Министерство образования Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра "Экономика и организация машиностроительного производства"

О.А. Лавренова

**ЗАДАНИЯ   
К РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЕ**

**по дисциплине «Информатика»  
на тему «Автоматизация решения задач пользователя   
средствами Microsoft Office Excel»  
для студентов 1 курса дневной формы обучения  
специальности 1-27 01 01**

**2018**

**Содержание**

[Общие сведения 3](#_Toc511682711)

[Структура работы (документ Word) 3](#_Toc511682712)

[Варианты 5](#_Toc511682713)

[Вопросы для изучения 6](#_Toc511682714)

[Задания практической части 8](#_Toc511682715)

[2.1 Разработка функции пользователя 8](#_Toc511682716)

[2.2 Разработка формы в VBA MS Excel 9](#_Toc511682717)

[2.3 Создание пользовательской электронной таблицы 11](#_Toc511682718)

[2.4 Решение задачи математического программирования средствами MS Excel 14](#_Toc511682719)

[Приложение А Краткие требования к оформлению расчетно-графической работы 18](#_Toc511682720)

[Приложение Б Титульный лист работы 19](#_Toc511682721)

[Приложение В Пример оформления задания 2 в документе Word 20](#_Toc511682722)

[2 Практическая часть 20](#_Toc511682723)

[2.1 Построение графиков функций 20](#_Toc511682724)

[2.1.1 Постановка задачи 20](#_Toc511682725)

[2.1.2 Разработка алгоритма 20](#_Toc511682726)

[2.1.3 Разработка макета электронной таблицы 21](#_Toc511682727)

[2.1.4 Вычисления в электронной таблице 22](#_Toc511682728)

[2.1.5 Построение графиков функций 22](#_Toc511682729)

# Общие сведения

Расчетно-графическая работа на тему «Автоматизация решения задач пользователя средствами Microsoft Office Excel» выполняется во 2-ом учебном семестре.

В процессе выполнения работы создаются 2 файла:

* **документ Word** (с ответами на вопросы и описанием выполнения заданий)**;**
* **рабочая книга Excel**.

**Номер варианта определяется по списку группы.**

**Готовая работа сдается на проверку не менее чем за 2 недели до начала сессии (ауд. 311, к.1).**

**К экзамену предоставляется краткий конспект ответов на вопросы.**

**Цель работы**

1. Изучение этапов решения задач пользователя на ЭВМ.
2. Освоение правил построения алгоритмов в соответствии с ГОСТ 19.701-90.
3. Приобретение практических навыков работы в среде электронных таблиц Microsoft Office Excel.

# Структура работы (документ Word)

Титульный лист

Содержание (**автоматически сформированное**)

Введение (**1 страница**)

1. **Теоретическая часть (10-15 страниц)**
   1. Автоматизация решения задач пользователя
      1. Этапы решения задач пользователя на ЭВМ
      2. Понятие алгоритма, его свойства и способы описания
      3. Базовые управляющие структуры алгоритма
      4. Правила оформления блок-схем алгоритмов по ГОСТ 19.701-90 **(краткая характеристика основных элементов блок-схем в виде таблицы)**
   2. Основы языка программирования VBA Microsoft Office Excel
      1. Переменные, константы, типы данных, правила задания имен объектов и их описания (примеры)
      2. Выражения VBA: правила записи, приоритет операций. Способы организации ввода/вывода данных, синтаксис диалоговых окон
      3. Понятие и виды операторов VBA MS Excel (присваивание, условие, цикл)
      4. Встроенные функции VBA MS Excel: математические, преобразования типов данных и форматирования (в виде таблиц)
      5. Пользовательская функция в MS Excel: ее синтаксис, аргументы и результат. Основные правила написания программного кода в VBA MS Excel
   3. Формы в VBA Microsoft Office Excel
      1. Понятие формы, порядок ее конструирования, программирования и настройки свойств
      2. Краткая характеристика основных элементов управления в формах (в виде таблицы)
   4. Вопрос l
   5. Вопрос 2
2. **Практическая часть (15-20 страниц с описанием выполнения заданий, включая распечатки листов рабочей книги MS Office Excel)**
   1. Задание 1 **(по варианту)**
      1. Постановка задачи
      2. Разработка алгоритма
      3. Разработка программного кода функции пользователя
      4. Проверка программного кода функции
      5. Выводы
   2. Задание 2 **(по варианту)**
      1. Постановка задачи
      2. Разработка алгоритма
      3. Разработка макета пользовательской формы
      4. Разработка программного кода процедур обработки событий пользовательской формы
      5. Проверка работоспособности формы
      6. Выводы
   3. Задание 3**(по варианту)**
      1. Постановка задачи
      2. Разработка макета электронной таблицы
      3. Вычисления в электронной таблице
      4. Применение Специальной вставки
      5. Применение Автофильтра
      6. Подведение промежуточных итогов
      7. Построение сводной таблицы
      8. Выводы
   4. Задание 4 **(по варианту)**
      1. Постановка задачи
      2. Разработка математической модели задачи
      3. Разработка макета электронной таблицы
      4. Вычисления в электронной таблице
      5. Графическая интерпретация результатов
      6. Выводы

Заключение **(1 страница)**

Список использованных источников **(нумерованный)**

Краткие требования к оформлению работы приведены в **приложении А**.

Вид титульного листа приведен в **приложении Б**.

Пример описания практического задания приведен **в приложении В**.

Краткие сведения о программировании в среде VBA MS Office Excel приведены **в приложении Г**.

**Дополнительные материалы** приведены в общей папке с заданиями по дисциплине **(изучить перед выполнением работы)**.

# Варианты

Номера заданий по вариантам выбираются из таблицы 1.

Таблица 1 – Варианты заданий

| **Вариант** | **Вопросы** | | **Номера заданий** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **2.1** | **2.2** | **2.3** | **2.4** |
|  | 1 | 17 | 4 | 17 | 1 | 1 |
|  | 2 | 18 | 5 | 16 | 2 | 2 |
|  | 3 | 19 | 6 | 15 | 3 | 3 |
|  | 4 | 20 | 7 | 14 | 4 | 4 |
|  | 5 | 21 | 8 | 13 | 5 | 5 |
|  | 6 | 22 | 9 | 12 | 6 | 6 |
|  | 7 | 23 | 10 | 11 | 7 | 7 |
|  | 8 | 24 | 11 | 10 | 8 | 8 |
|  | 9 | 25 | 12 | 9 | 9 | 9 |
|  | 10 | 26 | 13 | 8 | 10 | 10 |
|  | 11 | 27 | 14 | 7 | 11 | 11 |
|  | 12 | 28 | 15 | 6 | 12 | 12 |
|  | 13 | 29 | 16 | 5 | 13 | 13 |
|  | 14 | 30 | 17 | 4 | 14 | 14 |
|  | 15 | 31 | 18 | 3 | 15 | 15 |
|  | 16 | 32 | 19 | 2 | 16 | 16 |
|  | 17 | 37 | 20 | 1 | 17 | 17 |
|  | 18 | 38 | 21 | 17 | 3 | 18 |
|  | 19 | 39 | 22 | 16 | 4 | 1 |
|  | 20 | 40 | 23 | 15 | 5 | 2 |
|  | 21 | 41 | 24 | 14 | 6 | 3 |
|  | 22 | 42 | 25 | 13 | 7 | 4 |
|  | 23 | 1 | 26 | 12 | 8 | 5 |
|  | 24 | 2 | 27 | 11 | 9 | 6 |
|  | 25 | 3 | 28 | 10 | 10 | 7 |
|  | 26 | 4 | 29 | 9 | 11 | 8 |
|  | 27 | 5 | 30 | 8 | 12 | 9 |
|  | 28 | 6 | 31 | 7 | 13 | 10 |
|  | 29 | 7 | 32 | 6 | 14 | 11 |
|  | 30 | 8 | 33 | 5 | 15 | 12 |
|  | 31 | 9 | 34 | 4 | 16 | 13 |
|  | 32 | 10 | 35 | 3 | 17 | 14 |
|  | 33 | 11 | 36 | 2 | 1 | 15 |
|  | 34 | 12 | 37 | 14 | 2 | 16 |
|  | 35 | 13 | 38 | 13 | 3 | 17 |
|  | 36 | 14 | 39 | 12 | 4 | 18 |
|  | 37 | 15 | 40 | 11 | 5 | 3 |
|  | 38 | 16 | 41 | 10 | 6 | 4 |
|  | 39 | 17 | 42 | 9 | 7 | 5 |
|  | 40 | 18 | 1 | 8 | 8 | 6 |
|  | 41 | 19 | 2 | 7 | 9 | 7 |
|  | 42 | 20 | 3 | 6 | 10 | 8 |
|  | 8 | 21 | 4 | 5 | 11 | 9 |
|  | 11 | 22 | 5 | 4 | 12 | 10 |

# 

# Вопросы для изучения

1. Основные понятия объектно-ориентированного программирования (классы, объекты, свойства, методы, события).
2. Структура проекта VBA в MS Excel. Иерархия объектов MS Excel.
3. Характеристика интегрированной среды разработки VBA.
4. Встроенные диалоговые окна VBA. Синтаксис InputBox, MsgBox.
5. Базовые элементы языка VBA. Константы, переменные, типы данных.
6. Правила описание констант, переменных, массивов в VBA.
7. Выражение VBA и его элементы. Приоритет операций. Правила записи выражений.
8. Понятие и виды инструкций VBA. Инструкции для организации циклов.
9. Понятие и виды инструкций VBA. Организация ветвлений. Встроенные функции выбора.
10. Организация ввода/вывода информации в VBA.
11. Встроенные математические функции VBA.
12. Встроенные функции VBA проверки, определения и преобразования типов.
13. Встроенные строковые функции VBA.
14. Встроенные финансовые функции VBA.
15. Встроенные функции VBA для обработки даты и времени.
16. Отладка программ в VBA. Просмотр значений переменных и выражений.
17. Использование макросов в книгах Excel. Порядок записи и запуска макросов. Структура макроса. Особенности личной книги макросов.
18. Надстройка Пакет анализа в Excel: назначение и краткая характеристика инструментов анализа.
19. Надстройка Пакет анализа в Excel: назначение и краткая характеристика инструментов для моделирования данных (генерация случайных чисел).
20. Надстройка Пакет анализа в Excel: назначение и краткая характеристика инструментов для оценки распределения случайной величины.
21. Надстройка Пакет анализа в Excel: назначение и краткая характеристика инструментов для сглаживания данных.
22. Функции рабочего листа Excel: математические. Использование Мастера функций.
23. Функции рабочего листа Excel: статистические. Использование Мастера функций.
24. Функции рабочего листа Excel: ссылок и массивов. Использование Мастера функций.
25. Функции рабочего листа Excel: даты и времени. Использование Мастера функций.
26. Функции рабочего листа Excel: финансовые. Использование Мастера функций.
27. Функции рабочего листа Excel: текстовые. Использование Мастера функций.
28. Функции рабочего листа Excel: инженерные. Использование Мастера функций.
29. Функции рабочего листа Excel: проверка свойств и значений. Использование Мастера функций.
30. Контроль ввода данных в таблицы Excel. Назначение и особенности применения Условного форматирования в таблицах Excel.
31. Инструменты анализа деловых данных в Excel. Особенности использования надстройки Поиск решения.
32. Инструменты анализа деловых данных в Excel. Особенности использования команды Подбор параметра.
33. Понятие списка (базы данных) в Excel. Основные требования к спискам.
34. Формирование списков Excel при помощи экранной формы. Возможности экранной формы данных.
35. Инструменты для обработки списков (баз данных) в Excel. Фильтрация списков (баз данных) в Excel. Назначение и режимы Автофильтра.
36. Инструменты для обработки списков (баз данных) в Excel. Фильтрация списков (баз данных) в Excel. Назначение и режимы Расширенного фильтра.
37. Анализ данных в списках Excel с помощью сводных таблиц (структура сводной таблицы, порядок ее создания и редактирования).
38. Консолидация данных из списков MS Excel. Режимы консолидации.
39. Использование промежуточных итогов для анализа списков (баз данных) в Excel.
40. Краткая характеристика инструментов графического анализа деловых данных в Excel.
41. Создание сложных документов. Использование закладок и гиперссылок на рабочих листах.
42. Создание и использование пользовательских функций. Структура функции. Порядок вызова.

# Задания практической части

Задания практической части выполняются в одной рабочей книге Microsoft **Excel** на отдельных рабочих листах.

Для каждого задания на рабочем листе записывается постановка задачи, порядок выполнения задания (оформление содержания – самостоятельно).

Описание выполнения заданий в соответствии с постановкой задачи приводится в документе **Word** в отдельных подпунктах (в соответствии со структурой работы). Описание должно включать пояснения в виде связного текста с рисунками (читаемые распечатки (скриншоты, Alt+PrintScr) листов рабочей книги с формулами).

Оформление документа **Word** выполняется с обязательным соблюдением требований ГОСТ 2.105-95.

**Внимание!**

**Файлы РГР (документ Word и рабочая книга Excel) предоставляются до сессии!**

# 2.1 Разработка функции пользователя

**2.1.1 Постановка задачи**

**Для обработки значений двумерного массива разработайте в среде VBA Microsoft Excel пользовательскую функцию (по варианту).**

**Для решения задачи**:

1. Разработайте **алгоритм** решения задачи с учетом проверки корректности исходных данных. Оформите блок-схему алгоритма в соответствии с требованиями ГОСТ 19.701-90 с помощью встроенных средств Microsoft Excel.

2. На основании алгоритма разработайте программный код функции в среде редактора **VBA Microsoft Excel**.

Предусмотрите в программном коде определение размерности массива и проверку корректности исходных данных (на наличие нечисловых значений в массиве).

Для обработки значений массива используйте инструкции цикла.

**3. Подготовьте на рабочем листе Microsoft Excel двумерный массив** исходных данных и ячейку для вызова пользовательской функции.

**4. Выполните проверку работоспособности созданной функции на рабочем листе Microsoft Excel** при различных наборах исходных данных (не менее 2).

5. Сделайте выводы по результатам вычислений.

**Задания по вариантам**

Разработайте пользовательскую функцию для:

1. Нахождения числа нулевых элементов массива.
2. Нахождения суммы положительных элементов массива.
3. Нахождения числа положительных элементов массива.
4. Нахождения доли положительных элементов массива.
5. Нахождения доли отрицательных элементов массива.
6. Нахождения доли нулевых элементов массива.
7. Нахождения суммы квадратов четных элементов массива.
8. Нахождения суммы нечетных элементов массива.
9. Нахождения суммы четных элементов массива.
10. Нахождения числа четных элементов массива.
11. Нахождения числа нечетных элементов массива.
12. Нахождения доли четных элементов массива.
13. Нахождения доли нечетных элементов массива.
14. Нахождения суммы целых элементов массива.
15. Нахождения произведения целых элементов массива.
16. Нахождения суммы дробных элементов массива.
17. Нахождения произведения дробных элементов массива.
18. Нахождения суммы квадратов элементов массива в четных столбцах.
19. Нахождения суммы элементов массива в четных столбцах.
20. Нахождения суммы квадратов дробных элементов массива.
21. Нахождения квадрата минимального элемента массива.
22. Нахождения квадрата максимального элемента массива.
23. Нахождения суммы квадратов отрицательных элементов массива.
24. Нахождения суммы элементов массива, стоящих на четных строках.
25. Нахождения суммы квадратов элементов массива, стоящих на нечетных строках.
26. Нахождения суммы отрицательных элементов массива.
27. Нахождения суммы квадратов положительных элементов массива.
28. Нахождения числа отрицательных элементов массива.
29. Нахождения суммы элементов массива, стоящих на нечетных строках.
30. Нахождения суммы элементов массива в нечетных столбцах.
31. Нахождения суммы квадратов элементов массива в нечетных столбцах.
32. Нахождения суммы квадратов нечетных элементов массива.
33. Нахождения произведения положительных элементов массива.
34. Нахождения произведения отрицательных элементов массива.
35. Нахождения произведения четных элементов массива.
36. Нахождения произведения нечетных элементов массива.
37. Нахождения максимального элемента массива.
38. Нахождения минимального элемента массива.
39. Нахождения числа элементов массива, не кратных n.
40. Нахождения суммы элементов массива, не кратных n.
41. Нахождения суммы квадратов элементов массива, не кратных n.
42. Нахождения произведения элементов массива, не кратных n.

# 2.2 Разработка формы в VBA MS Excel

**2.2.1 Постановка задачи**

В редакторе VBA разработайте макет формы (и соответствующий программный код) для выполнения линейных вычислений по варианту.

1. При разработке макета предусмотрите в области формы следующие элементы:

* **надпись** с постановкой задачи,
* **рисунок**-иллюстрацию в соответствии с постановкой задачи,
* **текстовые поля для ввода** исходных данных и пояснительные надписи к ним,
* **текстовое поле** для вывода результата вычислений с пояснительной надписью,
* **командные кнопки** Расчет, Очистить, Отмена,
* **дополнительно предусмотрите вывод информации об авторе** (навкладке Об авторе или по нажатию кнопки Об авторе, приведите фото и контактные данные).

**Дизайн и расположение элементов формы определите самостоятельно.**

1. Измените свойства формы:

* **имя формы** – *Famila\_studenta* (латиница!),
* **заголовок формы** - *№ варианта-Фамилия студента*.

1. **Разработайте алгоритм** решения задачи с учетом проверки корректности исходных данных (блок-схему алгоритма приведите на листе).
2. **Разработайте программный код для каждой кнопки формы** (предусмотрите вывод сообщения об ошибке при попытке вычислений без исходных данных или вводе некорректных исходных данных: нечисловых, отрицательных или нулевых значений).
3. **Для вызова пользовательской формы** на листе расположите **кнопку (имя определите самостоятельно), запишите соответствующий макрос для кнопки.**
4. **Выполните проверку работоспособности разработанной формы** (при разных наборах исходных данных, включая некорректные)
5. Сделайте выводы.

**Задания по вариантам**

Разработайте форму, с помощью которой:

1. По внешнему и внутреннему радиусам кольца *R,r* определите его площадь .



1. По внешнему и внутреннему радиусам кольца *R,r* ицентральному углу ϕ определите площадь части кольца.



1. По радиусу *R* определите площадь поверхности шара .



1. По радиусу *R* определите объем шара



1. По радиусу цилиндра *R* и высоте *H* определите его объем *V = πR2H.*
2. По радиусу цилиндра *R* и высоте *H* определите площадь поверхности   
   *S = 2πR2+2πRH.*
3. По внешнему и внутреннему радиусам цилиндрической трубы *R,r* и высоте *H* определите ее объем .



1. По радиусу основания *R* и образующей конуса *L* определите площадь боковой поверхности конуса .



1. По радиусу основания *R* и образующей конуса *L* определите площадь полной поверхности конуса .



1. По радиусу основания *R* и высоте *H* определите объем конуса



1. По радиусам оснований усеченного конуса *R, r* и образующей *L* определите площадь боковой поверхности усеченного конуса .



1. По радиусам оснований усеченного конуса *R, r* и образующей *L* определите площадь полной поверхности усеченного конуса .



1. По радиусам оснований усеченного конуса *R, r* и высоте *H* определите объем усеченного конуса



1. По радиусам образующей и сечения тора *R,r* определите площадь поверхности тора



1. По радиусам образующей и сечения тора *R,r* определите объем тора



1. По заданному радиусу *R* и высоте *H* определите объем шарового сектора



1. По радиусу *R* и высоте шарового сегмента *H* определите объем шарового сегмента .



# 2.3 Создание пользовательской электронной таблицы

**2.3.1 Постановка задачи**

В рабочей книге MS Office Excel создайте пользовательскую таблицу (список, БД) (номер задания – по варианту).

1. При заполнении таблицы учитывайте следующее:

* в таблице должно быть не менее 20 записей;
* при заполнении поля **Фамилия…** первая фамилия в списке – фамилия студента, остальные – фамилии одногруппников (не менее 5 фамилий студентов, **фамилии в списке должны повторяться не менее 4 раз)**;
* к ячейкам в столбцах примените операцию **Проверка данных**, для ограничения допустимых значений при вводе (условия проверки задайте самостоятельно). В документе Word опишите порядок выполнения действий и приведите результат;
* поля с числовыми значениями и датами заполните произвольно, отформатируйте с применением соответствующих числовых форматов. Поля с датами должны включать значения в пределах одного года, но разных месяцев, хотя бы по два значения для месяца;
* последний столбец в созданной таблице должен содержать **формулу для расчета** (формулу записать самостоятельно по условию задачи, применить соответствующий числовой формат к результату).

2. После заполнения таблицы-списка выполните следующие действия:

* с помощью команды **Специальная вставка** увеличьте/уменьшите (нечетный/четный вариант) на (Nвар+20)% все числовые значения в одном столбце с числовыми значениями (по выбору, за исключением столбца с формулой). Опишите процедуру, отобразите результат;
* скопируйте таблицу на новый лист и примените по одному из полей **Автофильтр** (условие для фильтрации задайте самостоятельно, опишите процедуру, отобразите результат);
* выполните **сортировку** записей в таблице, например, по столбцу Фамилия…(столбец для сортировки и ее порядок задайте самостоятельно);
* выполните подведение **промежуточных итогов** в таблице, например, по полю Фамилия… (условие подведения промежуточных итогов задайте самостоятельно, опишите процедуру, отобразите результат);
* выполните построение **Сводной таблицы** для анализа данных, например, по срезу Фамилия (условие построения сводной таблицы сформулируйте самостоятельно, опишите процедуру, приведите макет сводной таблицы, отобразите результат, например, для собственной фамилии. Самостоятельно измените структуру сводной таблицы, приведите результат, поясните).

**1. Таблица «Учет почасовой оплаты» должна содержать следующие поля:**

1. Дата работы.
2. Фамилия сотрудника.
3. Должность сотрудника.
4. Смена.
5. Почасовая ставка сотрудника.

6. Отработано часов.

**7. Сумма оплаты.**

**2. Таблица «Учет торговых операций» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия продавца.
3. Категория товара.
4. Модель товара.
5. Цена товара.
6. Количество проданного товара.
7. Скидка, %.
8. **Общая стоимость заказа.**

**3. Таблица «Учет продаж в автосалоне» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия сотрудника автосалона.
3. Марка автомобиля.
4. Модель автомобиля.
5. Заводская цена.
6. Транспортные издержки.
7. Скидка, %.
8. ***Общая стоимость.***

**4. Таблица «Учет проката видеокассет» должна содержать следующие поля:**

1. Дата выдачи видеокассеты.
2. Дата возврата видеокассеты (плановая).
3. Дата возврата видеокассеты (фактическая).
4. Жанр видеокассеты
5. Наименование видеокассеты.
6. Фамилия сотрудника видеосалона.
7. Стоимость проката за сутки.
8. ***Общая стоимость проката.***

**5. Таблица «Учет продаж туристических путевок» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия туроператора.
3. Название маршрута.
4. Страна.
5. Длительность маршрута (в днях).
6. Цена 1-го дня.
7. Скидка, %.
8. **Общая стоимость путевки.**

**6. Таблица «Учет заказов салона проката спортинвентаря» должна содержать следующие поля:**

1. Дата выдачи спортинвентаря.
2. Дата возврата спортинвентаря (плановая).
3. Дата возврата спортинвентаря (фактическая).
4. Фамилия клиента.
5. Категория спортинвентаря.
6. Наименование спортинвентаря.
7. Стоимость проката за сутки.
8. **Общая стоимость проката.**

**7. Таблица «Учет заказов на доставку обедов» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия курьера.
3. Название обеда.
4. Цена 1-й порции.
5. Количество заказанных порций.
6. Скидка, %.
7. Стоимость доставки.
8. **Общая сумма заказа.**

**8. Таблица «Учет заказов фотосервиса» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия сотрудника.
3. Наименование фотоуслуги.
4. Количество единиц заказа.
5. Цена за единицу.
6. Надбавка за срочность исполнения, %.
7. Скидка, %.
8. **Общая стоимость заказа.**

**9. Таблица «Учет выдачи учебно-методической литературы» должна содержать следующие поля:**

1. Дата выдачи книги.
2. Дата возврата книги плановая.
3. Дата возврата книги фактическая.
4. Название книги.
5. Автор книги.
6. Фамилия библиотекаря.
7. Сумма штрафа за сутки.
8. **Общая сумма штрафа (если просрочен возврат книги).**

**10. Таблица «Учет коммунальных платежей» должна содержать следующие поля:**

1. Дата оплаты.
2. Фамилия бухгалтера.
3. Номер лицевого счета.
4. Количество жильцов.
5. Площадь квартиры, кв.м.
6. Название платежа.
7. Стоимость услуги на квадратный метр площади.
8. Стоимость услуги на 1 жильца.
9. **Общая сумма оплаты.**

**11. Таблица «Учет заказов салона проката бытовой техники» должна содержать следующие поля:**

1. Дата выдачи техники.
2. Дата возврата техники (плановая).
3. Дата возврата техники (фактическая).
4. Фамилия сотрудника.
5. Категория техники.
6. Наименование техники.
7. Стоимость проката за сутки.
8. **Общая стоимость проката.**

**12. Таблица «Учет стоимости отработки пропущенных занятий» должна содержать следующие поля:**

1. Дата занятия.
2. Фамилия преподавателя.
3. Фамилия студента.
4. Название предмета.
5. Вид занятия (лекция, лабораторная работа, практическое занятие, семинар).
6. Количество часов пропусков.
7. Стоимость 1 часа отработки.
8. **Общая стоимость отработки.**

**13. Таблица «Учет заказов на доставку пиццы» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия курьера.
3. Название пиццы.
4. Цена 1-й порции пиццы.
5. Количество заказанных порций.
6. Скидка, %.
7. Стоимость доставки.
8. **Общая сумма заказа.**

**14. Таблица «Учет текущей успеваемости студентов» должна содержать следующие поля:**

1. Дата аттестации.
2. Фамилия студента.
3. Фамилия преподавателя.
4. Название предмета.
5. Вид аттестации.
6. Текущая рейтинговая оценка.
7. Максимальная рейтинговая оценка.
8. **Рейтинг, %.**

**15. Таблица «Учет мобильных телефонных разговоров сотрудников» должна содержать следующие поля:**

1. Дата разговора
2. Фамилия сотрудника.
3. Подразделение.
4. Название оператора.
5. Телефонный номер вызова.
6. Стоимость за 1 минуту разговора.
7. Продолжительность разговора, мин.
8. **Стоимость звонка.**

**16. Таблица «Учет бронирование мест в гостинице» должна содержать следующие поля:**

1. Дата заказа.
2. Фамилия администратора.
3. Номер комнаты.
4. Тип комнаты (люкс, 2-, 3-местный).
5. Стоимость проживания за 1 сутки.
6. Срок проживания.
7. Доплата за бронь, %.
8. **Общая стоимость проживания.**

**17. Таблица «Учет операций по банковским вкладам» должна содержать следующие поля:**

1. Дата операции.
2. Фамилия оператора.
3. Название вклада.
4. Вид вклада (рублевый, валютный).
5. Первоначальная сумма вклада.
6. Годовая процентная ставка.
7. Срок вклада, дней.
8. **Сумма дохода к выплате.**

# 2.4 Решение задачи математического программирования средствами MS Excel

**2.4.1 Постановка задачи**

Используя надстройку Microsoft Excel Поиск решения решите задачу оптимизации (по варианту).

1. Для решения задачи разработайте **математическую модель** задачи оптимизации:
   1. Определите и обозначьте искомые переменные.
   2. Запишите формулу целевой функции (критерия оптимизации, как зависимость от искомых переменных)
   3. Запишите ограничения (по варианту) и граничные условия для искомых переменных.
2. На основании математической модели постройте **макет таблицы** для решения задачи (приведите макет с отображением всех формул).
3. Примените надстройку **Поиск решения** (приведите окно Поиск решения с заполненными полями и параметрами).
4. Приведите в отчете таблицу с полученными результатами после нажатия кнопки **Выполнить** в окне **Поиск решения**.
5. Сохраните отчет по результатам надстройки **Поиск решения** (приведите распечатку листа **Отчет по результатам**).
6. Выполните графическое отображение целевой функции.
7. Выполните анализ полученных результатов.

**Задания по вариантам**

1. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), объемом 16000 см3, чтобы длина сварного шва при его изготовлении была минимальной.
2. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), объемом 8000 см3, чтобы длина сварного шва при его изготовлении была минимальной.
3. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), объемом 18000 см3, чтобы стоимость сварки при его изготовлении была минимальной. Цена 1 м сварного шва равна 5 у.е.
4. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), объемом 8000 см3, чтобы стоимость сварки при его изготовлении была минимальной. Цена 1 м сварного шва равна 3 у.е.
5. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), объемом 16000 см3, чтобы площадь материала, пошедшего на его изготовление, была минимальной.
6. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), объемом 18000 см3, чтобы площадь материала, пошедшего на его изготовление, была минимальной.
7. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), объемом 18000 см3, чтобы стоимость материала, пошедшего на его изготовление, была минимальной. Стоимость 1 м2 материала равна 14 у.е.
8. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), объемом 28000 см3, чтобы стоимость материала, пошедшего на его изготовление, была минимальной. Стоимость 1 м2 материала равна 20 у.е.
9. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), чтобы объем был максимальный, а стоимость материала, пошедшего на его изготовление, не превышала 2000 у.е. Цена 1 м2 материала равна 10 у.е.
10. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), чтобы объем был максимальный, а стоимость материала, пошедшего на его изготовление, не превышала 1600 у.е. Цена 1 м2 материала равна 10 у.е.
11. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), чтобы объем был максимальный, а стоимость сварки при его изготовлении не превышала 1600 у.е. Цена 1 м сварного шва равна 8 у.е.
12. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), чтобы объем был максимальный, а стоимость сварки при его изготовлении не превышала 1200 у.е. Цена 1 м сварного шва равна 4 у.е.
13. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), чтобы объем был максимальный, а площадь материала, пошедшего на его изготовление, не превышала 26000 см2.
14. Определите размеры бака (a, b, h), имеющего форму прямоугольного параллелепипеда (с крышкой), чтобы объем был максимальный, а стоимость сварки при его изготовлении не превышала 800 у.е. Цена 1 м сварного шва равна 4 у.е. Бак изготавливается из одной развертки.
15. Определите размеры бака (R, h) имеющего форму цилиндра (с крышкой), чтобы объем был максимальный, а площадь материала, пошедшего на его изготовление, не превышала 84000 см2.
16. Определите размеры бака (R, h), имеющего форму цилиндра (без крышки), чтобы объем был максимальный, а длина сварного шва при его изготовлении не превышала 2400 см. Размеры должны быть целыми числами.
17. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), чтобы объем был максимальный, а длина сварного шва при его изготовлении не превышала 2800 см. Размеры должны быть целыми числами.
18. Определите размеры бака (R, h), имеющего форму цилиндра (с крышкой), объемом 100 м3, чтобы стоимость материала, пошедшего на его изготовление, была минимальной. Стоимость 1 м2 материала равна 20 у.е.

**Список литературы**

**Основная литература**

1. **Гарнаев, А.Ю. Excel, VBA, Internet в экономике и финансах / А.Ю. Гарнаев. – Санкт-Петербург: БХВ-Петербург, 2005. – 796 с.**
2. Информатика: базовый курс: учебное пособие для втузов / под ред. Симоновича С.В. – 3-е изд. – Санкт-Петербург [и др.]: Питер, Санкт-Петербург: Питер, 2012. – 637 с.

**Дополнительная литература**

1. Гарнаев, А.Ю. Microsoft Office Excel 2010: разработка приложений / А.Ю. Гарнаев, Л.В. Рудикова. – Санкт-Петербург: БХВ-Петербург, 2011. –528 с.
2. Карлберг, К. Бизнес-анализ с помощью Microsoft Excel / К. Карлберг.  
    – 2-е изд. – Москва: Вильямс, 2003. – 446 с.
3. Курицкий, Б.Я. Поиск оптимальных решений средствами Excel / Б.Я.Курицкий. – СПб.: BHV, 1997. – 384 с.
4. Попов, А.А. Excel: практическое руководство: Учебное пособие для вузов / А.А. Попов. – 2-е изд. – Москва: ДЕСС КОМ, 2001. – 301 с.
5. Стоцкий, Ю.А. Microsoft Office 2010: самоучитель / Ю.А. Стоцкий, А.Васильев, И.Телина. – Санкт-Петербург [и др.]: Питер, 2011. – 425 с.
6. Уокенбах, Джон. Excel 2010: лучшие трюки Джона Уокенбаха / Джон Уокенбах, пер. Дубенок И. – СПб. [и др.]: Питер, 2011. – 464 с.
7. Шакель, Е.В. Табличный процессор Microsoft Excel 2010: учеб.-методическое пособие / Е.В. Шакель, под ред. С. И. Максимова. – Минск: РИВШ, 2013. – 131 с.

**Электронные ресурсы**

1. Автоматизация в Excel - бесплатные уроки. [Электронный ресурс]. – Режим доступа: <http://www.excel-eto-prosto.ru/>
2. Джин Желязны. Говори на языке диаграмм [Электронный ресурс]. – Режим доступа: http://www.publicant.ru/Demo/553035.htm
3. Интерактивное руководство пользователя по работе с командами в версиях Excel 2003 и Excel 2007 - Excel - Office.com [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/excel-help/HA010149151.aspx>
4. НОУ ИНТУИТ | Учебный курс | Работа в Microsoft Excel 2010. Автор: [Олег Спиридонов](http://www.intuit.ru/intuituser/userpage/o_v_spiridonov).[Электронный ресурс]. – Режим доступа: [http://www.intuit.ru/ studies/courses/613/469/info](http://www.intuit.ru/%20studies/courses/613/469/info)
5. Обучение - Microsoft Office [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/training/>
6. Планета Excel :: Главная. [Электронный ресурс]. – Режим доступа: <http://planetaexcel.ru/>
7. INTUIT.ru: Курс: Введение в математическое программирование. Автор: [Ю.В. Губарь](http://www.intuit.ru/lector/197.html). Лекция №3: Математическое программирование. Линейное программирование... [Электронный ресурс]. – Режим доступа: [http://www.intuit.ru/studies/courses/ 1020/188/lecture/4917](http://www.intuit.ru/studies/courses/%201020/188/lecture/4917)

**Стандарты**

1. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения: ГОСТ 19.701–90. – Введен 01.01.1992.
2. Единая система конструкторской документации. Общие требования к текстовым документам: ГОСТ 2.105–95. – Введен 01.07.1996.

# Приложение А Краткие требования к оформлению расчетно-графической работы

1. Объем работы ~ 30 печатных страниц формата А4.
2. Оформление работы выполняется при помощи вычислительной техники с соблюдением требований ГОСТ 2.105-95.
3. Размер бумаги А4, ориентация бумаги – книжная.
4. Размеры полей:

левое – 3 см;

правое – 1 см;

верхнее – 2,5 см;

нижнее – 2 см

1. Рекомендуемый шрифт – **Times New Roman Cyr**. Размер шрифта 12 пт. Для выделения заголовков – полужирный.
2. Установить межстрочный интервал, при котором на странице не более 39-40 строк.
3. Абзацный отступ в тексте - 1,25 см.
4. Выравнивание текста – по ширине.
5. Заголовки разделов и подразделов нумеровать и располагать с абзацного отступа.
6. Заголовки **Содержание**, **Введение**, **Заключение** и **Список использованных источников** не нумеровать, записывать симметрично тексту (по центру).
7. В тексте работы обязательны ссылки на литературу, по образцу: [1].
8. В тексте работы обязательны рисунки и таблицы с соответствующими ссылками. Например, … на рисунке 1 представлено …, … в таблице 1 приведены…
9. **Для всех заданий практической части приводить экранные копии макетов таблиц с формулами, заголовками строк и столбцов**, **со строкой заголовка рабочей книги**.
10. Номера страниц (арабские цифры без точки) располагаются в правом верхнем углу страницы.
11. Работа сшивается по левому краю.
12. К защите работы предоставить **2 файла**:

* текст работы (документ Word);
* рабочую книгу Excel (задания практической части выполняются на отдельных листах).

1. **Имена файлов – Вариант\_ФамилияСтудента\_2**, например, 13\_Иванов\_2.хххх
2. **К экзамену предоставляется краткий конспект ответов на вопросы.**

# Приложение Б Титульный лист работы

Министерство образования Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра "Экономика и организация машиностроительного производства"

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

по дисциплине «Информатика»

на тему «\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*»

| **Вариант** | Вопросы | | Задания | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 2.1 | 2.2 | 2.3 | 2.4 |
| **хх** | 1 | 21 | 1 | 1 | 1 | 1 |

Выполнил: студент гр. 10302117

Иванов И.И.

Проверил: ст. преподаватель

Лавренова О.А.

**2018**

# Приложение В Пример оформления задания 2 в документе Word

# 2 Разработка пользовательской формы

## *2.1 Постановка задачи*

Разработать пользовательскую форму, реализующую линейные вычисления периметра квадрата по радиусу описанной окружности: .

Предусмотреть контроль вводимых исходных данных с выводом предупреждающих сообщений в случае некорректных данных или их отсутствия.

Выполнить отладку и проверку работоспособности пользовательской формы при различных исходных данных (включая некорректные данные).

## *2.2 Разработка макета формы*

Чтобы создать новую пользовательскую форму в VBA необходимо в окне редактора выполнить команду **Insert(Вcтавить)/UserForm(Форма)**.

Затем непосредственно в режиме Конструктора формы (рисунок В.1) необходимо изменить размеры формы, в окне **Properties** (Свойства) задать заголовок формы (свойство Caption) и другие необходимые параметры.

Элементы управления (**ЭУ**) формы (таблица В.1) добавляются в режиме Конструктора формы с панели инструментов Toolbox с помощью мыши.

Настройка свойств выбранного ЭУ выполняется в соответствующем окне **Properties** (Свойства) ЭУ.

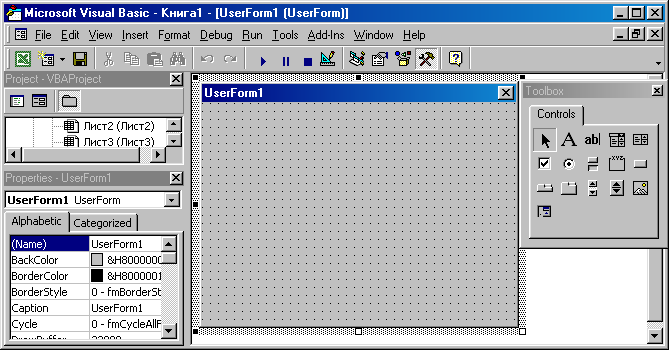


Рис. В.1. Создание формы

Таблица В.1

Характеристики элементов управления

|  |  |  |  |
| --- | --- | --- | --- |
| ***Элемент управления*** | ***Кнопка*** | ***Префикс  в имени*** | ***Назначение*** |
| ***Надпись*** | *Label* | ***lbl*** | Отображение произвольного текстового выражения (пояснения) |
| ***Поле*** | *TextBox* | ***txt*** | Ввод текстовой информации, которая может преобразовываться в числа и даты |
| ***Поле  со списком*** | *ComboBox* | ***cbo*** | Хранение списка значений, из которого пользователь может выбрать или ввести с клавиатуры только одно значение |
| ***Список*** | *ListBox* | ***lst*** | Хранение списка значений, из которого пользователь может выбрать одно или несколько значений |
| ***Флажок*** | *CheckBox* | ***chk*** | Выбор из нескольких возможных вариантов |
| ***Переключатель*** | *OptionButton* | ***opt*** | Выбор одного из нескольких взаимоисключающих вариантов |
| ***Frame*** | *Рамка* |  | Визуальная группировка элементов управления. |
| ***Выключатель*** | *ToggleButton* | ***tgl*** | Выбор из двух альтернатив |
| ***Кнопка*** | *CommandButton* | ***cmd*** | Выполнение некоторых действий при нажатии |
| ***Набор вкладок*** | *TabStrip* |  | Создание нескольких вкладок в диалоговом окне |
| ***Набор страниц*** | *MultiPage* |  | Создание многостраничных окон |
| ***Полоса прокрутки*** | *ScrollBar* | ***scr*** | Скроллинг (прокручивание) информации, не умещающейся в границы окна |
| ***Счетчик*** | *SpinButton* | ***spn*** | Изменение числовых величин (+/-) на заданный шаг |
| ***Рисунок*** | *Image* | ***img*** | Отображение в форме графических файлов форматов: BMP, GIF, .JPG, ICO, WMF. |
| ***Редактор ссылок*** | *RefEdit* |  | Ввод ссылок на ячейки и диапазоны |

**Исходные данные** задаются посредством элементов управления: ***Поле, Поле со списком, Список, Счетчик, Полоса прокрутки.***

**Пояснительные надписи** на форме осуществляются посредством элемента управления ***Надпись****.*

**Для выбора действий** используются элементы управления: ***Флажок, Переключатель, Выключатель****.*

**Выполнение и завершение макросов** осуществляется посредством элемента управления ***Кнопка****.*

**Для вставки рисунков** используется элемент управления ***Рисунок****.*

**Для работы с вкладками и страницами** используются элементы управления ***Набор Вкладок*** и ***Набор Страниц****.*

**Выходные данные** отображаются посредством элементов управления ***Поле, Поле со списком, Список.***

Пример макета формы приведен на рисунке В.2.

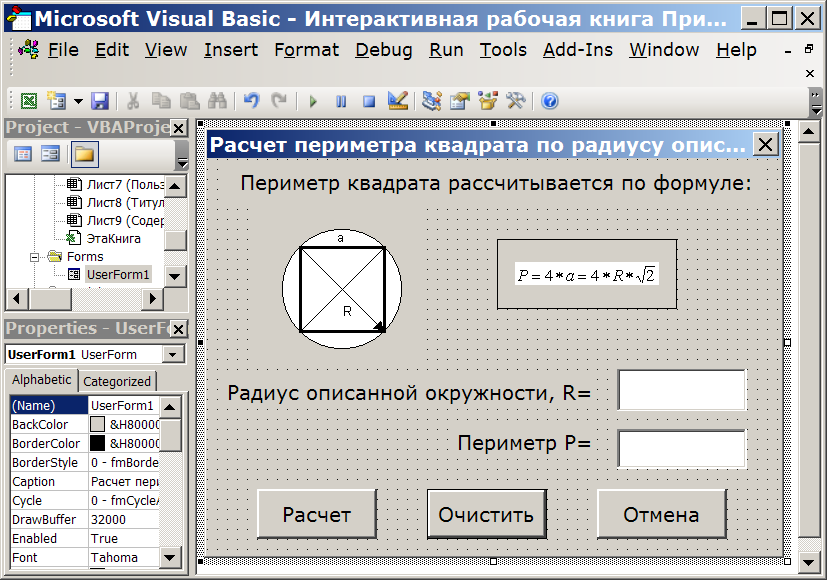


Рис. В.2. Пример макета формы пользователя

Пояснительная информация в форме размещена с помощью элементов управления Надпись – Label1, Label2, Label3.

Иллюстрации к решению задачи добавлены при помощи элемента управления Рисунок – Image1, Image2.

Для ввода исходных данных предусмотрен элемент управления Поле – TextBox1 (радиус R).

Для вывода результата вычислений предусмотрено Поле TextBox2 (периметр Р).

Вычисление должно выполняться после нажатия кнопки Расчет.

При нажатии кнопки Очистить должны удаляться значения в полях TextBox1 и TextBox2 (без закрытия формы).

При нажатии кнопки Отмена должно происходить закрытие формы.

## *2.3 Разработка программного кода*

После разработки макета формы необходимо выделить командную кнопку **Расчет** (CommandButton1) и выполнить команду **View (Вид)/Code (Код).** При этом откроется окно кода с командами начала и завершения процедуры CommandButton1\_Click() – нажатия кнопки **Расчет** (рисунок В.3).

Между указанными командами записываются необходимые вычисления.

Аналогично создается программный код для кнопок **Отмена** и **Очистить** (CommandButton2 и CommandButton3, рисунок В.4).

В тексте программного кода необходимо предусмотреть проверку вводимых исходных данных и обработку ошибок при попытке нажатия кнопки Расчет без исходных данных.

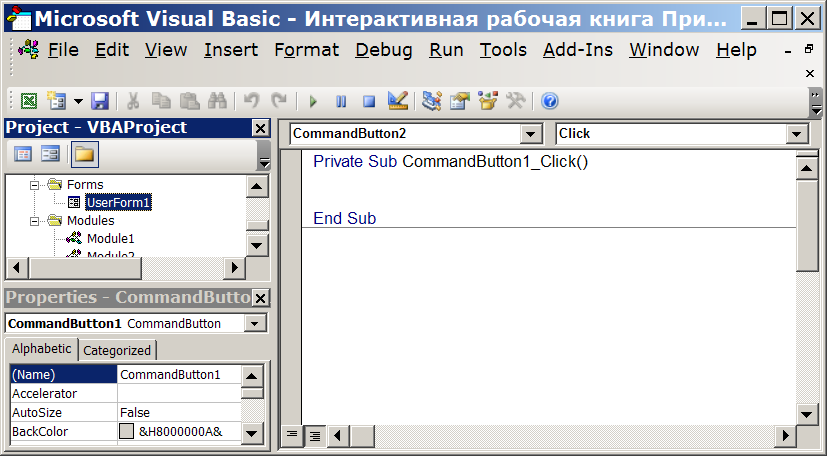


Рис. В.3. Шаблон программного кода процедуры для нажатия кнопки

Окончательный вид макета формы и код процедуры, позволяющей по радиусу описанной окружности вычислить периметр квадрата по формуле: , приведены на рисунке В.5.

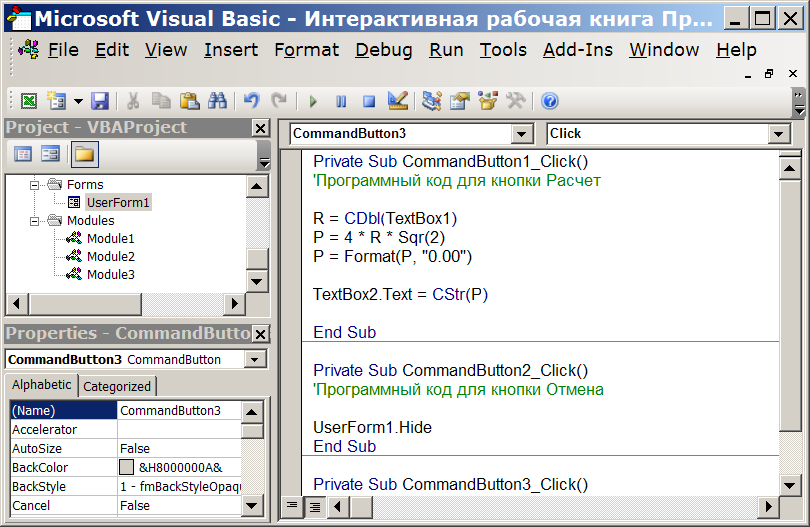


Рис. В.4. Программный код процедур нажатия кнопок Расчет и Отмена

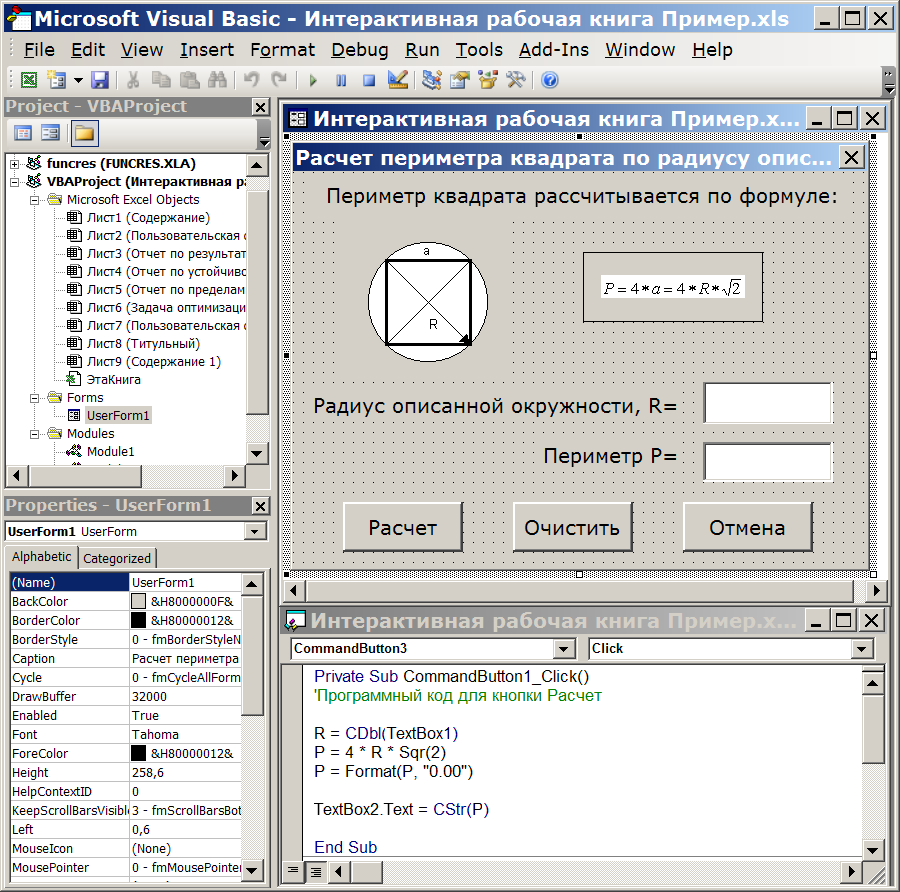


Рис. В.5. Макет и программный код пользовательской формы

Для ввода исходных данных (радиус R) использован элемент поле TextBox1, значение из которого преобразуется к типу Double.

Вычисления выполняются после нажатия кнопки **Расчет** (CommandButton1).

Вывод результата осуществляется в поле TextBox2 (значение периметра приводится к числовому формату с двумя знаками после запятой и преобразуется к типу String).

Нажатие кнопки **Отмена** (CommandButton2) приведет к очистке от текущих значений текстовых полей TextBox1, TextBox2 и закрытию формы.

Нажатие кнопки **Очистить** (CommandButton3) приведет к очистке от текущих значений текстовых полей TextBox1 и TextBox2 (без закрытия формы).

## *2.4 Проверка работоспособности пользовательской формы*

Отладка программного кода формы осуществляется командой **Debug (Отладка)/ Compile VBAProject (компилировать).**

Проверка работоспособности разработанной формы производится путем ее вызова при нажатии кнопки на рабочем листе Excel.

Для создания кнопки вызова формы используется панель инструментов Формы.

Макрос для кнопки вызова формы имеет вид (рисунок В.6).

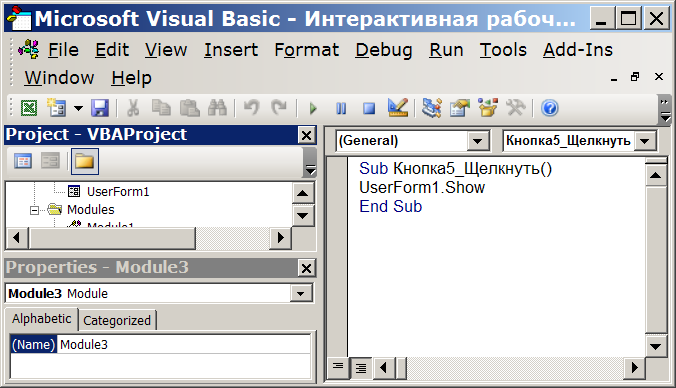


Рис. В.6. Процедура для кнопки вызова формы

Пример вызова на листе Excel разработанной формы, включая:

* + окно рабочей книги с постановкой задачи и кнопкой вызова формы (рисунок В.7);

окно формы с исходными данными и полученным результатом вычисления.

Например, на рисунках В.8 и В.9 приведена форма до и после нажатия кнопки **Расчет**;

* + диалоговое окно, которое выводится в случае попытки расчета при отсутствии исходных данных или отрицательных значениях (рисунок В.10).

Программные коды процедур для кнопок необходимо привести в виде обычного текста на рабочем листе Excel.

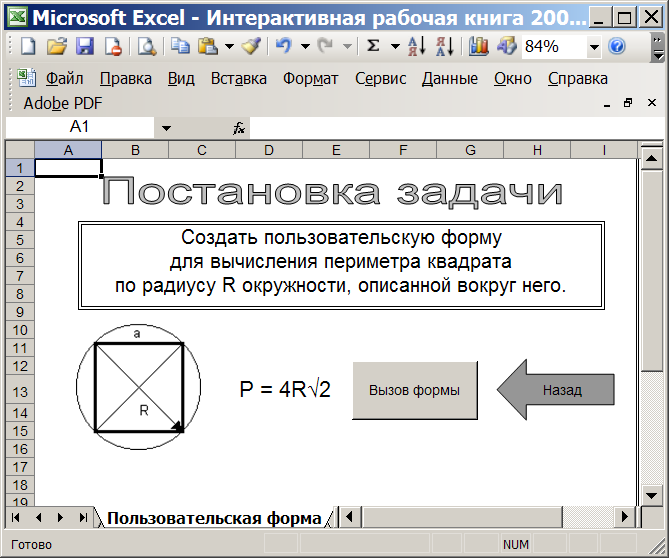


Рис. В.7. Кнопка вызова формы

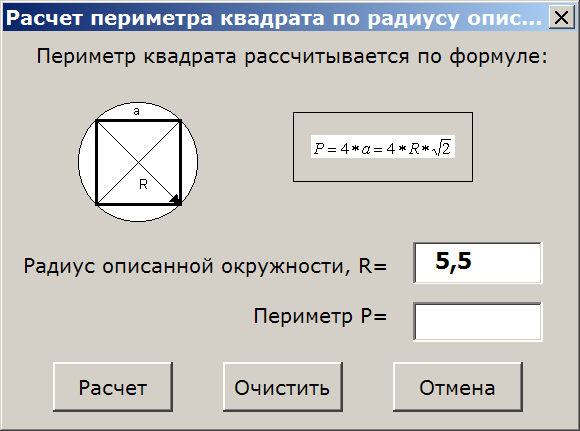


Рис. В.8. Ввод исходных данных

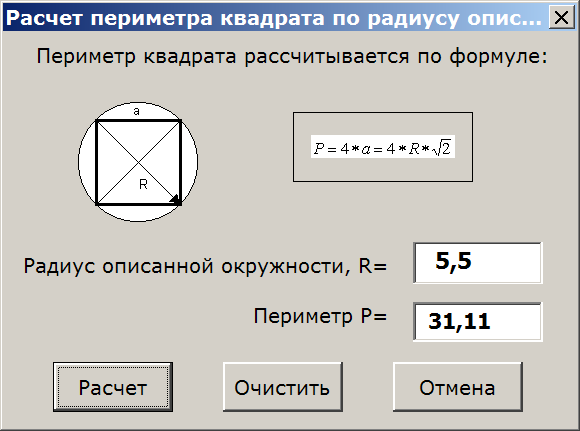


Рис. В.9. Вывод результата после нажатия кнопки Расчет

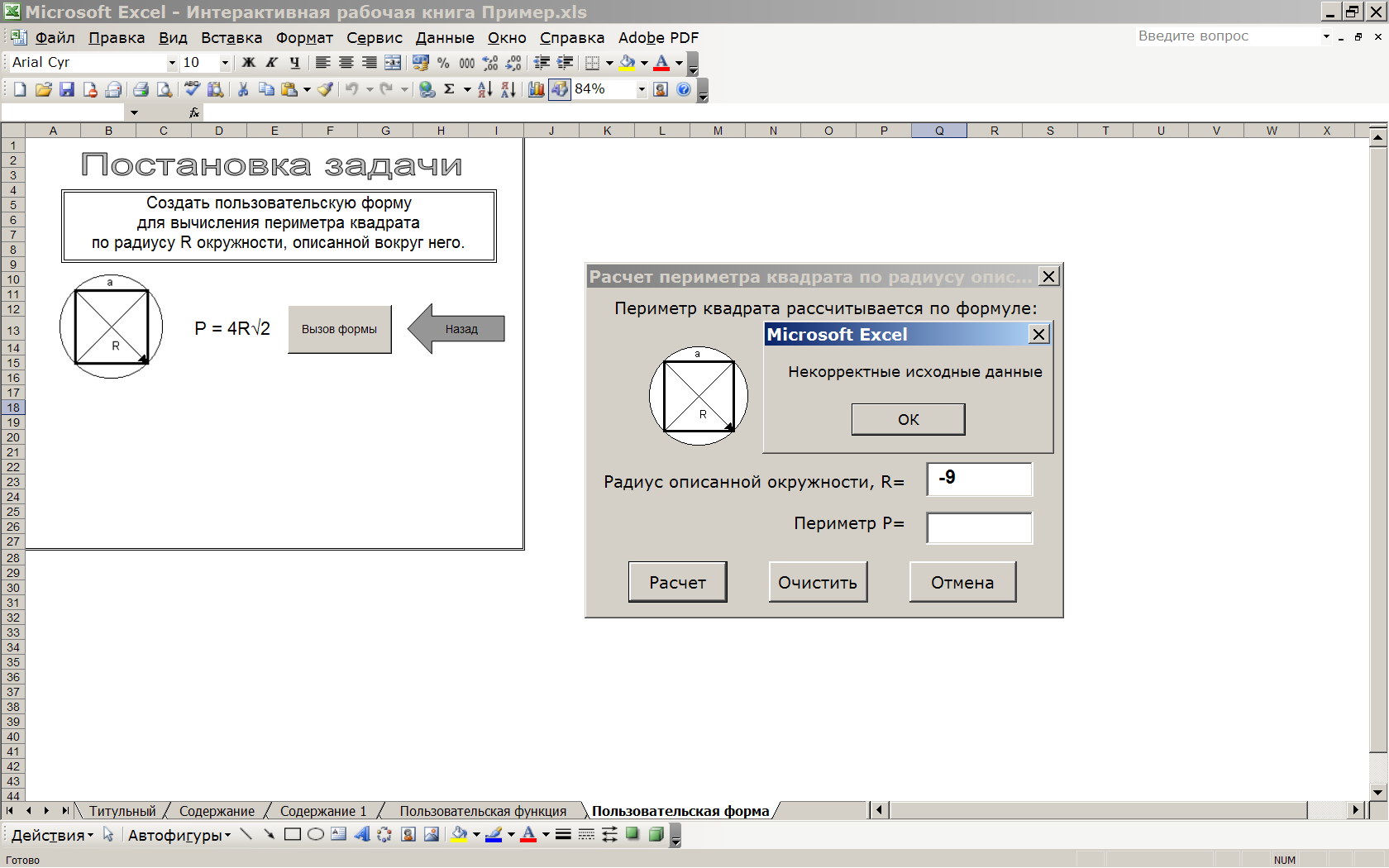
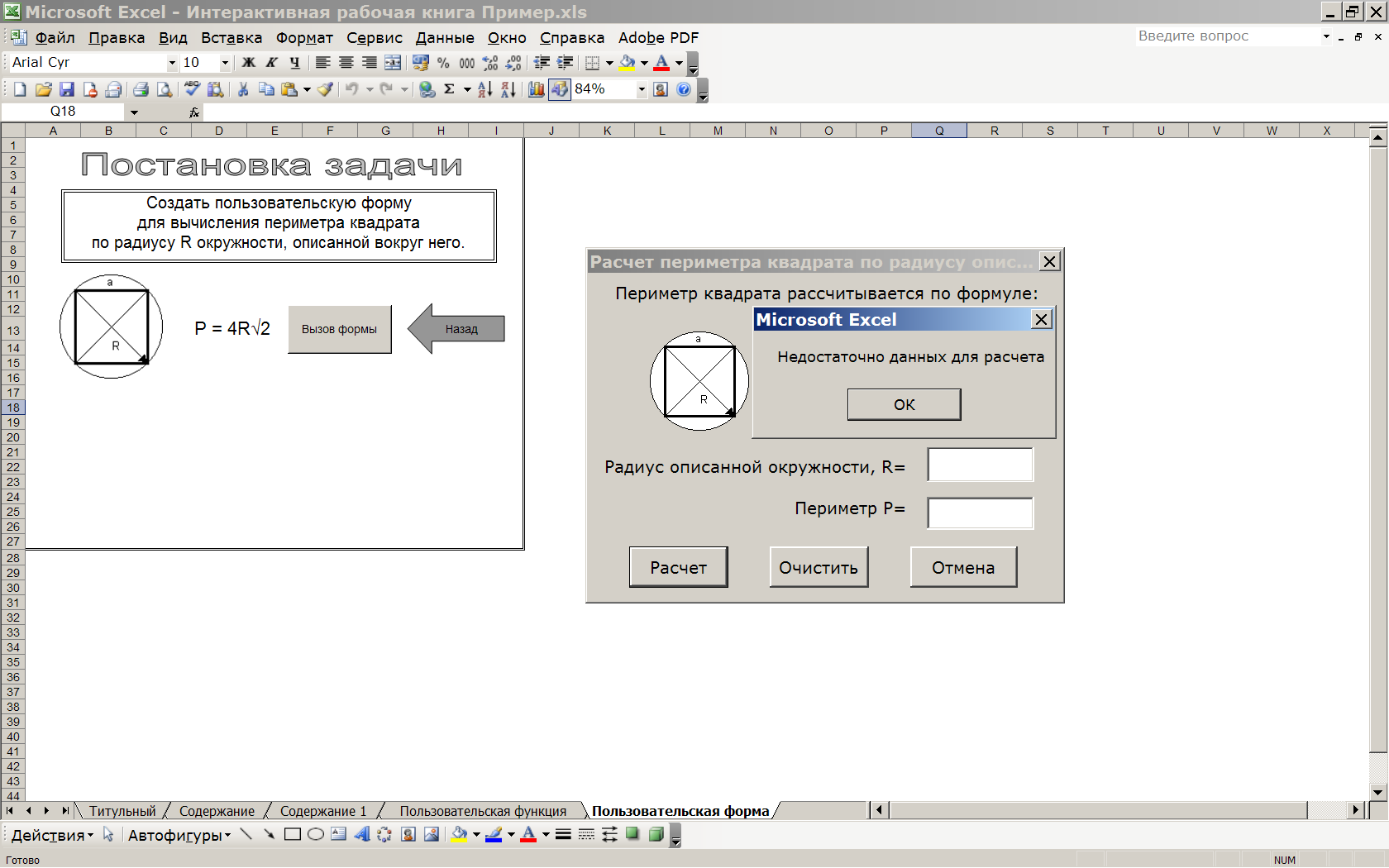


Рис. В.10. Вывод сообщений при нажатии кнопки Расчет   
при отсутствии или некорректных исходных данных в поле TextBox1

**Приложение Г  
Краткие теоретические сведения о программировании   
в среде VBA Microsoft Office Excel**

**1 Основные понятия языка VBA**

**VBA** относится к языкам объектно-ориентированного программирования, которые можно описать как методику анализа, проектирования и написания приложения с помощью объектов.

В **VBA** имеется более 100 встроенных объектов:

1. **Application** – приложение Excel, имеет более 120 свойств и 40 методов, позволяет вызывать более 400 функций рабочего листа при помощи конструкции вида: Application.ФункцияРабочегоЛиста(Аргументы).
2. **Woorkbook** – рабочая книга, свойства и методы рабочей книги позволяют работать с файлами.
3. **Woorksheet** – рабочий лист.
4. **Range** – диапазон.
5. **Selection** – выбор, возникает либо как результат работы метода **Select**, либо при вызове свойства **Selection;** при работе с ним можно использовать свойства и методы объекта **Range**.
6. **Userform** – пользовательская форма и другие.

**Семейство** (объект *Collection*) представляет собой объект, содержащий несколько других объектов, обычно одного и того же типа.

***Например***, семейство *Workbooks* содержит все открытые объекты *Workbook* (рабочая книга).

Каждый элемент семейства нумеруется и идентифицируется либо по номеру, либо по имени. Объектная библиотека VBA имеет иерархическую структуру. Полная ссылка на объект представляет собой полный путь к объекту с учетом всех уровней вложенности этой структуры. Но обычно ограничиваются неявной ссылкой на объект, т.е. в ней опускаются все объекты, которые в данный момент активны.

**Класс** – это важнейшее понятие объектно-ориентированного программирования. Класс определяет имя объекта, его свойства и действия, выполняемые над объектом.

Каждый объект, в свою очередь, является экземпляром класса.

**Метод** представляет собой действие, выполняемое над объектом. Метод может применяться ко всем объектам семейства.

**Синтаксис применения метода:** *Объект.Метод.*

***Например***,закрыть приложение: **Application.Quit***.*

**Свойство** – это атрибут объекта, определяющий его характеристики (размер, цвет, положение на экране, доступность или видимость объекта и др.). Для того, чтобы изменить характеристики объекта, достаточно изменить значения его свойств. Свойства можно изменять одновременно у всех объектов семейства.

**Синтаксис установки значения свойства:** *Объект.Свойство=ЗначениеСвойства***.**

***Например***, задать цвет текста в ячейке А2:

**Range("А2").Font.Colorindex=43**

**Событие** представляет собой действие, распознаваемое объектом (например, щелчок мышью), для которого можно запрограммировать отклик. Программирование на VBA заключается в том, что если пользователь производит некоторое воздействие на систему (например, щелчок левой кнопки мыши), то в качестве отклика выполняется код созданной пользователем процедуры.

**Редактор VBA**

Вызов редактора **VBA** осуществляется в MS Excel

**2010**: команда **Visual Basic** из группы **Код** вкладки **Разработчик;**

**2003**: при помощи команды **Сервис –****Макрос – редактор Visual Basic**.

**Среда программирования** **VBA** включает в себя окна: проекта, свойств, формы, отладки, локальных переменных, контрольного значения, программы, просмотра объектов и панели инструментов (рис. Г.1).

**Окно проекта** активизируется командой **Вид – Окно проекта** (*View* - *Project Explorer*) или нажатием кнопки **Окно проекта** (*Project Explorer*). В окне проекта отображается иерархическая структура файлов форм и модулей текущего проекта.

В проекте автоматически создается **модуль** для каждого рабочего листа, всей книги, а также для каждой пользовательской формы, макроса и класса. Модули делятся на два типа: ***стандартные*** (содержат макросы, добавляются в проект командой **Вставка – Модуль** (*Insert – Module*)) и ***модули объектов*** (модули, связанные с рабочей книгой, рабочими листами, формами и модули классов).

**Формы** создаются командой **Вставка – Форма** (*Insert – UserForm*), а **модули класса** – командой **Вставка – Модуль класса** *(Insert – Module*).

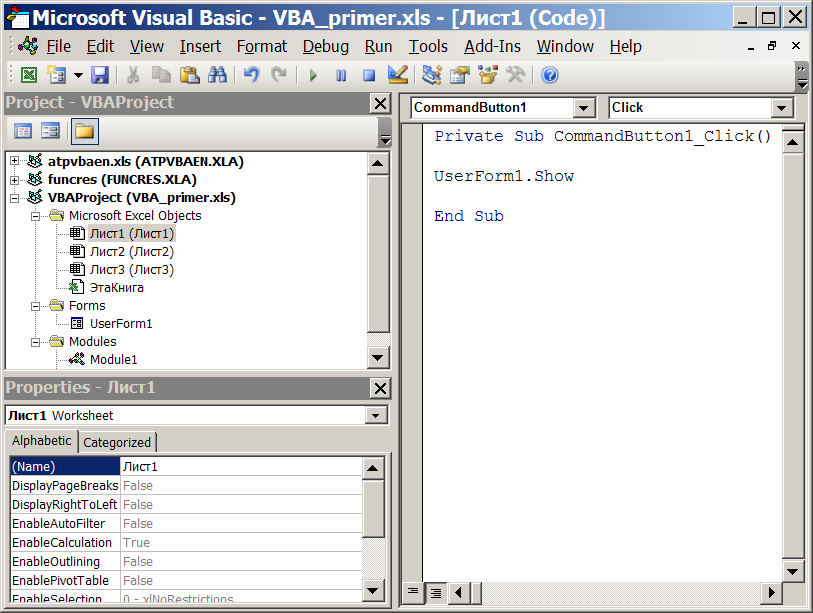


Рис.Г.1 Окно редактора кода VBA

Чтобы открыть окно для редактирования программного кода, следует дважды щелкнуть кнопкой мыши по значку файла в окне проекта. Это окно позволяет редактировать либо код отдельной процедуры, либо код всего модуля. Для того, чтобы выбрать один из этих режимов редактирования, следует воспользоваться одной из кнопок в левом нижнем углу окна редактирования **Отдельная процедура** *(Procedure View*), либо **Все процедуры модуля** *(Full Module View).*

**Редактор кода** позволяет существенно облегчить написание программ за счет способности автоматически завершать написание операторов, свойств и параметров. При написании кода редактор предлагает пользователю список компонентов, логически завершающих вводимую пользователем инструкцию. Ввод выбранного компонента в код программы завершается нажатием клавиши **<Tab>** либо двойным щелчком по выбранному компоненту. Автоматическое отображение списка компонентов осуществляется при установленном флажке **Список компонентов** *(Auto List Members)* вкладки **Редактор** *(Editor)* диалогового окна команды **Сервис - Параметры** *(Tools - Options).*

Кроме этого, редактор кода позволяет автоматически отображать на экране сведения о процедурах, функциях, свойствах и методах после выбора их имени. Для этого должен быть установлен флажок **Краткие сведения** *(Auto Quick* *Info)* на вкладке **Редактор** *(Editor*) диалогового окна команды **Сервис - Параметры** *(Tools - Options).*

Редактор кода также производит автоматическую проверку синтаксиса набранной строки кода сразу после нажатия клавиши **Enter**, если установлен флажок **Проверка синтаксиса** *(Auto Syntax Check)* вкладки **Редактор** *(Editor)* диалогового окна команды **Сервис - Параметры** *(Tools - Options).*

**Правила задания имен объектов VBA**

В **VBA** существуют определенные правила задания имен переменных, функций, процедур, типов, констант и других объектов:

1. Длина имени не должна превышать 255 символов.

2. Имя не должно содержать точек, пробелов и следующих символов: %, &. !, #, @, $.

3. Имя может состоять из любой комбинации букв, цифр и символов, начинающихся с буквы.

4. Имена должны быть уникальными внутри той области, в которой они определены.

5. Имена не должны совпадать с ключевыми словами VBA и именами встроенных функций и процедур.

6. Регистр в именах не имеет значения.

**Типы данных VBA**

Одним из самых фундаментальных понятий любого языка программирования является **тип данных**. Тип данных определяет множество допустимых значений, которое может принимать указанная переменная. В VBA поддерживаются следующие типы данных:

* *Byte* (байт);
* *Boolean* (логический);
* *Integer* (целое число);
* *Long* (длинное целое);
* *Single* (с плавающей запятой обычной точности);
* *Double* (число с плавающей запятой двойной точности);
* *Currency* (денежный);
* *Decimal* (масштабируемое целое число);
* *Date* (значения даты и времени);
* *Object* (объект);
* *String* (строка переменной или постоянной длины);
* *Variant* (числовые или строковые подтипы) и тип данных, *определяемый пользователем*.

**Переменные VBA**

Описание каждой переменной делает программу надежнее и убыстряет ее работу. Инструкция **Dim** предназначена для описания типа данных переменной на уровне модуля или процедуры, а также для описания объектного типа переменных. Для каждой описываемой переменной следует использовать отдельное предложение **As *Тип***. Если тип данных переменной или объекта не задан явно, по умолчанию переменной присваивается тип **Variant***.*

***Пример* *1.*** Описание переменных стандартного типа.

**Dim N As Integer ‘**целочисленная переменная

**Dim S As String ‘**строковая переменная

В VBA можно использовать **массивы**. Массив в программе обрабатывается поэлементно с использованием операторов цикла.

***Пример 2.*** Описание массивов различной размерности (одномерного массива (вектора), состоящего из действительных чисел и двумерного массива 5х5 (матрицы), состоящего из целых чисел).

**Dim A (15) As Integer ‘**одномерный массив (вектор),

**‘**состоящий из 15 целых чисел

**Dim B (5, 5) As Single ‘**двумерный массив (матрица) 5х5,

**‘**состоящий из действительных чисел

***Примеp 3.*** Использование оператора цикла для обработки элементов одномерного массива.

**For i=1 to n**

**A[ i ] = A[ i ]^2**

**Next i**

**Использование констант VBA**

Использование констант в программах делает их более читаемыми и позволяет легко вносить исправления. Константы, в отличие от переменных, не изменяют своих значений.

Синтаксис объявления константы:

**[Public | Private] Const ИмяКонстанты [As Тип] = Выражение,** где

* **Public –** ключевое слово, используется на уровне модуля для описания констант, доступных всем процедурам во всех модулях, не допускается в процедурах;
* **Private –**ключевое слово, используемое на уровне модуля для описания констант, доступных только внутри модуля, в котором выполняется описание, не допускается в процедурах;
* **ИмяКонстанты –** имя константы, соответствующее стандартным правила именования констант;
* **Тип –** один из поддерживаемых типов данных;
* **Выражение –** литерал, константа, либо любое сочетание, которое включает все арифметические или логические операторы, кроме **Is**.

**Организация ввода/вывода информации средствами VBA**

Ввод/вывод информации средствами VBA можно организовать двумя способами:

1. непосредственный ввод в ячейки рабочего листа с клавиатуры;
2. ввод с использованием диалоговых окон.

***Примеp 4.*** Присвоить переменной *X* значение свойства *Value* ячейки *C5.*

*Способ 1.* **X=Cells(5,3).Value**

*Способ 2.* **X=Cells(5,3)**

*Способ 3.* **X=Range(“C5”)**

***Пример 5.*** Присвоить переменной*X* значение ячейки, номер строки и столбца которой определяется значениями выражения *1+i* и *1+j* соответственно.

**X=Cells(1+i,1+j)**

***Пример 6.*** В ячейке *С2* вывести значение выражения *2x+2y.*

**Cells(2,3)=2\*x+2\*y**

**Второй способ** организации ввода/вывода предполагает использование окна ввода **InputBox** (для ввода данных с клавиатуры) и окна сообщений **MsgBox** (для вывода информации на экран).

*Здесь и в последующем необязательные элементы заключены в квадратные скобки. При наборе текстов программ квадратные скобки задавать не следует.*

**Синтаксис диалоговых окон:**

**MsgBox** ( prompt [,buttons] [,title] [helpfile, context] )

**InputBox** ( prompt [,title] [,default] [,xpoz] [,ypoz] [helpfile, context] )

**Аргументы:**

**prompt** — строковое выражение, которое выводится в диалоговом окне. Если оно состоит из нескольких строк, в качестве разделителя их используют символы возврата каретки (**Chr(13)**), перевода строки (**Chr(10)**) или комбинацию этих символов (**Chr(13)&Chr(13)**);

**title** — строковое выражение, отображаемое в строке заголовка окна. По умолчанию в строку заголовка помещается имя приложения;

**helpfile** — строковое выражение, задающее имя файла справки, содержащего справочную информацию о данном диалоговом окне. В случае указания этого аргумента необходимо задать **context**, определяющий номер соответствующего раздела справочной системы;

**default** — строковое выражение, размещаемое в поле ввода как значение по умолчанию;

**xpos** — расстояние по горизонтали от левого края экрана до левой границы окна. По умолчанию диалоговое окно центрируется по горизонтали;

**ypos** — расстояние по вертикали от верхнего края экрана до верхней границы окна. По умолчанию диалоговое окно позиционируется приблизительно на одну треть высоты экрана;

**buttons** — целое число, представляющее собой значение, которое определяет число и тип отображаемых в диалоговом окне кнопок, вид используемого значка, основную кнопку, модальность окна сообщений. Символические константы VBA и их числовые значения, применяемые в качестве параметра **buttons**, приведены в табл. Г.1.

Таблица Г.1

Символические константы аргумента **buttons**

| **Константа** | **Значение** | **Описание** | |
| --- | --- | --- | --- |
| **vbOKOnly** | 0 | Только кнопка "OK" (по умолчанию) | |
| **vbOKCancel** | 1 | Кнопки "OK" и "Отмена" (Cancel) | |
| **vbAbortRetryIgnore** | 2 | Кнопки "Прервать" (Abort), "Повторить" (Retry) и "Пропустить" (Ignore) | |
| **vbYesNoCancel** | 3 | Кнопки "Да" (Yes), "Нет" (No) и "Отмена" (Cancel) | |
| **vbYesNo** | 4 | Кнопки "Да" и "Нет" | |
| **vbRetryCancel** | 5 | Кнопки "Повторить" и "Отмена" | |
| **vbCritical** | 16 | Значок "Критическое сообщение" |  |
| **vbQuestion** | 32 | Значок "Предупреждающий запрос" |  |
| **vbExclamation** | 48 | Значок "Предупреждающее сообщение" |  |
| **vbInformation** | 64 | Значок "Информационное сообщение" |  |
| **vbDefaultButton1** | 0 | Основная — первая кнопка (по умолчанию) | |
| **vbDefaultButton2** | 256 | Основная — 2-я кнопка | |
| **vbDefaultButton3** | 512 | Основная — 3-я кнопка | |
| **vbDefaultButton4** | 768 | Основная — 4-я кнопка | |
| **vbApplicationModal** | 0 | Модальное окно сообщения на уровне приложения (по умолчанию) | |
| **vbSystemModal** | 4096 | Модальное окно сообщения на уровне системы | |

Функция **MsgBox** возвращает одно из значений, приведенных в табл. Г.2.

Таблица Г.2

Значения функции **MsgBox**

|  |  |  |
| --- | --- | --- |
| **Константа** | **Значение** | **Нажатая кнопка** |
| **vbOK** | 1 | OK |
| **vbCancel** | 2 | Отмена (Cancel) |
| **vbAbort** | 3 | Прервать (Abort) |
| **vbRetry** | 4 | Повторить (Retry) |
| **vbIgnore** | 5 | Пропустить (Ignore) |
| **vbYes** | 6 | Да (Yes) |
| **vbNo** | 7 | Нет (No) |

***Пример 7.*** Создать программу для вычисления значения функции ***y(х)=3х2+2x-1***. Значение переменной ***х*** ввести с клавиатуры при помощи окна ввода **InputBox**. Результат расчета вывести при помощи окна сообщения **MsgBox** (вывести в окне значок «Информационное сообщение»). Для заголовка окон диалога использовать строку вида: «*Функция Y(X)= 3X^2 + 2X -1*».

***Решение.***

В окне редактора программного кода VBA необходимо записать код процедуры (рис. Г.2). Для выполнения программы использовать команду меню **Run** – **Run** **Sub/UserForm** или клавишу **F5.** В процессе выполнения ввести значение переменной ***x*** (рис. Г.3) и оценить полученный результат (рис. Г.4).

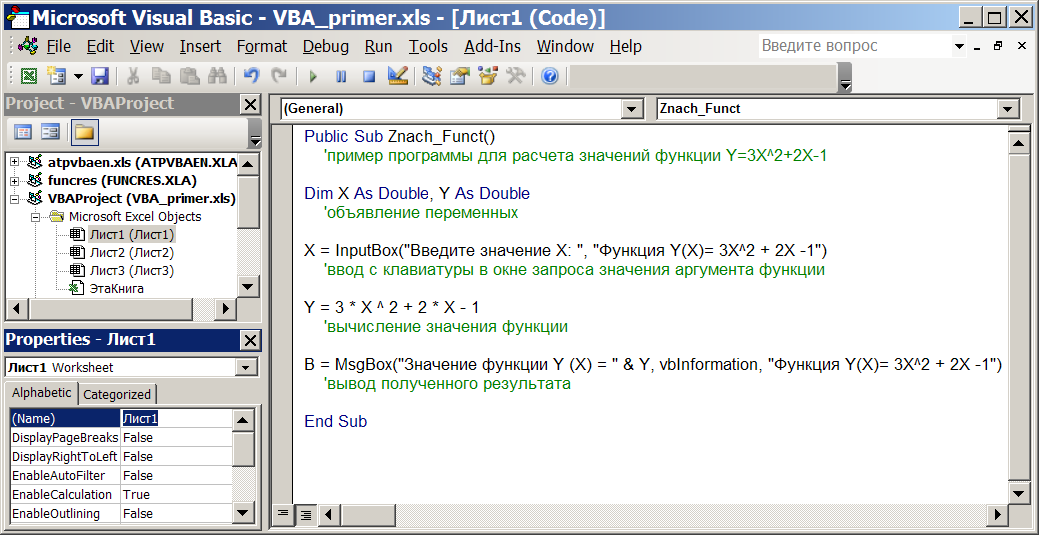


Рис. Г.2. Код программы на языке VBA

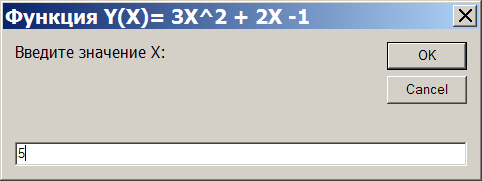


Рис Г.3. Окно диалога InputBox для ввода значения переменной *x*

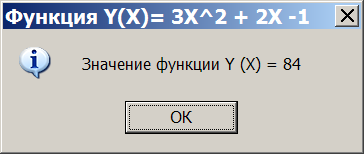


Рис. Г.4. Окно диалога MsgBox для вывода полученного результата

**Создание пользовательских функций в VBA**

Пользовательская функция представляет собой самостоятельную часть кода программы, которая имеет имя и содержит аргументы (иногда аргументы могут отсутствовать).

Создание пользовательских функций – это одна из наиболее простых задач, которые позволяет решать VBA. С пользовательскими функциями в MS Excel можно работать при помощи Мастера функций так же, как и со встроенными стандартными функциями. Мастер функций помещает имена созданных функций в категорию «**Определенные пользователем**». При вызове пользовательской функции список передаваемых функции аргументов должен по количеству и типу соответствовать списку, заданному в ее описании.

Для того, чтобы построить пользовательскую функцию, следует выполнить команду

**2010**: **Visual Basic** из группы **Код** на вкладке **Разработчик;**

**2003**: **редактор Visual Basic** в меню **Сервис/Макрос**.

В окне редактора **VBA** выполнить последовательно выполнить команды: **Вставка – Модуль** (Insert – Module), затем **Вставка – Процедура** (Insert – Procedure), выбрать переключатель Функция (Function) и задать имя функции. После этого в окне программного кода следует ввести текст функции. Доступные программные модули отображены в окне диспетчера проекта (*Project Explorer*).

При написании программного кода на языке VBA следует учитывать следующие **правила**:

- каждая инструкция должна располагаться на отдельной строке. Если возникает необходимость перенести инструкцию на следующую строку, в конце строки следует указать **сочетание символа пробела и знака «нижнее подчеркивание»**;

- возможно не более семи продолжений одной и той же строки, сама строка не должна содержать более 1024 символов;

- запрещается разделять переносом строковые константы, идентификаторы переменных и т.п. конструкции;

- чтобы разместить в одной строке несколько программных инструкций, их разделяют двоеточием.

**Комментарии** используются для добавления в текст программы пояснительного текста, который не является исполняемой частью программного кода, а служит для пояснения отдельных фрагментов этого кода. Комментарий начинается с символа **апострофа** (**‘**) или ключевого слова **Rem**. Компилятор игнорирует текст программы от начала комментария до конца строки. Таким образом, комментарии позволяют также временно предотвращать выполнение отдельных участков кода при отладке программы.

**Синтаксис функции пользователя**

**Publuc Function Имя (**[СписокАргументов]) **[As Тип]**

**[Инструкции]**

**[Имя = Выражение]**

**[Exit Function]**

**[Инструкции]**

**[Имя = Выражение]**

**End Function**

Ключевое слово **Public** указывает, что функция доступна на всех рабочих листах и во всех модулях проекта.

**Тип** — один из встроенных (Byte, Boolean, Integer, Long, Single, Double, Currency, Decimal, Date, String, Variant) или определенных пользователем типов.

Конструкция **Exit Function** приводят к принудительному немедленному завершению функции.

**Синтаксис элемента СписокАргументов**:

**[Optional] ByVal [ParamArray] ИмяПерем[()][As Тип][= поУмолч]**

Ключевое слово **Optional** указывает, что аргумент является необязательным и может быть опущен. Все последующие аргументы в списке аргументов должны быть также описаны с помощью ключевого слова **Optional** и иметь тип **Variant**. Если используется параметр **ParamArray**, необязательные аргументы недопустимы.

Инструкция **ByVal** определяют способ передачи аргумента в тело функции (по значению).

Конструкция **ParamArray** позволяет задавать произвольное количество аргументов. Используется для описания только последнего элемента в списке аргументов. Обработка произвольного количества элементов осуществляется с помощью инструкции **For…Each…Next**.

При работе с необязательными аргументами следует использовать функцию **IsMissing (***Аргумент***)**, которая возвращает значение **True**, если указанный параметр был опущен, и **False** — в противном случае. Для необязательного параметра можно задавать значение по умолчанию (**поУмолч**), которое присваивается переменной, если аргумент отсутствует.

**Основные операторы языка VBA**

**Оператором** называют синтаксическую единицу языка программирования, которая используется в программе для выполнения отдельного предписания. Операторы подразделяются на две категории – алгоритмические и функциональные.

**Алгоритмические операторы** используются для организации последовательности выполняемых пользователем действий. Важнейшие из них – операторы безусловного перехода, условные операторы и операторы циклов.

**Функциональные операторы** – это встроенные в язык функции и процедуры, с помощью которых производятся важные и распространенные действия, такие, как ввод данных, действия над числами.

Любая программа состоит из последовательности операторов, которые записываются в соответствии со строгими синтаксическими правилами.

Как отмечалось выше, функции пользователя размещаются в специально отведенной части VBA-проекта — модуле.

***Пример 8.***

Функция для вычисления площади треугольника по формуле Герона.

Аргументы функции: a, b, c – длины сторон треугольника.

**Public Function Geron (ByVal a As Double, ByVal b As Double, \_**

**ByVal c As Double) As Double**

**p = (a + b + c) / 2** ‘вычисление полупериметра

**Geron = sqr(p \* (p – a) \* (p – b) \* (p – c))** ‘вычисление площади и

‘присвоение результата ‘функции

**End Function**

***Пример 9.***

Функция для вычисления объема цилиндра.

Аргументы функции: R – радиус основания, H – высота.

**Public Function VCilindra(R, H)**

**If Not IsNumeric(R) Or Not IsNumeric(H) Then**

**VCilindra = "Введите числовые значения"**

**Else**

**If R <= 0 Or H <= 0 Then**

**VCilindra = "Введите числа больше 0"**

**Else**

**V = 4 \* Atn(1) \* R ^ 2 \* H**

**VCilindra = Format(V, "0.00")**

**End If**

**End If**

**End Function**

**Дополнительные источники по теме**

1. [**http://www.planetaexcel.ru/techniques/3/59/**](http://www.planetaexcel.ru/techniques/3/59/)
2. [**http://excelexpert.ru/avtomatizaciya-raboty-v-dokumentax-microsoft-excel**](http://excelexpert.ru/avtomatizaciya-raboty-v-dokumentax-microsoft-excel)
3. [**http://msdn.microsoft.com/ru-ru/library/office/ee814737(v=office.14).aspx**](http://msdn.microsoft.com/ru-ru/library/office/ee814737(v=office.14).aspx)
4. [**http://www.excelworld.ru/stuff/vba\_function/all\_vba\_function/all\_vba\_function/17-1-0-5**](http://www.excelworld.ru/stuff/vba_function/all_vba_function/all_vba_function/17-1-0-5)
5. [**http://excelvba.ru/code**](http://excelvba.ru/code)

# 2 Разработка пользовательской формы

## *2.1 Постановка задачи*

Разработать пользовательскую форму, реализующую линейные вычисления периметра квадрата по радиусу описанной окружности: .

Предусмотреть контроль вводимых исходных данных с выводом предупреждающих сообщений в случае некорректных данных или их отстутствия.

Выполнить отладку и проверку работоспособности пользовательской формы при различных исходных данных (включая некорректные данные).

## *2.2 Разработка макета формы*

Чтобы создать новую пользовательскую форму в VBA необходимо в окне редактора выполнить команду **Insert(Вcтавить)/UserForm(Форма)**.

Затем непосредственно в режиме Конструктора формы (рисунок Г.5) необходимо изменить размеры формы, в окне **Properties** (Свойства) задать заголовок формы (свойство Caption) и другие необходимые параметры.

Элементы управления (**ЭУ**) формы (таблица Г.3) добавляются в режиме Конструктора формы с панели инструментов Toolbox с помощью мыши.

Настройка свойств выбранного ЭУ выполняется в соответствующем окне **Properties** (Свойства) ЭУ.

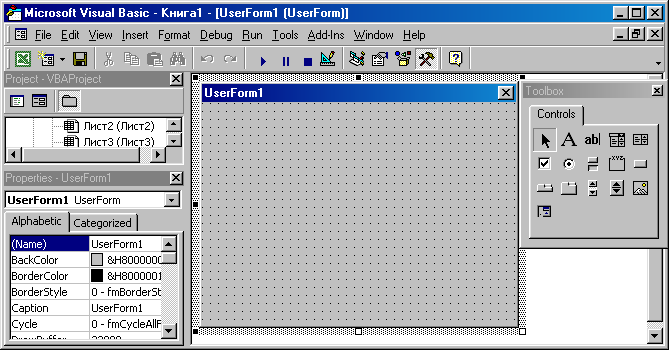


Рис. Г.5. Создание формы

Таблица Г.3

Характеристики элементов управления

|  |  |  |  |
| --- | --- | --- | --- |
| ***Элемент управления*** | ***Кнопка*** | ***Префикс  в имени*** | ***Назначение*** |
| ***Надпись*** | *Label* | ***lbl*** | Отображение произвольного текстового выражения (пояснения) |
| ***Поле*** | *TextBox* | ***txt*** | Ввод текстовой информации, которая может преобразовываться в числа и даты |
| ***Поле  со списком*** | *ComboBox* | ***cbo*** | Хранение списка значений, из которого пользователь может выбрать или ввести с клавиатуры только одно значение |
| ***Список*** | *ListBox* | ***lst*** | Хранение списка значений, из которого пользователь может выбрать одно или несколько значений |
| ***Флажок*** | *CheckBox* | ***chk*** | Выбор из нескольких возможных вариантов |
| ***Переключатель*** | *OptionButton* | ***opt*** | Выбор одного из нескольких взаимоисключающих вариантов |
| ***Frame*** | *Рамка* |  | Визуальная группировка элементов управления. |
| ***Выключатель*** | *ToggleButton* | ***tgl*** | Выбор из двух альтернатив |
| ***Кнопка*** | *CommandButton* | ***cmd*** | Выполнение некоторых действий при нажатии |
| ***Набор вкладок*** | *TabStrip* |  | Создание нескольких вкладок в диалоговом окне |
| ***Набор страниц*** | *MultiPage* |  | Создание многостраничных окон |
| ***Полоса прокрутки*** | *ScrollBar* | ***scr*** | Скроллинг (прокручивание) информации, не умещающейся в границы окна |
| ***Счетчик*** | *SpinButton* | ***spn*** | Изменение числовых величин (+/-) на заданный шаг |
| ***Рисунок*** | *Image* | ***img*** | Отображение в форме графических файлов форматов: BMP, GIF, .JPG, ICO, WMF. |
| ***Редактор ссылок*** | *RefEdit* |  | Ввод ссылок на ячейки и диапазоны |

**Исходные данные** задаются посредством элементов управления: ***Поле, Поле со списком, Список, Счетчик, Полоса прокрутки.***

**Пояснительные надписи** на форме осуществляются посредством элемента управления ***Надпись****.*

**Для выбора действий** используются элементы управления: ***Флажок, Переключатель, Выключатель****.*

**Выполнение и завершение макросов** осуществляется посредством элемента управления ***Кнопка****.*

**Для вставки рисунков** используется элемент управления ***Рисунок****.*

**Для работы с вкладками и страницами** используются элементы управления ***Набор Вкладок*** и ***Набор Страниц****.*

**Выходные данные** отображаются посредством элементов управления ***Поле, Поле со списком, Список.***

Пример макета формы приведен на рисунке Г.6.

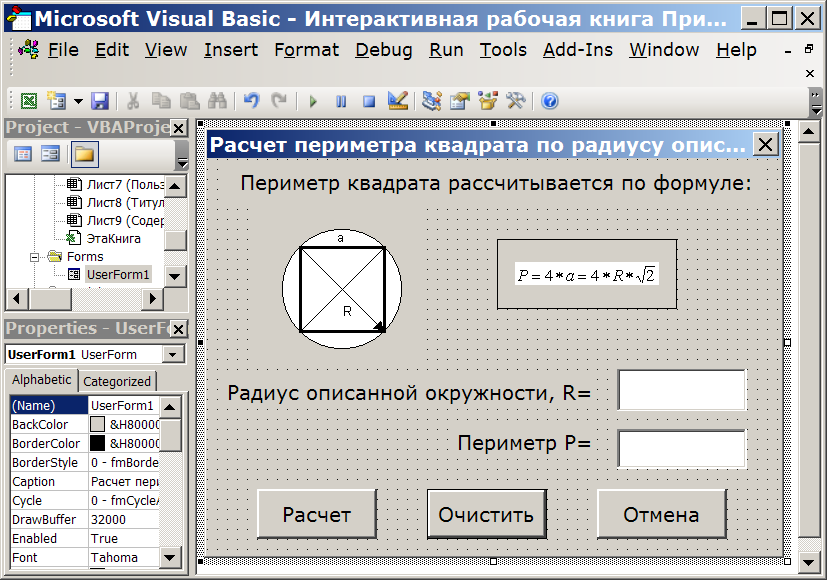


Рис. Г.6. Пример макета формы пользователя

Пояснительная информация в форме размещена с помощью элементов управления Надпись – Label1, Label2, Label3.

Иллюстрации к решению задачи добавлены при помощи элемента управления Рисунок – Image1, Image2.

Для ввода исходных данных предусмотрен элемент управления Поле – TextBox1 (радиус R).

Для вывода результата вычислений предусмотрено Поле TextBox2 (периметр Р).

Вычисление должно выполняться после нажатия кнопки Расчет.

При нажатии кнопки Очистить должны удаляться значения в полях TextBox1 и TextBox2 (без закрытия формы).

При нажатии кнопки Отмена должно происходить закрытие формы.

## *2.3 Разработка программного кода*

После разработки макета формы необходимо выделить командную кнопку **Расчет** (CommandButton1) и выполнить команду **View (Вид)/Code (Код).** При этом откроется окно кода с командами начала и завершения процедуры CommandButton1\_Click() – нажатия кнопки **Расчет** (рисунок Г.7).

Между указанными командами записываются необходимые вычисления.

Аналогично создается программный код для кнопок **Отмена** и **Очистить** (CommandButton2 и CommandButton3, рисунок Г.8).

В тексте программного кода необходимо предусмотреть проверку вводимых исходных данных и обработку ошибок при попытке нажатия кнопки Расчет без исходных данных.

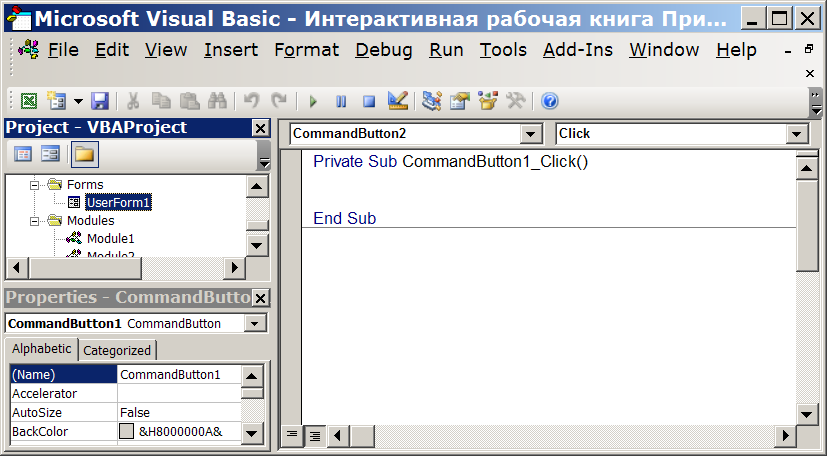


Рис. Г.7. Шаблон программного кода процедуры для нажатия кнопки

Окончательный вид макета формы и код процедуры, позволяющей по радиусу описанной окружности вычислить периметр квадрата по формуле: , приведены на рисунке Г.9.

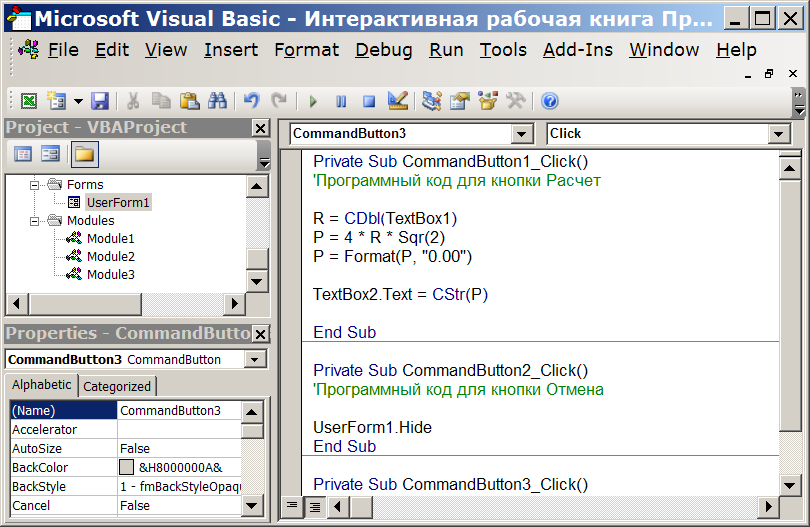


Рис. Г.8. Программный код процедур нажатия кнопок Расчет и Отмена

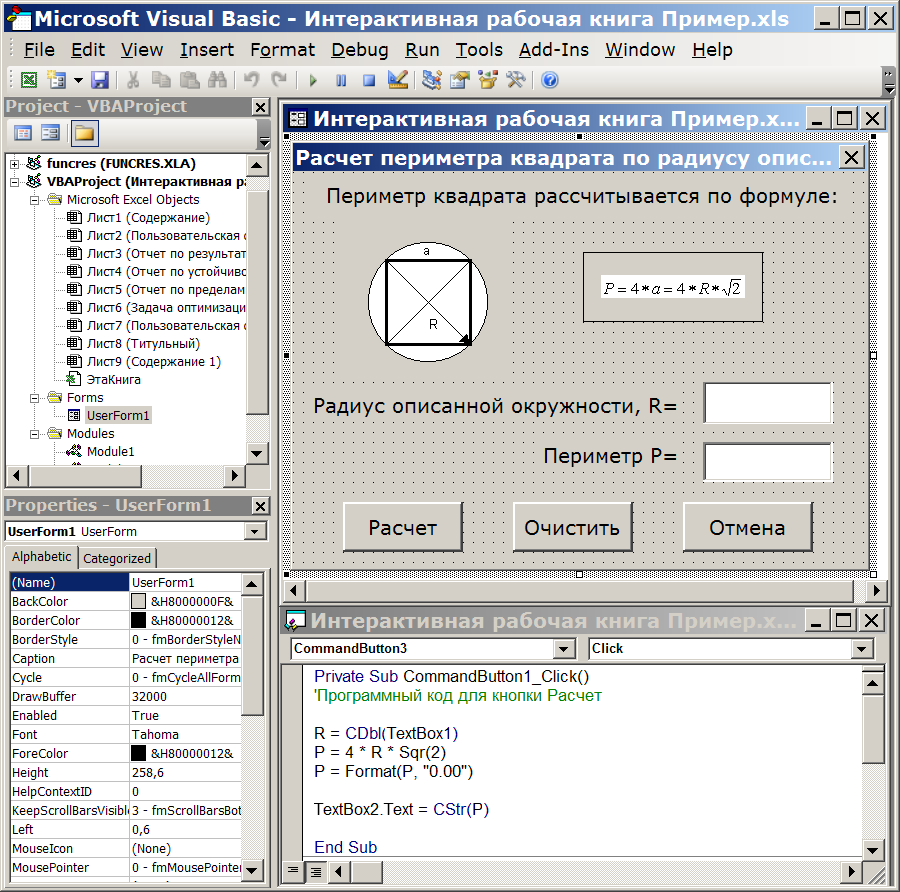


Рис. Г.9. Макет и программный код пользовательской формы

Для ввода исходных данных (радиус R) использован элемент поле TextBox1, значение из которого преобразуется к типу Double.

Вычисления выполняются после нажатия кнопки **Расчет** (CommandButton1).

Вывод результата осуществляется в поле TextBox2 (значение периметра приводится к числовому формату с двумя знаками после запятой и преобразуется к типу String).

Нажатие кнопки **Отмена** (CommandButton2) приведет к очистке от текущих значений текстовых полей TextBox1, TextBox2 и закрытию формы.

Нажатие кнопки **Очистить** (CommandButton3) приведет к очистке от текущих значений текстовых полей TextBox1 и TextBox2 (без закрытия формы).

## *2.4 Проверка работоспособности пользовательской формы*

Отладка программного кода формы осуществляется командой **Debug (Отладка)/ Compile VBAProject (компилировать).**

Проверка работоспособности разработанной формы производится путем ее вызова при нажатии кнопки на рабочем листе Excel.

Для создания кнопки вызова формы используется панель инструментов Формы.

Макрос для кнопки вызова формы имеет вид (рисунок Г.10).

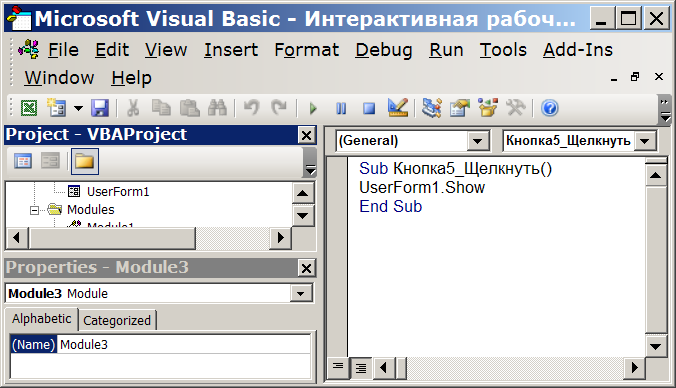


Рис. Г.10. Процедура для кнопки вызова формы

Пример вызова на листе Excel разработанной формы, включая:

* + окно рабочей книги с постановкой задачи и кнопкой вызова формы (рисунок Г.11);
  + окно формы с исходными данными и полученным результатом вычисления. Например, на рисунках Г.12 и Г.13 приведена форма до и после нажатия кнопки Расчет;
  + диалоговое окно, которое выводится в случае попытки расчета при отсутствии исходных данных или отрицательных значениях (рисунок Г.14).

Программные коды процедур для кнопок необходимо привести в виде обычного текста на отдельном рабочем листе Excel (Фамилия\_Код).

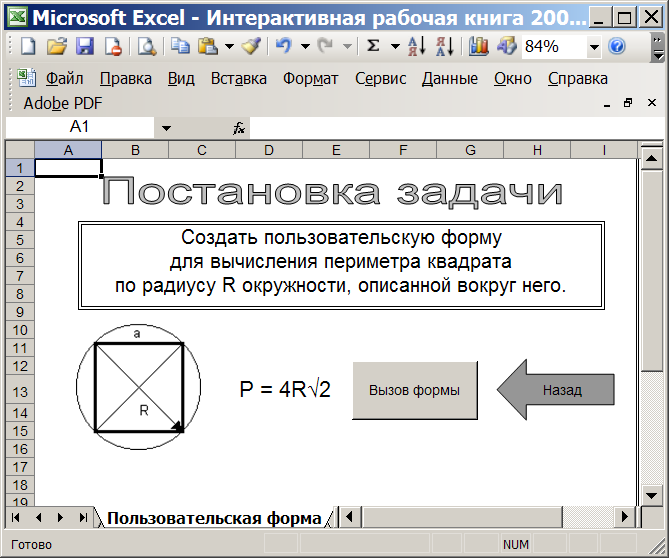


Рис. Г.11. Кнопка вызова формы

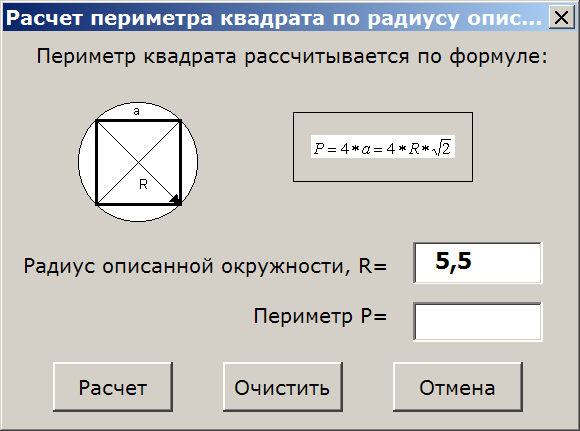


Рис. Г.12. Ввод исходных данных

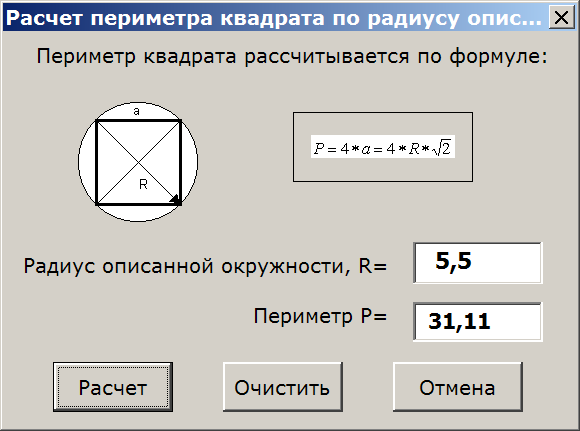


Рис. Г.13. Вывод результата после нажатия кнопки Расчет

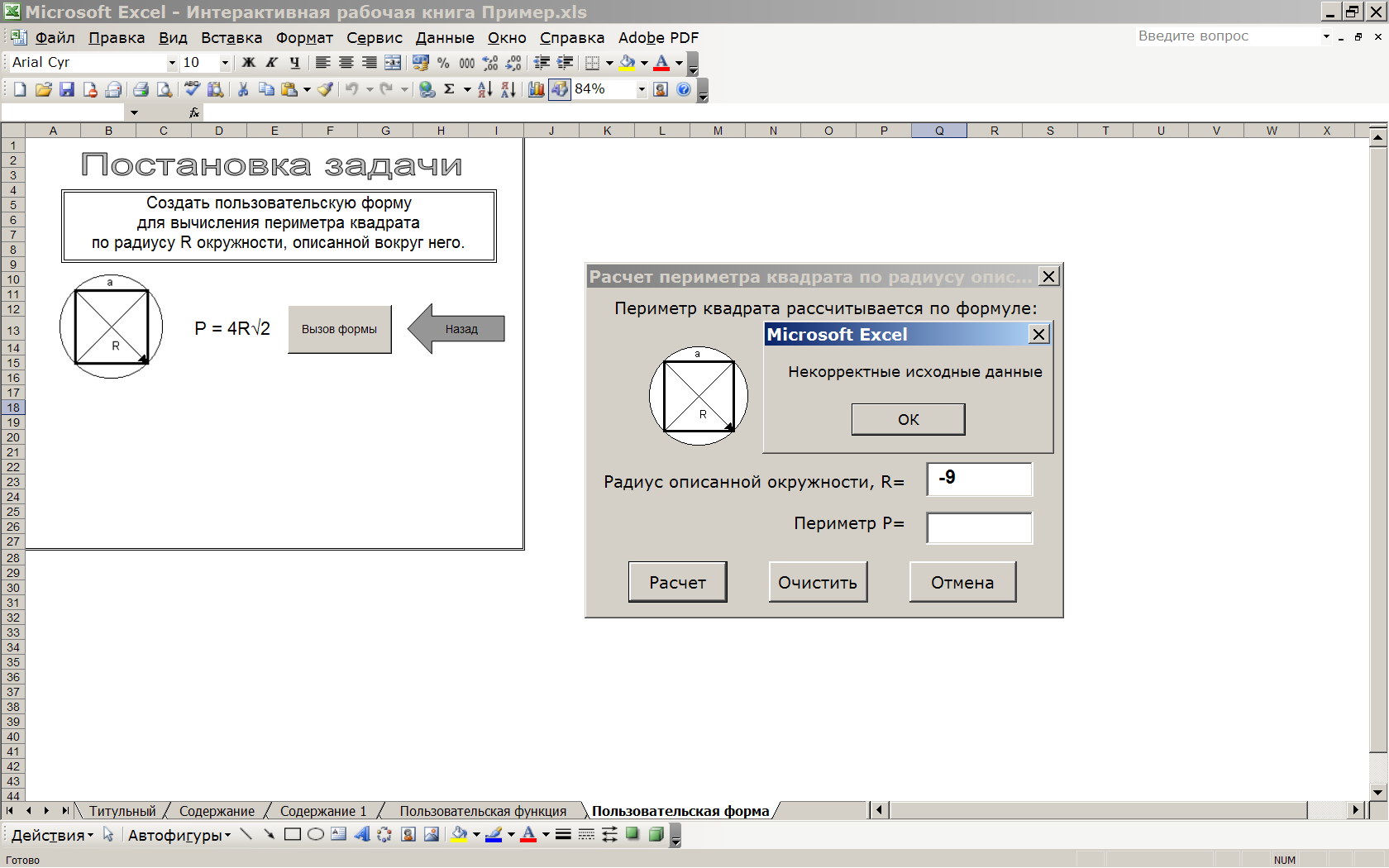
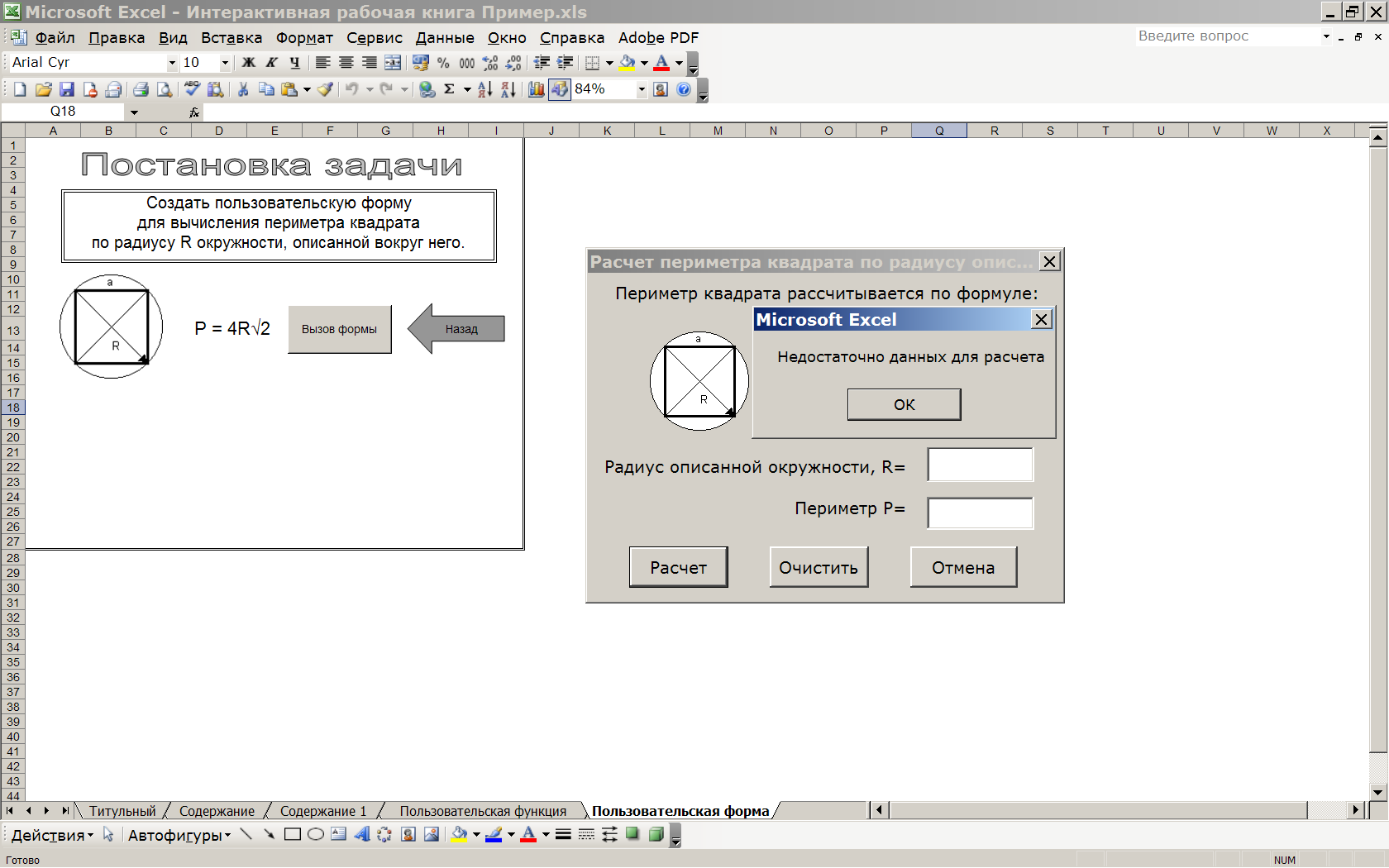


Рис. Г.14. Вывод сообщений при нажатии кнопки Расчет   
при отсутствии или некорректных исходных данных в поле TextBox1