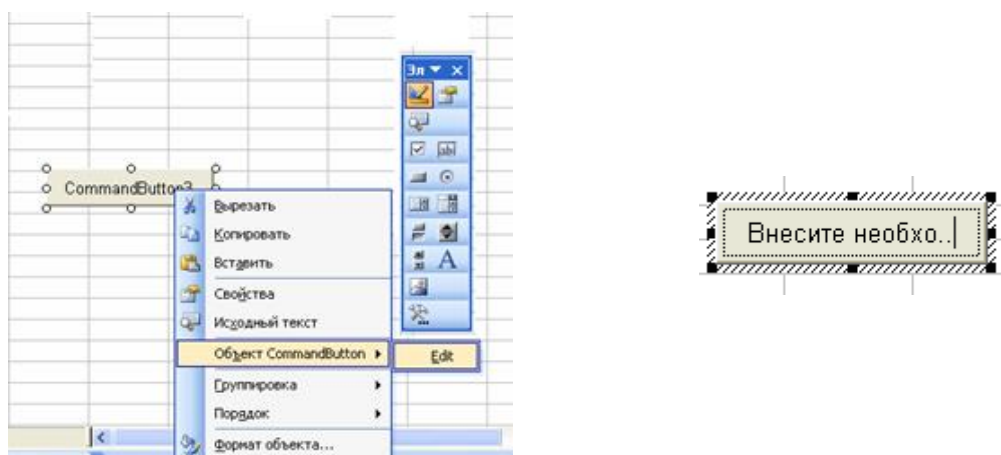
Рис. 14. Программный код к решению примера на тему «Управляющая инструкция *For – Next*»

	A	B	C	D	E	F	G	H	I
1	1,225582		4,167703		Работа с элементами одномерного массива				
2	0,050635		5,66309						
3	3,575475		3,526824						
4	0,345165								
5	4,72451								
6	1,586473								
7	3,656377								
8	4,167703	4,167703							
9	3,42001								
10	5,66309	5,66309							
11	8,933851								
12	1,777359								
13	3,246835								
14	3,526824	3,526824							
15	5,34265								
16									
17									
18									
19									
20									

Рис. 15. Рабочая область листа *Excel* к решению примера на тему «Управляющая инструкция *For – Next*» к рис. 14

Для проведения расчетов использован элемент управления **Кнопка**. Последовательность действий для создания **Кнопки**, как

элемента управления следующая. Выполнить команду **Вид / Панели инструментов / Элементы управления / Кнопка**. Появится кнопка. Затем отредактировать наименование кнопки (выделите кнопку, обратитесь к контекстному меню, нажав правую кнопку мыши, затем внесите изменение в наименование кнопки).



*a*

*б*

Рис. 16. Последовательность действий для редактирования **Кнопки**: *a* – шаг 1, *б* – шаг 2

После создания кнопки появляется окно, где размещается рабочее поле для написания кода программы. Кроме того в рабочем поле появляются строки — начало процедуры (*Private Sub CommandButton1\_Click()*) и конец процедуры. Запускается программа с листа *Excel*, где располагается кнопка управления.

### Управляющая инструкция *While - Wend*

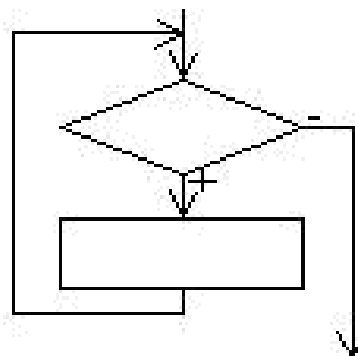
Управляющая инструкция *While - Wend* выполняется до соблюдения определенного условия.

**Пример.** Выделить значения из последовательности случайных чисел, когда  $M = 7$

```

Sub CCCC ()
  Dim M As Integer
  Dim n As Integer
  M = 0
  n=0
  Randomize
  While M <> 7
    M = Int(10 * Rnd())
    n = n + 1
  Wend
  MsgBox "n=" &n
End Sub

```



### Управляющая инструкция *Do – Loop*

Условие завершения цикла может задаваться не только в начале цикла, но и в конце. Условие в конце цикла гарантирует, что он будет выполнен хотя бы один раз. Кроме того, условие можно сделать критерием, как выполнения цикла, так и его завершения.

**Пример.** Выделение определенного значения (равного 7) из последовательности случайных чисел.

Вариант 1.

```

Sub DoWhileLoop()
  Dim Number As Integer
  Number = 0
  Do While Number <> 7
    Number = Int(10 * Rnd())
  Loop
  MsgBox "Your number is " & Number & " ."
End Sub

```

Вариант 2.

```

Sub DoUntilLoop()
  Dim Number As Integer

```

```

Number = 0
Do Until Number = 7
Number = Int(10 * Rnd())
Loop
MsgBox "Your number is " & Number & "."
End Sub

```

### **Управляющая инструкция *For – Each - Next***

Количество повторений операций цикла определяется количеством элементов массива.

Пример 1:

**Option Base 1**

```

Sub StructureFEN()
    Dim Countries(5) As String
    Dim Country As Variant
    Countries(1) = "India"
    Countries(2) = "Peru"
    Countries(3) = "Greece"
    Countries(4) = "Canada"
    Countries(5) = "Kenya"
    For Each Country In Countries
        MsgBox Country
    Next
End Sub

```

## **5. ИТОГОВЫЙ КОНТРОЛЬ**

После завершения изучения дисциплины студенты сдают зачет. Билет для зачета включает два вопроса из приведенного ниже списка. При определении результата зачета учитываются результаты выполненных лабораторных работ и индивидуального домашнего

задания. По итогам успешной защиты индивидуальных заданий студент получает зачет.

## 5.2. Вопросы для подготовки к зачету

1. Системы счисления и формы представления чисел. Единицы измерения объемов информации.

2. Кодирование символьных данных.

3. Функционально структурная организация ПК.

4. Программный комплекс операционной системы *Windows*.

5. Операционная система *MS DOS*. Понятие файла и каталога.

Полное имя файла. Типы файлов

6. Объекты файловой системы в среде *Windows*.

7. Базовые приемы работы в текстовом процессоре *Microsoft Word*

8. Типовая структура интерфейса табличного процессора.

Структура ячейки *MS Excel*

9. Виды вводимой информации в *Excel*.

10. Что включает «Редактирование и Форматирование Листа» в *Excel*?

11. Что такое колонтитулы и как их вставить в документ?

12. Как объединить/разделить ячейки в таблице?

13. Какие встроенные функции *Excel* Вы знаете?

14. Как разместить на одной диаграмме 2 и более графиков?

15. Классификация баз данных. Основные термины и определения.

16. Обобщенная структура компьютерной сети. Элементы сети.

17. Классификация вычислительных сетей.

18. Основные характеристики *WWW*.

19. Структурное проектирование программных продуктов.

20. Объектно-ориентированное проектирование программных продуктов.

21. Ввод/вывод данных с клавиатуры и на экран на *VBA*.

22. Встроенные математические функции на *VBA*.

23. Логические операции на *VBA*.

24. Типы данных в языке *VBA*.

25. Передача данных при вызове подпрограммы по ссылке, по значению.

26. Объявление локального массива, объявление глобального массива. Какой инструкцией задается нумерация элементов массива?

27. Управляющая структура *VBA*: *If - Then - Else*.

28. Управляющая структура *VBA*: *For - Next*.

29. Управляющая структура *VBA*: *While - Wend*.

30. Управляющая структура *VBA*: *Do - Loop*.

31. Управляющая структура *VBA*: *For – Each – Next*.
32. Как открыть окно для набора программы *VBA*?
33. Как вставить командную кнопку на лист?
34. Как вставить программу обработки нажатия на командную кнопку?
35. Как считать значение ячейки листа *Excel* в переменную *VBA*?
36. Как записать значение переменной *VBA* в ячейку листа *Excel*?
37. Как заполнить диапазон ячеек одной строкой на *VBA*?
38. Как очистить содержимое диапазона ячеек листа с помощью *VBA*?
39. Как заполнить массив *VBA* содержимым ячеек на листе *Excel*?

## **6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **6.1. Литература обязательная**

1. Марков Н. Г. Базы данных: учебное пособие / Н. Г. Марков; Томский политехнический университет (ТПУ), Институт дистанционного образования (ИДО). – Томск : Изд-во ТПУ, 2001. – 107с.

2. Демин А.Ю. Компьютерная графика : учебное пособие / А. Ю. Демин, А. В. Кудинов; Томский политехнический университет; Институт дистанционного образования. – Томск : Изд-во ТПУ, 2005. – 164 с.

3. Шефер О.В., Огородов С.В. Информатика. Особенности работы в *Word, Excel, MathCAD*. Методические указания и индивидуальные задания к лабораторным работам / сост. О.В. Шефер, С.В. Огородов. Электронное Изд. ТПУ. 2009. Рег. № 71 от 06.05.09. 68 с.

4. Хамухин А. А. Введение в информатику : учебно-методическое пособие / А. А. Хамухин; Томский политехнический университет (ТПУ) . – 2-е изд. – Томск : Изд-во ТПУ, 2009. – 284 с.

5. Информатика: метод. указ. к выполнению лабораторных работ для студентов ИДО, обучающихся по напр. 280700 «Техносферная безопасность» / сост. О.В. Шефер; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2011.– 52 с.

### **6.2. Литература дополнительная**

6. Леонтьев В.П. Большая энциклопедия компьютера и Интернета. – М., Олма-пресс образование, 2005. – 1104 с.

7. Васильев А. *VBA в Office 2000*: учебный курс / А. Васильев, А. Андреев. Спб.: Питер, 2001. – 325 с.

### **6.3. Internet-ресурсы**

8. Государственный образовательный стандарт высшего профессионального образования. Специальности 031600 «Реклама и связи с общественностью», 100400 «Туризм». 2010. URL: <http://www.intuit.ru> (дата обращения: 20.08.2011).



Варианты задач 1, 2, 3 ИДЗ

Варианты задачи 1

$$1) \quad y = \frac{\sqrt{abc}}{2.4} - \frac{0.7abc}{(a+b)^2} x - e^{(a+b)}$$

$$a = 3.4; b = 1.7; c = 4.5; x = 1.7$$

$$2) \quad y = \frac{\sqrt{c} + d \cdot x^3 - 0.41}{|a-b|} + \frac{\ln(a-b)}{(\sqrt{c} + d \cdot x^2)^2} - e^{(a-b)}$$

$$a = 4.5; b = 1.2; c = 4.3; d = 3.1; x = 1.4$$

$$3) \quad y = \frac{\sqrt{a^2 + b^2}}{x^2 - a} - 1.7 \frac{(a^2 + b^2) \cdot 10^{-2}}{a+b} - \sin(a^2 - b^2)$$

$$a = 1.4; b = 27; x = 1.6$$

$$4) \quad y = \frac{abx + \operatorname{tga} \cdot bx}{|a-b| + 0.4x} - 10^{-2} \ln x$$

$$a = 1.4; b = 27; x = 1.6$$

$$5) \quad y = \frac{(ax-b)^3 + |a-b| + e^{ax}}{(a-b)^2 + 10^3 b^2 + \ln(ax)}$$

$$a = 1.5; b = 2; x = 4.7$$

$$6) \quad y = \frac{\arcsin(a-b)}{2.5x} + \sqrt{x} - \frac{(a-b)^2}{x}$$

$$a = 4.5; b = 2.5; x = 4.7$$

$$7) \quad y = \frac{a^3 x - b|d|}{|d| + bc} - 10^{-2} \cos^2 x + \sqrt{a^3 x}$$

$$a = 5.2; b = 3.1; c = -1.5; d = 0.07; x = 0.5$$

$$8) \quad y = 1.2 \frac{(a-b)^3 - \frac{c}{x^2}}{(|a|-b)(a-b)} e^{x^2} + 10^{-1} \operatorname{tg}(a-b)$$

$$a = 7.4; b = 4.5; c = -0.75; x = 1.5$$

$$9) \quad y = 1.1 \frac{\sqrt{(a+b)^3 + (b-c)^2}}{ax^3 + |b-c|} - 3^{-2} ad^2 \sin^2(b-c)$$

$$a = 0.07; b = 1.7; c = 2.6; d = 0.27; x = -0.72$$

$$10) \quad y = 3.1 \frac{\sqrt{ac^2} - |a+b|}{|a| + |b|} + \ln(\sin^2 x)$$

$$a = 4.5; b = -1.7; c = 2.74; x = 1.57$$

$$11) \quad y = 2.7 \frac{\sqrt{ax+cd}}{(x+a)^2} - e^{-3x} \sin(ax^2 + cd) + e^{(x+a)}$$

$$a = 5.5; d = 5.25; c = -0.2; x = 7.2$$

$$12) \quad y = 4.1 \frac{a^3 x - |b|}{(a+b)^2} - \operatorname{tg} \frac{a+b}{(b-c)^2} - e^{(x+b)}$$

$$a = 2.4; b = -3.2; c = 5.7; x = 0.75$$

$$13) \quad y = 0.5 \frac{a^2 x + |d|}{a + \ln b} - \sqrt{\frac{a^3 x^2}{a-b}} - \cos(a-b)$$

$$a = 4.7; b = 2.4; d = -0.01; x = 3.5$$

$$14) \quad y = 10^{-2} \frac{\sqrt{c-d} + |a+b|}{b^2 + c^2 x} + e^{(c-d)} + 4.1 \operatorname{tg}(c-d)^2$$

$$a = 2.2; b = -18; c = 7.7; d = 4.5; x = 0.12$$

$$15) \quad y = \frac{a^2 x + |d|}{(b-c)^2 - a^2 + x} - 2.5 \ln(c-b) + 10^{-2} \frac{(b-c)^2}{\sin x}$$

$$a = 4.5; b = 2.5; c = 3.7; d = -1.7; x = -1.2$$

$$16) \quad y = 1.7 \frac{|a+b| - |c-d|}{bx^2 + c^2d} - \sqrt{\frac{(a+b)^2}{c-d}} - 10^{-2} \cos^2(c-d)$$

$a = -4.5; b = 3.7; c = 7.4; d = 1.5; x = -0.7$

$$17) \quad y = 2.5 \frac{axd^2}{2} + 10^{-1} \frac{\sqrt{x-a}}{ax} - \cos^3(x^2 - a^2)$$

$a = 3.4; d = 1.2; x = 7.5$

$$18) \quad y = 3.2 \frac{\operatorname{tg}(x-a)}{(x-a)^2} - 10^{-2}(x-a) + \operatorname{tg}x^2 \cdot e^{(x-a)}$$

$a = 5.1; x = 7.7$

$$19) \quad y = 1.5 \frac{|bx| - \sqrt{acd}}{\lg|acd|} \cdot e^{acd} - 10^{-3} \arcsin^2(bx)$$

$a = 3.2; b = 1.7; c = 2; d = 3.7; x = 7.1$

$$20) \quad y = \sqrt{cx} - 2.7 \frac{|c| + |x|}{c^2x^2} \cdot e^{cx} + \cos \frac{(a+b)^2}{cx-b}$$

$a = 3.7; b = 0.07; c = 1.5; x = 5.75$

$$21) \quad y = 4.5 \frac{(a+b)^2}{(a-b)^2} - \sqrt{(a+b)(a-b)} + 10^{-1} \frac{\ln(a-b)}{\ln(a+b)} \cdot e^{x^2}$$

$a = 7.5; b = 1.2; x = 0.5$

$$22) \quad y = 2.4 \left| \frac{x^2 + b}{a} \right| + (a-b) \sin^2(a-b) + 10^{-2}(x-b)$$

$a = 5.1; b = 0.7; x = -0.05$

$$23) \quad y = \frac{ax - \sqrt{b}}{5.7(x^2 + b^2)} - \frac{|x+b| - a^2}{x^2} \operatorname{tg}^2 b$$

$a = 0.1; b = 2.4; x = -0.3$

$$24) \quad y = \sqrt{\frac{c - dx^2}{x}} + \frac{\ln(x^2 + c)}{0.7x + ad} - \frac{10^{-2}}{c - dx^3}$$

$$a = 4.5; c = 7.4; d = -2.1; x = 0.15$$

$$25) \quad y = \arcsin \frac{x^2}{a} - \frac{ax^3 - \sqrt{b}}{\sqrt{a + b^2}} + 0.05 \frac{e^{x^2}}{x^2}$$

$$a = 2.1; b = 3.12; x = 1.14$$

### Варианты задачи 2

$$1) \quad y = \begin{cases} ab - x^3, & a < x < b, \\ 4(x^2 + a), & x \geq b, \\ x^2(a + b), & x \leq a, \end{cases}$$

$$a = -0.0000124, b = \frac{2}{5}$$

$$2) \quad y = \begin{cases} 7 \cdot (b/x - a/b)^2, & x > a, \\ 2 \cdot (a - b)^3 (a \cdot x + b \cdot c), & x < b, \\ 3 \cdot a \cdot b \cdot x, & x = a, \end{cases}$$

$$a = 2, b = 3.5, c = 4.1$$

$$3) \quad y = \begin{cases} \cos(a \cdot x^2 + b)^2, & -19 \leq x < -16, \\ 5 \cdot (b \cdot x + a + d), & x \geq -16, \\ 2 \cdot x \cdot d + a^2, & x < -19, \end{cases}$$

$$a = -9/2, b = 5.61, d = 24$$

$$4) \quad y = \begin{cases} a^2 \cdot b/9 - |x|, & x \leq 2, \\ 4 \cdot (a^3 \cdot x^2 - c^2), & -2 < x < 4, \\ \operatorname{tg}[(a - b)^2 + x^2], & x \geq 4, \end{cases}$$

$$a = -0.0025, b = -2.7, c = 3.$$

$$5) \quad y = \begin{cases} \sqrt{a^2 \cdot |x| + |b \cdot c|}, & 1 \leq x < a, \\ 10 \cdot (|a \cdot x| + b \cdot x), & x > a, \\ a \cdot x^2, & x < -4, \end{cases}$$

$$a = 21.345, b = -2/5, c = 1$$

$$6) \quad y = \begin{cases} x - a, & x < -1, \\ \sqrt{x - a}, & -1 \leq x \leq 1, \\ 9 \cdot x^2 - a, & x > 1, \end{cases}$$

$$a = -1.5$$

$$7) \quad y = \begin{cases} \sqrt{(b + x^2)}, & -2 < x \leq 3, \\ a \cdot b / x - x^2, & x > 3, \\ 4 \cdot (x + a/b), & x \leq -2, \end{cases}$$

$$a = -0.024, b = 0.752479$$

$$8) \quad y = \begin{cases} 5 \cdot |b - a| / (|b| - |c|) \cdot x^2, & 3 \leq x \leq 5, \\ 2.5(|x| - c)(a \cdot x + b), & x < 3, \\ 10^2 \cdot d \cdot b \cdot c / x, & x > 5, \end{cases}$$

$$a = 7.4, b = 2.5, c = 3.1, d = 0.5$$

$$9) \quad y = \begin{cases} 5 \cdot c^2 \cdot d^2 \cdot b, & x = 3.5, \\ 4 \cdot (c \cdot d^2 + x^2), & 1.5 \leq x < 3.5, \\ (a \cdot x + b)^2, & x > 3.5, \end{cases}$$

$$a = 4.7, b = 2.1, c = 1.7, d = 0.5$$

$$10) \quad y = \begin{cases} 4 \cdot (x/a^2 - a), & x < -1, \\ \sqrt{x + a}, & -1 \leq x \leq 1, \\ x^3, & x > 12.1, \end{cases}$$

$$a = 12.1$$

$$11) \quad y(x) = \begin{cases} x^2 + \cos(a + b \cdot x), & x < 0, \\ 4 \cdot (x - 1), & x > 1, \\ (x + 2) + 0.0024, & 0 \leq x \leq 1, \end{cases}$$

$$a = -0.24, b = 2$$

$$12) \quad y(x) = \begin{cases} (a^2 - 0.008 \cdot b^2)/x, & a < x \leq b, \\ b \cdot x - a, & x \leq a, \\ (a + b) \cdot \cos x, & x > b, \end{cases}$$

$$a = -0.2, b = 2$$

$$13) \quad y(x) = \begin{cases} 0.003 \cdot a/x, & x \leq c, \\ 2 \cdot x/a, & x \geq a + b, \\ c \cdot (a \cdot x + b)^2, & c < x < a + b, \end{cases}$$

$$a = 0.8, b = -0.26, c = -0.24.$$

$$14) \quad y(x) = \begin{cases} |a|b - x^2, & x \leq 0, \\ a^2 \cdot x + b^2, & 0 < x \leq 0.1, \\ b \cdot (x^2 + a^2), & x > 0.1, \end{cases}$$

$$a = 2/7, b = -3.01.$$

$$15) \quad y(x) = \begin{cases} a^3 + d \cdot x^2, & 0.1 \leq x \leq a, \\ (b - x)^2, & x > a, \\ (b^2 + x^2) \cdot a - x, & x < 0.1, \end{cases}$$

$$a = 2.51, b = -3.06, d = 5/3.$$

$$16) \quad y(x) = \begin{cases} x^2 + a, & x < a, \\ 8.4 \cdot (x + a), & a \leq x \leq d, \\ d \cdot x/15 - a^x, & x > d, \end{cases}$$

$$a = 0.2, d = 1.9.$$

$$17) \quad y(x) = \begin{cases} e^{a \cdot x}, & x < -1, \\ \cos x + 0.0085, & -1 \leq x < 0, \\ 4 \cdot x^2 - a, & x \geq 0, \end{cases}$$

$$a = 29.4.$$

$$18) \quad y(x) = \begin{cases} (a \cdot x^2 + 2)/(x^2 + 1) & 1 < x < 3, \\ a, & |x| \geq 3, \\ a \cdot x/(x + 2), & |x| \leq 1, \end{cases}$$

$$a = 9/4.$$

$$19) \quad y(x) = \begin{cases} (|a| - |b|)/(a \cdot b + x)/c, & 3 < x \leq 5, \\ (a - b)^2 \cdot x^3, & 5 < x < 9.8, \\ (a \cdot x^2 + b)/4 \cdot x, & x \geq 9.8, \end{cases}$$

$$a = 19.2457, b = 24.7, c = 1.5.$$

$$20) \quad y(x) = \begin{cases} (4 \cdot [a \cdot x - 6 \cdot (a + b)^3]), & a + b \leq x < a, \\ a^2 + (a - b) + x^2, & x \geq a, \\ x^2 \cdot (a + b), & x < a + b, \end{cases}$$

$$a = 2.005, b = -0.0009$$

$$21) \quad y(x) = \begin{cases} a^2(a - b) \cdot x, & |a| < x \leq |b|, \\ 5 \cdot (a \cdot x^2 + b), & x > |b|, \\ (a \cdot x - |a|)^2, & x \leq |a|, \end{cases}$$

$$a = 0.6, b = -1.5$$

$$22) \quad y = \begin{cases} (a \cdot b - c \cdot x)/a \cdot b \cdot c \cdot x^2, & x < 0, \\ \sqrt{a \cdot x^2 + |b \cdot x|}, & 0 \leq x \leq 0.83, \\ x^2 + a \cdot b, & x > 0.83, \end{cases}$$

$$a = -5/3, b = 2.3, c = 3.335$$

$$23) \quad y(x) = \begin{cases} \cos(x^2 - a \cdot b - a^{1/2}), & a < x \leq b, \\ (x - a)^3 + b^2, & x \leq a, \\ (c + d) + c \cdot x, & x > 1, \end{cases}$$

$$a = 0.24, b = 2, c = 8/3, d = 0.5$$

$$24) \quad y = \begin{cases} 9.25 - |a \cdot c + d \cdot x|, & a \leq x \leq d, \\ 4 \cdot (d^2/a + c^2 \cdot x/b^2), & x > d, \\ a \cdot d^2 \cdot c \cdot x, & x < a, \end{cases}$$

$$a = -2/3, b = 5.61, c = 4, d = 2.5$$

$$25) \quad y = \begin{cases} \cos(a \cdot x + z), & a \leq x < z, \\ x + z + 4.28, & x < a, \\ 9 \cdot (x + z \cdot a), & x \geq z, \end{cases}$$

$$a = -0.025, z = 0.25$$

### Варианты задачи 3

- |  |   |
|--|---|
| 1) $y = 10^{-2}bc/x + \cos\sqrt{a^3x}$ ,<br>$x_0 = -1.5; x_k = 3.5; dx = 0.5;$<br>$a = -1.25; b = -1.5; c = 0.75;$ | 2) $y = 1.2(a-b)^3 e^{x^2} + x$ ,<br>$x_0 = -0.75; x_k = -1.5; dx = -0.05;$<br>$a = 1.5; b = 1.2;$        |
| 3) $y = 10^{-1}ax^3 \operatorname{tg}(a - bx)$ ,<br>$x_0 = -0.5; x_k = 2.5; dx = 0.05;$<br>$a = 10.2; b = 1.25;$   | 4) $y = a \cdot x^3 + \cos^2(x^3 - b)$ ,<br>$x_0 = 5.3; x_k = 10.3; dx = 0.25;$<br>$a = 1.35; b = -6.25;$ |
| 5) $y = x^4 + \cos(2 + x^3 - d)$ ,<br>$x_0 = 4.6; x_k = 5.8; dx = 0.2;$<br>$d = 1.3;$                              | 6) $y = x^2 + \operatorname{tg}(5x + b/x)$ ,<br>$x_0 = -1.5; x_k = -2.5; dx = -0.5;$<br>$b = -0.8;$       |
| 7) $y = 9 \cdot (x + 15\sqrt{x^3 + b^3})$ ,<br>$x_0 = -2.4; x_k = 1; dx = 0.2;$<br>$b = 2.5;$                      | 8) $y = 9x^4 + \sin(57.2 + x)$ ,<br>$x_0 = -0.75; x_k = -2.05; dx = -0.2;$                                |
| 9) $y = 0.0025 \cdot b \cdot x^3 + \sqrt{x + e^{0.85}}$ ,<br>$x_0 = -1; x_k = 4; dx = 0.5;$<br>$b = 2.3;$          | 10) $y = x \cdot \sin(\sqrt{x + b - 0.0084})$ ,<br>$x_0 = -2.05; x_k = -3.05; dx = -0.2;$<br>$b = 3.4;$   |
| 11) $y = x + \sqrt{ x^3 + a - be^x }$ ,<br>$x_0 = -4; x_k = -6.2; dx = -0.2;$<br>$a = 0.1; b = 1.25$               | 12) $y = 9(x^3 + b^3) \operatorname{tg}x$ ,<br>$x_0 = 1; x_k = 2.2; dx = 0.2;$<br>$b = 3.2;$              |
| 13) $y =  x - b ^{1/2} /  b^3 - x^3 ^{3/2} + \ln x - b $ ,<br>$x_0 = -0.73; x_k = -1.73; dx = -0.1;$<br>$b = -2;$  | 14) $y = (x^{5/2} - b) \ln(x^2 + 12.7)$ ,<br>$x_0 = 0.25; x_k = 5.2; dx = 0.3;$<br>$b = 0.8;$             |



- 15)  $y = 10^{-3}|x|^{5/2} + \ln|x+b|$ ,  
 $x_0 = 1.76; x_k = -2.5; dx = -0.25;$   
 $b = 35.4;$
- 16)  $y = 15.28|x|^{-3/2} + \cos(\ln|x|+b)$ ,  
 $x_0 = 1.23; x_k = -2.4; dx = -0.3;$   
 $b = 12.6;$
- 17)  $y = 0.00084(\ln|x|^{5/4} + b)/(x^2 + 3.88)$ ,  
 $x_0 = -2.35; x_k = -2; dx = 0.05;$   
 $b = 74.2;$
- 18)  $y = 0.8 \cdot 10^{-5}(x^3 + b^3)^{7/6}$ ,  
 $x_0 = -0.05; x_k = 0.15; dx = 0.01;$   
 $b = 6.74;$
- 19)  $y = (\ln(\sin(x^3 + 0.0025) + 1))^{3/2} + 0.8$ ,  
 $x_0 = 0.12; x_k = 0.64; dx = 0.2;$
- 20)  $y = a + x^{2/3} \cos(x + e^x)$ ,  
 $x_0 = 5.62; x_k = 15.62; dx = 0.5;$   
 $a = 0.41;$
- 21)  $y = x^{b^b} + \cos(x^{3/2} + b^{3/4})$ ,  
 $x_0 = 13.7; x_k = 19.1; dx = 0.4;$   
 $b = 2;$
- 22)  $y = 10^{-2}(a + bx) - e^{x^3+b}$ ,  
 $x_0 = -3.4; x_k = -1.4; dx = 0.1;$   
 $a = 5; b = 4;$
- 23)  $y = ax^3 + b^{5/4}xe^{-x}$ ,  
 $x_0 = 2.51; x_k = 10.59; dx = 1.01;$   
 $a = 4; b = 2;$
- 24)  $y = a|x|^{5/2} + \cos(\sqrt{e^x})$ ,  
 $x_0 = -0.31; x_k = 0.61; dx = 0.3;$   
 $a = 8;$
- 25)  $y = 3.1\sqrt{ax^2} - |a+b|x$ ,  
 $x_0 = -2.35; x_k = -5.55; dx = -0.05;$   
 $a = 2; b = -5;$

*11 заданий к лабораторной работе №1*

1.

```
Sub UsingVariable()
    Dim Num1 As Integer
    Num1 = Worksheets(1).Range("A1").Value
    MsgBox Num1
End Sub
```

2.

```
Public Num1 As Integer
'
*****
Sub UsingVariable()
    Num1 = Worksheets(1).Range("A1").Value
    MsgBox Num1
End Sub
*****
```

3.

```
*****
Sub UsingConst()
    Const Num1 As String = "Tomsk Polytechnic Univ."
    MsgBox Num1
End Sub
*****
```

4.

```
Const Num1 As String = "Tomsk Polytechnic Univ."
*****
```

```
Sub UsingConst()  
    MsgBox Num1  
End Sub  
!*****
```

5.

```
Public Const Num1 As String = "Tomsk Polytechnic Univ."  
!*****  
Sub UsingConst()  
    MsgBox Num1  
End Sub  
!*****
```

6.

```
Sub Call_Function()  
    Dim var1 As Integer  
    Dim var2 As Integer  
    Dim var3 As Integer  
    var1 = 5  
    var2 = 10  
    var3 = Multiply(var1, var2)  
    MsgBox (var3)  
End Sub
```

```
Function Multiply(ByVal var1 As Integer, ByVal var2 As Integer)  
    Multiply = var1 * var2  
End Function
```

7.

```
Sub Proc()  
    var1 = GetRand
```

```

    ModuleLevelProcedure var1
End Sub
Private Sub ModuleLevelProcedure(ByVal var1)
    MsgBox var1 * GetRand
End Sub
Private Function GetRand()
    GetRand = Int(100 * Rnd())
End Function

```

8.

```

Sub PassArgumentByReference()
    Dim Username As String
    Username = "Mike"
    ChangeName Username
    MsgBox Username
End Sub

Sub ChangeName(ByRef Username)
    Username = "Kate"
End Sub

```

9.

```

Sub PassArgumentbyValue()
    Dim Username As String
    Username = "Mike"
    ChangeName1 Username
    MsgBox "Yet remains as " & Username & " in the original proc."
End Sub

```

```

Sub ChangeName1(ByVal Username)
    MsgBox "The original name is " & Username & "."
    Username = "Kate"
    MsgBox "The name is changed to " & Username & " in the called
proc."
End Sub

```

10.

### Option Base 1

```

Sub UsingArray()
    Dim Vals(3) As Integer
    Vals(1) = Int(100 * Rnd())
    Vals(2) = Int(100 * Rnd())
    Vals(3) = Int(100 * Rnd())
    MsgBox "Lottery numbers: " & Vals(1) & ", " & Vals(2) & "," &
Vals(3)
End Sub

```

11.

### Option Base 1

```

Sub Using2DVariantArray()
    Dim Data2D(10, 3) As Variant
    Data2D(1, 1) = "Kate"
    Data2D(1, 2) = 43
    Data2D(1, 3) = #4/15/1962#
    MsgBox Data2D(1, 1) & ", age " & Data2D(1, 2) & ", born " &
Data2D(1, 3)
End Sub

```

## Приложение 3

### Путь к обращению «Редактор *Visual Basic. Excel 2010* »

Если в главном меню *Excel* отсутствует **Разработчик** (необходимый для работы с объектами *VBA*), то его надо установить.

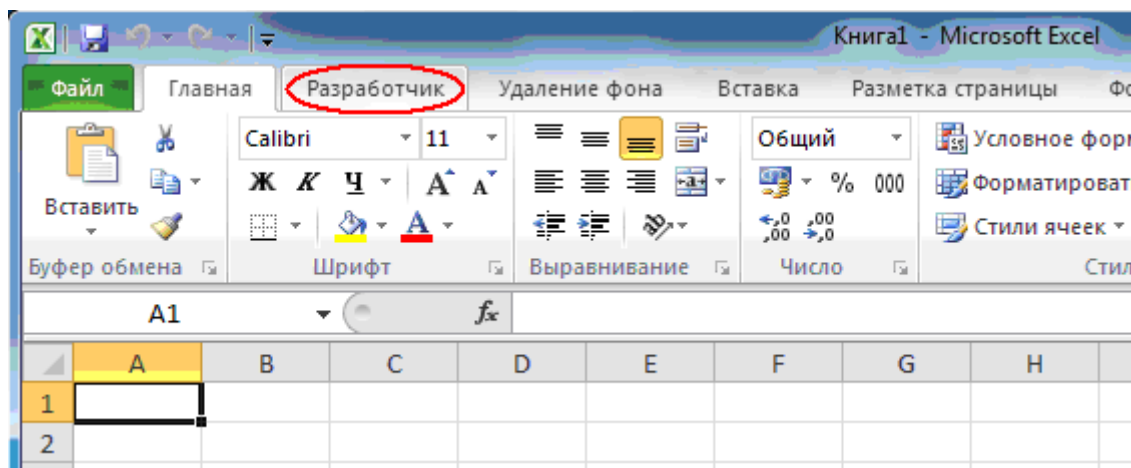
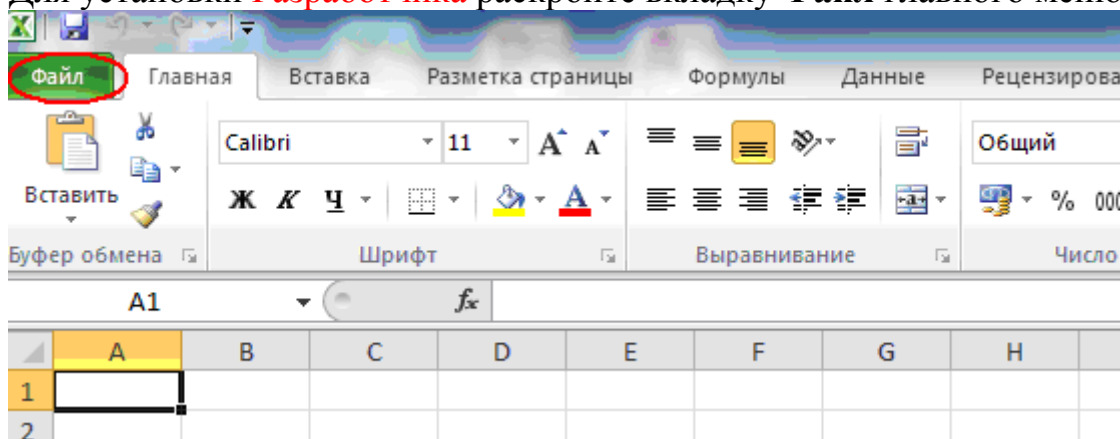
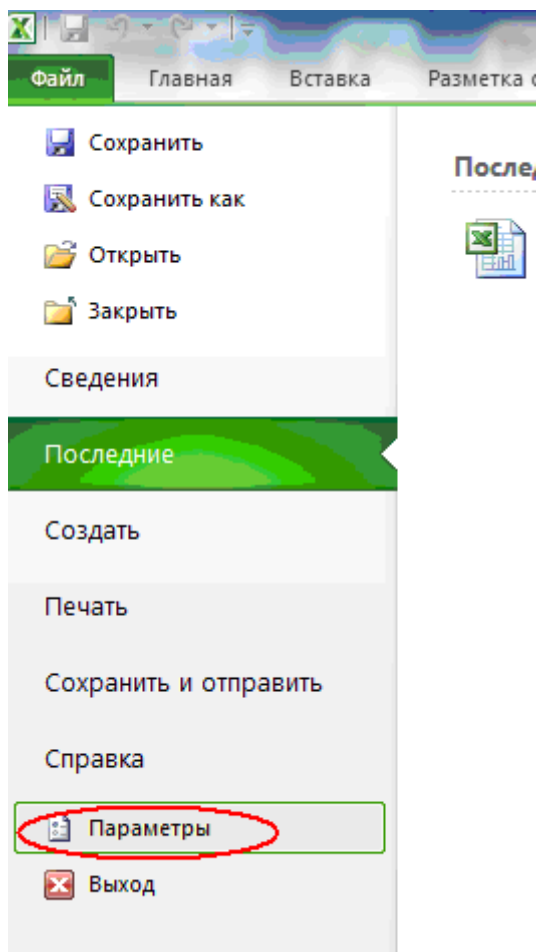


Рис. Главное меню *Excel*, содержащее «**Разработчик**»

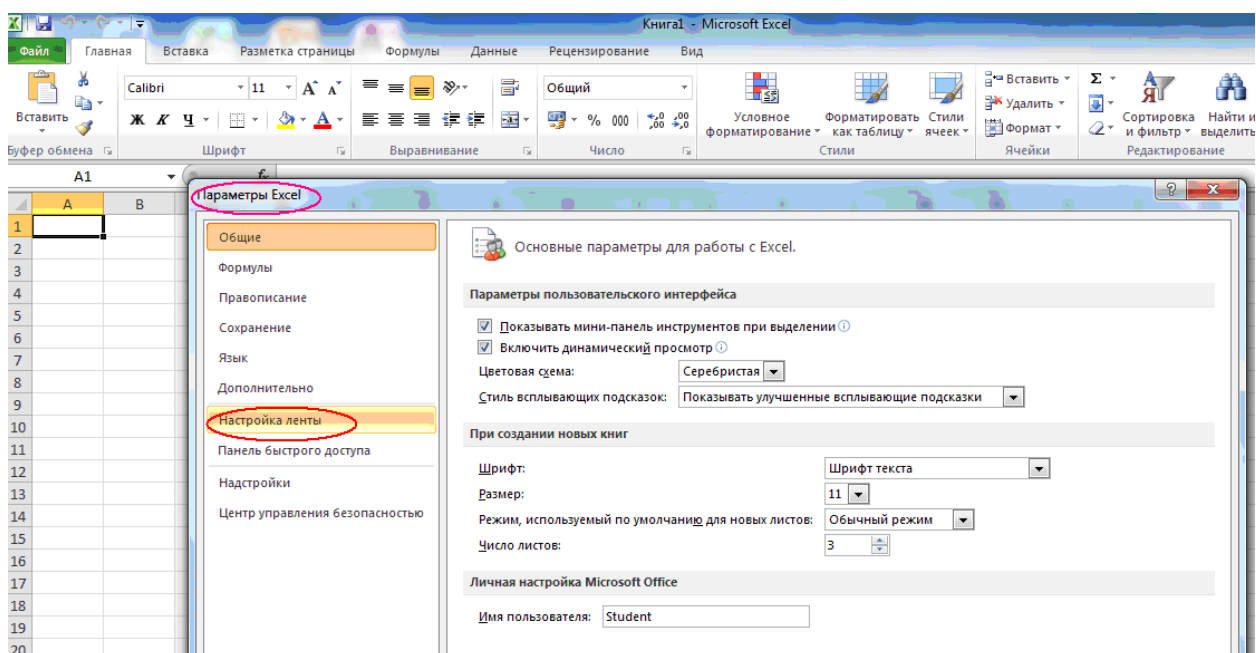
Для установки **Разработчика** раскройте вкладку **Файл** главного меню.



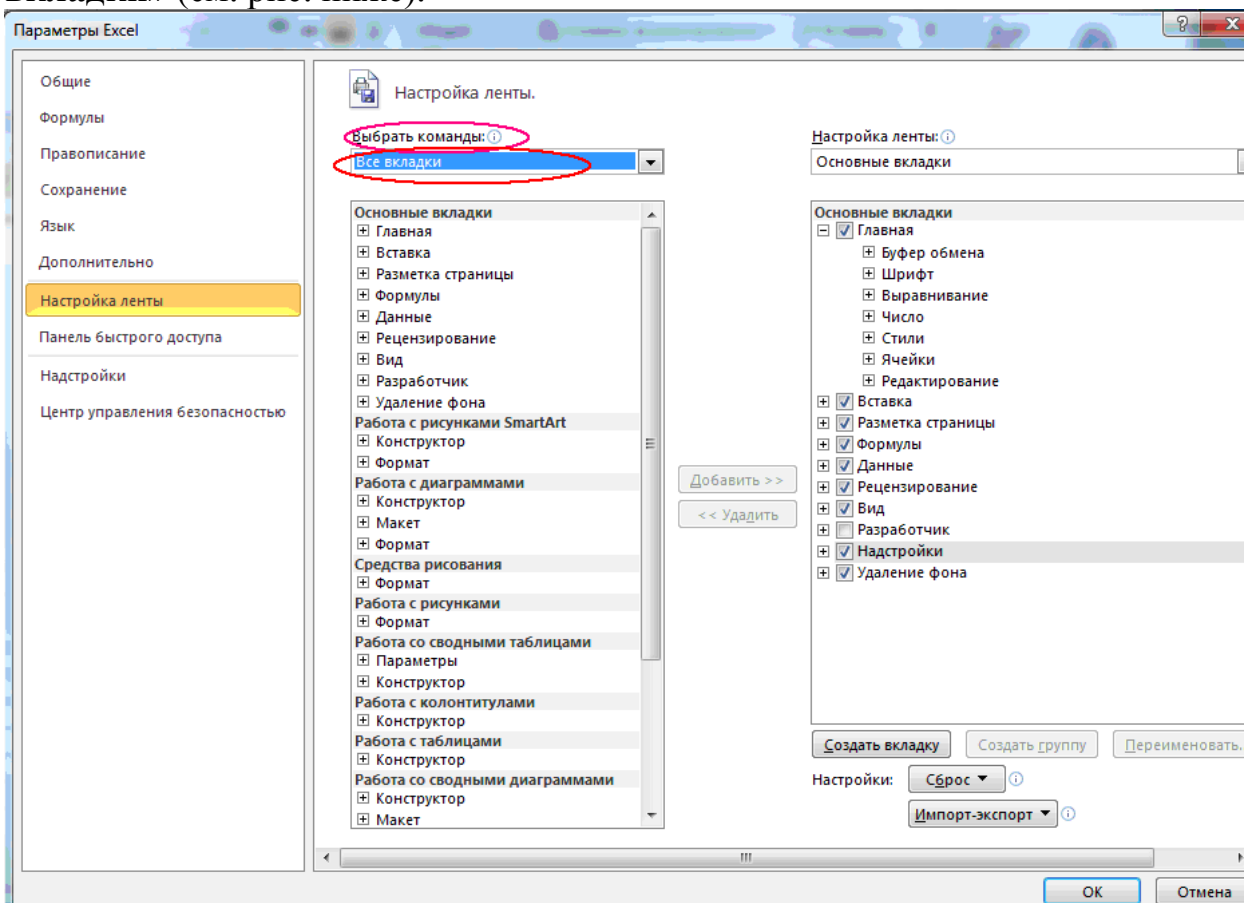
Выберите пункт **Параметры** (см. рис. ниже).



В появившемся окне **Параметры Excel** выберите пункт **Настройка ленты** (см. рис. ниже).

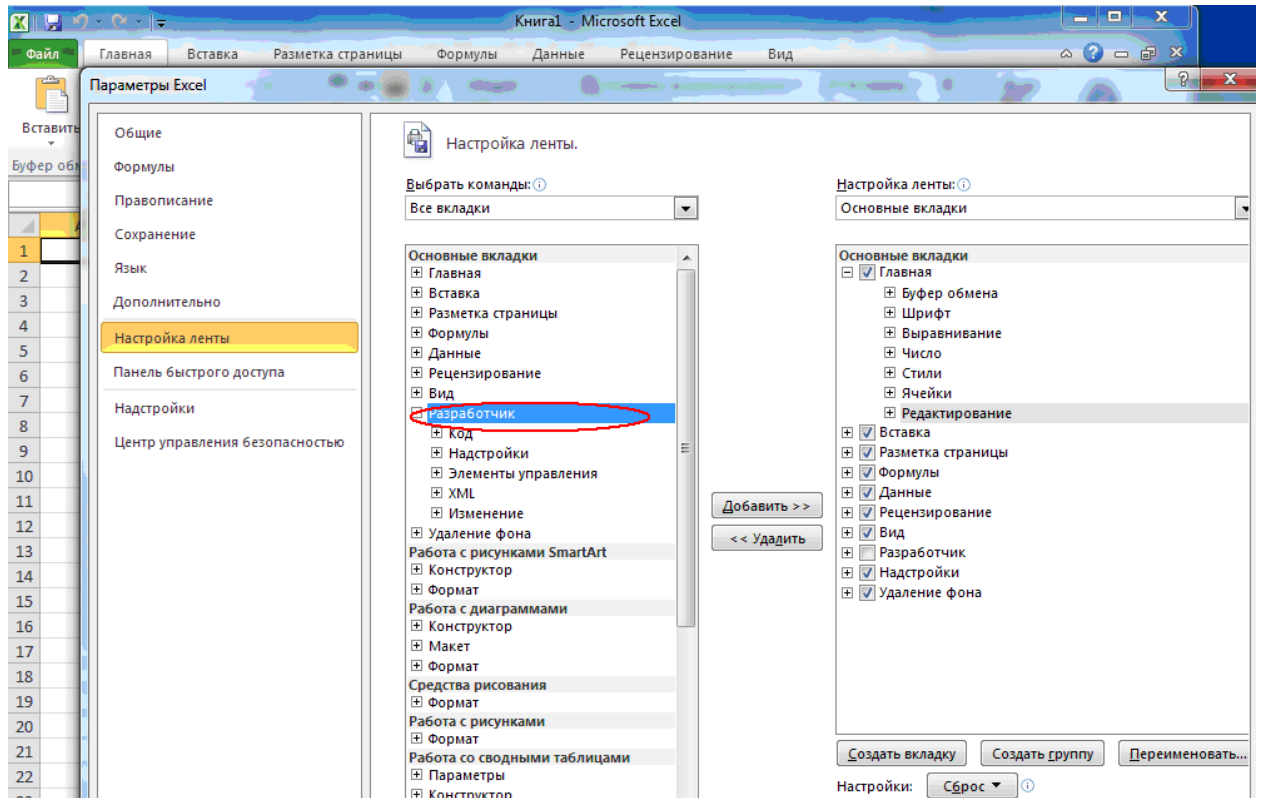


В настройке ленты в меню «**Выбрать команды**» выберите пункт «**Все вкладки**» (см. рис. ниже).

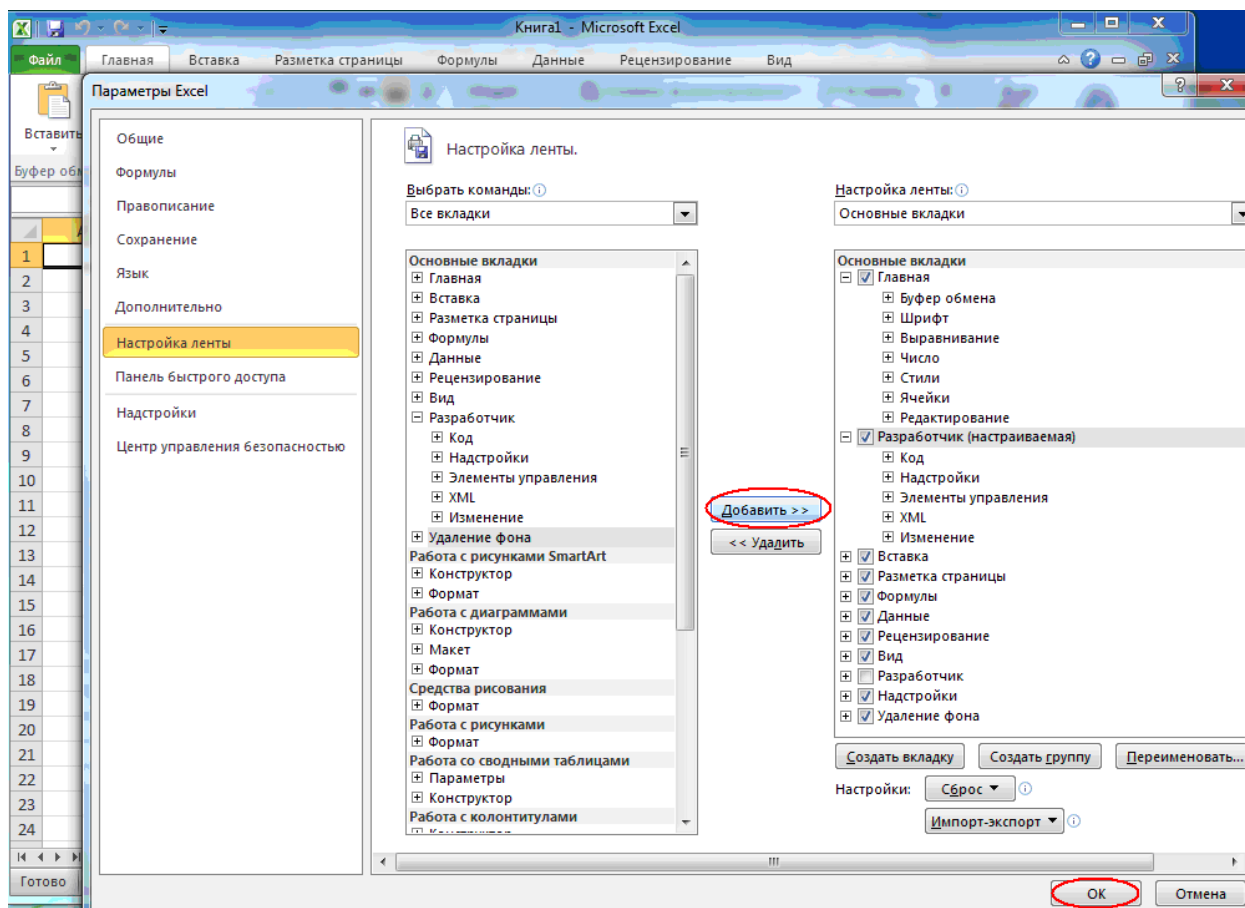


Затем выбрать «**Разработчик**».

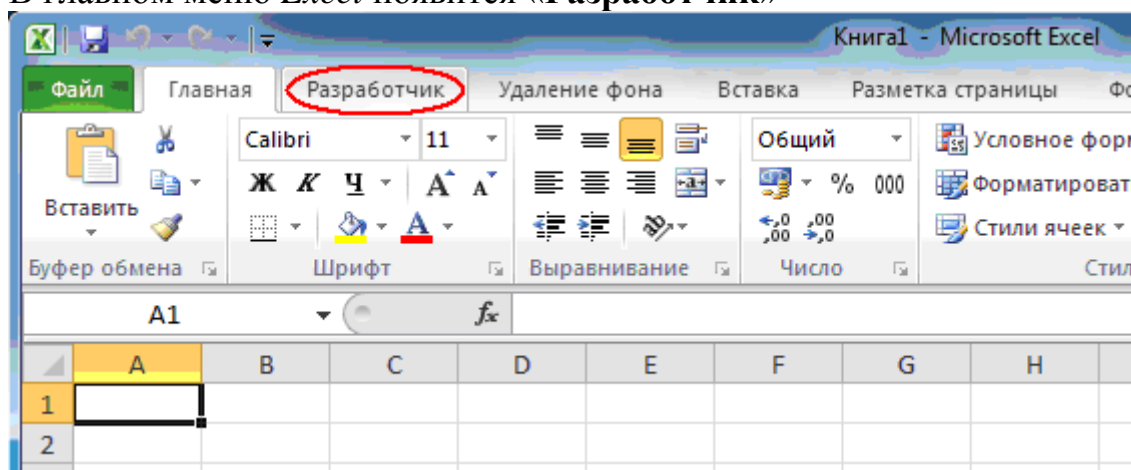




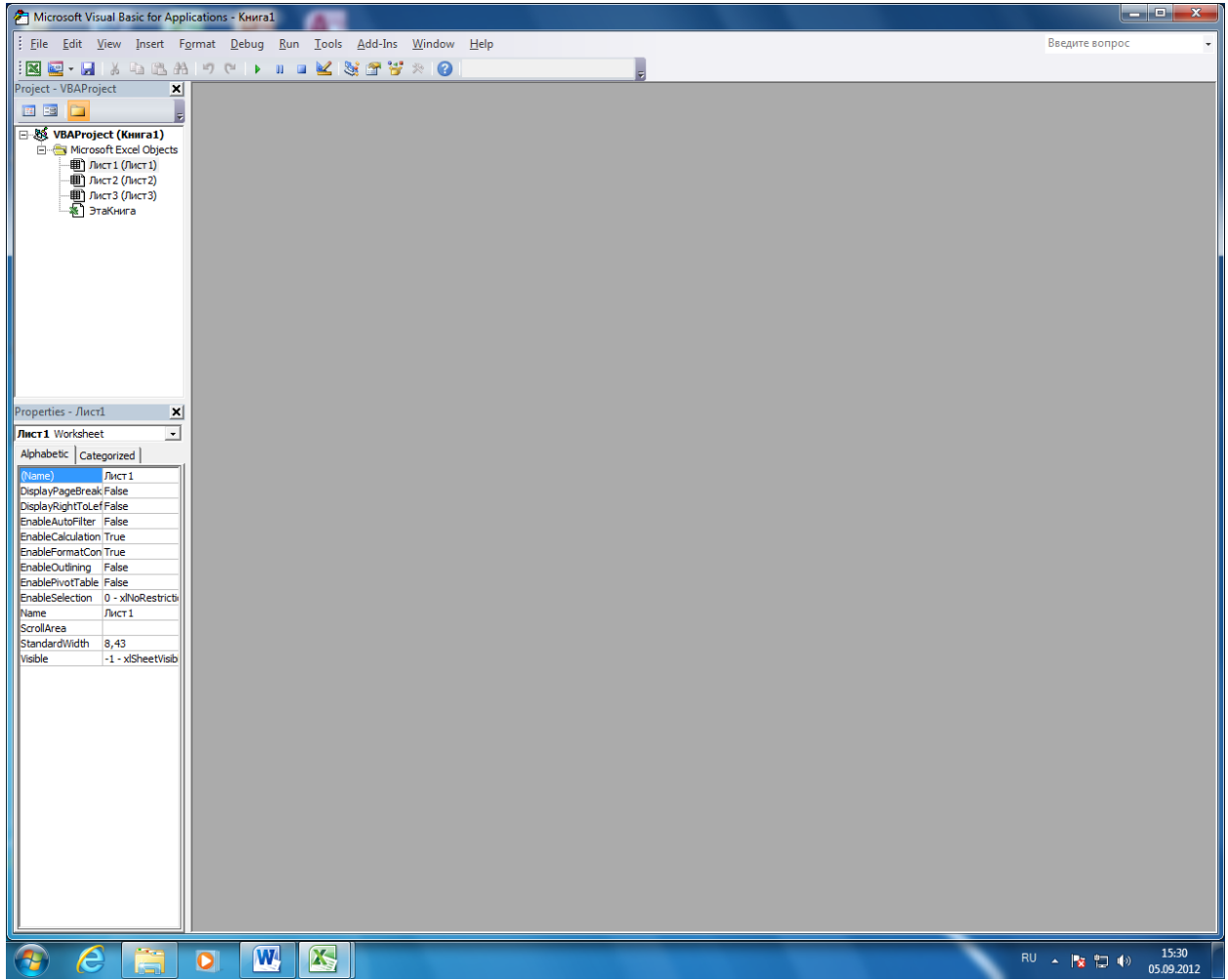
Нажать на кнопку «Добавить» и затем нажать на «ОК».



В главном меню *Excel* появится «Разработчик»



Раскрываем список **Разработчика**, вызываем *VBA*, далее работаем, как указано в методических материалах применительно к *Excel 2003*.



Учебное издание

## ИНФОРМАТИКА

Методические указания и индивидуальные задания

*Составитель*

ШЕФЕР Ольга Владимировна

Рецензент

*канд. физ.-мат. наук  
доцент кафедры ИПС*

*В.И. Рейзлин*

Редактор *С.В. Ульянова*

Компьютерная верстка *Т.И. Тарасенко*

**Отпечатано в Издательстве ТПУ в полном соответствии  
с качеством предоставленного оригинал-макета**


Подписано к печати . Формат 60×84/16. Бумага «Снегурочка».  
Печать Херох. Усл.печ.л. 1,92. Уч.-изд.л. 1,74.  
Заказ . Тираж экз.



Национальный исследовательский Томский политехнический университет  
Система менеджмента качества

Издательства Томского политехнического университета сертифицирована  
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008



**ИЗДАТЕЛЬСТВО**  **ТПУ**. 634050, г. Томск, пр. Ленина, 30.  
Тел./факс: 8(3822)56-35-35, [www.tpu.ru](http://www.tpu.ru)