

**МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)
ВОЛЖСКИЙ ФИЛИАЛ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ
по дисциплине
«ОСНОВЫ ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ»**

Кафедра: ИиТТП

Чебоксары

Оглавление

Лабораторная работа 1. Структурный подход к программированию. Написание первой программы, макрос.	3
Лабораторная работа 2. Цифровые активы в экосистеме беспилотного движения.....	5
Лабораторная работа 3. Выбор средств организации циклов	8
Лабораторная работа 4. Вставка пользовательских форм.....	11
Лабораторная работа 5. Данные для дашборда, анализ числовых данных показателей для построения дашбордов.....	14
КРИТЕРИИ ОЦЕНКИ.....	18
Список использованных источников.....	19

Лабораторная работа 1. Структурный подход к программированию. Написание первой программы, макрос.

1. Цель работы

1. Изучить основные принципы структурного программирования (линейная, ветвящаяся и циклическая структуры).
2. Освоить среду разработки макросов (на примере редактора VBA в MS Excel).
3. Приобрести навыки записи, редактирования и отладки макросов.
4. Написать первую программную процедуру, реализующую вычислительный алгоритм.

2. Краткие теоретические сведения

Структурный подход к программированию

Структурное программирование — это методология разработки программного обеспечения, в основе которой лежит разбиение программы на модули (подпрограммы, процедуры, функции), имеющие одну точку входа и одну точку выхода.

Основные базовые конструкции структурного программирования:

1. **Следование (Sequence):** Команды выполняются последовательно, одна за другой.
2. **Ветвление (Selection):** Выполнение той или иной группы команд зависит от условия (операторы If...Then...Else, Select Case).
3. **Цикл (Iteration):** Многократное выполнение группы команд до выполнения определенного условия (операторы For...Next, Do...Loop, While...Wend).

Преимущества подхода: читаемость кода, легкость отладки, возможность повторного использования кода.

Понятие макроса

Макрос — это последовательность команд, записанная на языке программирования (в данном случае Visual Basic for Applications), которая автоматизирует рутинные действия пользователя.

Макросы могут быть созданы двумя способами:

1. **Запись макроса:** Система автоматически записывает действия пользователя (клики, ввод данных) и переводит их в код.
2. **Написание кода вручную:** Программист пишет код в редакторе, используя возможности языка (переменные, циклы, условия).

Структура программы на VBA

Программа в VBA называется **Процедурой** (Sub) или **Функцией** (Function). Пример структуры процедуры:

```
Sub ИмяПроцедуры()  
    ' Объявление переменных  
    Dim a As Integer  
    Dim b As String  
  
    ' Тело программы  
    a = 10  
    b = "Результат"  
  
    ' Вывод результата  
    MsgBox b & ": " & a  
End Sub
```

Задание на лабораторную работу

Студенту необходимо выполнить три этапа работы:

Этап 1. Запись простого макроса. Записать макрос, который выполняет форматирование выделенной ячейки (например: жирный шрифт, желтый фон, граница). Проанализировать полученный код.

Этап 2. Написание первой программы (ручной код). Разработать макрос, который запрашивает у пользователя два числа, вычисляет их сумму и выводит результат в активную ячейку. Использовать структуру «Следование».

Этап 3. Реализация ветвления. Модифицировать программу из Этапа 2. Если сумма чисел больше 100, выводить сообщение «Большое число», иначе — «Малое число». Использовать конструкцию If.

4. Порядок выполнения работы

Подготовка рабочего места

1. Запустите MS Excel.
2. Включите вкладку «Разработчик» (Файл -> Параметры -> Настроить ленту -> Разработчик).
3. Установите уровень безопасности макросов: «Включить все макросы» (только на время лабораторной работы!) или «Уведомлять».

Выполнение Этапа 1 (Запись макроса)

1. На вкладке «Разработчик» нажмите «Запись макроса».
2. Присвойте имя макросу (латиницей, без пробелов), например FormatCell.
3. Выполните действия: выделите ячейку, сделайте шрифт жирным, залейте цветом.
4. Нажмите «Остановить запись».
5. Откройте редактор VBA (клавиши Alt + F11).
6. Найдите модуль с кодом макроса. Скопируйте код в отчет.
7. *Задание:* Удалите лишние строки (например, выбор конкретных ячеек Select), оставив только свойства форматирования. Это пример оптимизации кода.

Выполнение Этапа 2 (Первая программа)

1. В редакторе VBA создайте новый модуль (Вставка -> Модуль).
2. Напишите процедуру:

```
Sub Summa()  
    Dim x As Double  
    Dim y As Double  
    Dim res As Double  
  
    ' Ввод данных через диалоговые окна  
    x = InputBox("Введите первое число")  
    y = InputBox("Введите второе число")  
  
    ' Вычисление  
    res = x + y  
  
    ' Вывод в ячейку  
    ActiveCell.Value = res  
End Sub
```

3. Запустите макрос (клавиша F5 или через Excel). Проверьте работу.

Выполнение Этапа 3 (Ветвление)

1. Добавьте конструкцию условия в код:

```
If res > 100 Then  
    MsgBox "Большое число"  
Else  
    MsgBox "Малое число"  
End If
```

2. Протестируйте программу с разными входными данными (например, 10 и 20; 60 и 50).

3. Используйте точку останова (Breakpoint) для пошаговой отладки (клавиша F8), чтобы увидеть, как меняется поток выполнения программы.

5. Требования к отчету

Отчет по лабораторной работе должен содержать:

1. Титульный лист (название ВУЗа, кафедры, тема, ФИО студента и группы).
2. Цель работы.
3. Листинг кода макроса из Этапа 1 (с комментариями, что делает каждая строка).
4. Листинг кода программы из Этапа 2 и 3.
5. Скриншоты работы программы (диалоговые окна ввода, результат в ячейке, сообщение MsgBox).
6. Выводы: что было изучено, какие трудности возникли, в чем преимущество ручного написания кода перед записью макроса.

Лабораторная работа 2. Цифровые активы в экосистеме беспилотного движения

1. Цель работы

1. Изучить понятие переменной, правила именования и области видимости.
2. Освоить основные типы данных (числовые и строковые).
3. Научиться объявлять переменные с помощью оператора Dim.
4. Изучить методы и функции для обработки строковых данных (конкатенация, извлечение подстрок, изменение регистра, поиск).
5. Получить навыки явного преобразования типов данных.

2. Краткие теоретические сведения

Переменные и типы данных

Переменная — это именованная область памяти, предназначенная для хранения данных, которые могут изменяться в процессе выполнения программы.

В языке VBA перед использованием переменную рекомендуется объявлять. Это делается с помощью оператора Dim (Dimension). *Синтаксис:* Dim ИмяПеременной As ТипДанных

Основные типы данных:

Тип данных	Описание	Пример
Integer	Целое число (-32768 до 32767)	10, -5
Long	Длинное целое число	100000
Double	Число с плавающей точкой	3.14, 2.5
String	Строка (текст)	"Привет", "A1"
Boolean	Логический тип	True, False
Variant	Универсальный тип (может хранить что угодно)	Любое значение

Правила именования переменных:

1. Имя должно начинаться с буквы.
2. Не может содержать пробелы и специальные символы (точка, восклицательный знак и т.д.).
3. Не может совпадать с ключевыми словами языка (например, нельзя назвать переменную Loop или Sub).

4. Рекомендуется использовать «Венгерскую нотацию» или понятные имена (например, strName, iCount).

Управление строками

Строковая переменная (String) хранит последовательность символов.

Основные операции со строками:

1. **Конкатенация (сцепление):** Объединение двух строк в одну. В VBA используется оператор &.
 - Пример: "Привет, " & "Мир" → "Привет, Мир".
2. **Определение длины:** Функция Len(Строка). Возвращает количество символов.
3. **Извлечение подстроки:**
 - Left(Строка, Количество) — символы слева.
 - Right(Строка, Количество) — символы справа.
 - Mid(Строка, Начало, Количество) — символы из середины.
 - *Важно:* Нумерация символов в VBA начинается с 1.
4. **Изменение регистра:**
 - UCase(Строка) — перевод в верхний регистр (ВСЕ ЗАГЛАВНЫЕ).
 - LCase(Строка) — перевод в нижний регистр (все строчные).
5. **Поиск:** Функция InStr(Строка, Подстрока). Возвращает позицию первого вхождения.
6. **Удаление пробелов:** Функция Trim(Строка) удаляет пробелы в начале и в конце строки.

Преобразование типов

Часто данные, полученные от пользователя (через InputBox), приходят в виде строки. Для вычислений их нужно преобразовать.

- CInt(Выражение) — преобразование в целое число.
- CDbl(Выражение) — преобразование в дробное число.
- CStr(Выражение) — преобразование в строку.
- Val(Строка) — извлечение чисел из строки.

3. Задание на лабораторную работу

Студенту необходимо разработать программу (макрос), состоящую из последовательных шагов:

Задание 1. Объявление и инициализация. Создать процедуру, в которой объявляются три переменные: strFirstName (String), strLastName (String), intAge (Integer). Присвоить им значения и вывести каждое в отдельном сообщении (MsgBox).

Задание 2. Работа со строками (Анкета).

1. Запросить у пользователя Имя и Фамилию через InputBox.
2. Сформировать полное ФИО (Фамилия И.О. или просто Фамилия Имя) с помощью конкатенации.
3. Вывести результат в активную ячейку Excel.
4. Вычислить длину введенной фамилии и вывести её в соседнюю ячейку.

Задание 3. Манипуляции и преобразование.

1. Запросить у пользователя строку, содержащую число (например, "150").
2. Преобразовать эту строку в числовой тип.
3. Увеличить число на 10%.
4. Результат преобразовать обратно в строку и вывести в сообщении с текстом: "Новое значение: [результат]".
5. *Дополнительно:* Перевести введенное пользователем имя (из Задания 2) в верхний регистр перед выводом.

4. Порядок выполнения работы

Настройка среды

1. Откройте MS Excel.
2. Перейдите в редактор VBA (Alt + F11).

3. Вставьте новый модуль (Insert -> Module).
4. **Важно:** В самом начале модуля напишите Option Explicit. Это заставит программу выдавать ошибку, если вы забудете объявить переменную (хорошая практика).

Реализация Задания 1

Напишите код:

```
Option Explicit

Sub Task1_Variables()
    Dim strFirstName As String
    Dim strLastName As String
    Dim intAge As Integer

    strFirstName = "Иван"
    strLastName = "Иванов"
    intAge = 20

    MsgBox "Имя: " & strFirstName
    MsgBox "Фамилия: " & strLastName
    MsgBox "Возраст: " & intAge
End Sub
```

Запустите код (F5), убедитесь в работе.

Реализация Задания 2

Допишите новую процедуру:

```
Sub Task2_StringOps()
    Dim fName As String
    Dim lName As String
    Dim fullName As String
    Dim nameLen As Integer

    ' Ввод данных
    fName = InputBox("Введите имя")
    lName = InputBox("Введите фамилию")

    ' Обработка
    ' Trim убирает случайные пробелы, UCase делает заглавными
    fullName = UCase(Trim(lName)) & " " & Trim(fName)
    nameLen = Len(lName)

    ' Вывод в Excel
    ActiveCell.Value = fullName
    ActiveCell.Offset(0, 1).Value = nameLen ' Соседняя ячейка справа
End Sub
```

Протестируйте ввод с лишними пробелами и разным регистром.

Реализация Задания 3

Создайте процедуру для работы с числами в строках:

```
Sub Task3_Conversion()
    Dim strNum As String
    Dim dblNum As Double
    Dim dblResult As Double

    strNum = InputBox("Введите число (как текст)")

    ' Преобразование строки в число
    dblNum = CDbl(strNum)

    ' Вычисление
    dblResult = dblNum * 1.1

    ' Вывод (автоматическое преобразование числа в строку при сцеплении)
    MsgBox "Новое значение: " & CStr(dblResult)
End Sub
```

Проверка: Попробуйте ввести вместо числа текст (например, "абв"). Проанализируйте ошибку «Тип несовместим». Это демонстрирует важность контроля ввода.

5. Требования к отчету

Отчет должен содержать:

1. Титульный лист.
2. Цель работы.
3. Листинг кода всех трех процедур с комментариями к ключевым строкам (объявление, преобразование, функции).
4. Скриншоты:
 - Окна ввода данных (InputBox).
 - Результат в ячейках Excel.
 - Окна сообщений (MsgBox).
 - Окно ошибки (если удалось вызвать её вводом некорректных данных в Задании 3).
5. Выводы:
 - Зачем нужно объявлять тип переменной?
 - В чем разница между оператором + и & при работе со строками?
 - Какие функции обработки строк показались наиболее полезными?

6. Варианты индивидуальных заданий

(Преподаватель может назначить один из вариантов для усложнения)

- **Вариант А:** Написать программу, которая запрашивает полный адрес (Город, Улица, Дом) и формирует строку для почтового конверта в формате: «Куда: Город, Улица, д. Дом».
- **Вариант Б:** Написать программу, которая берет слово и выводит его задом наперед (используя цикл или функции строки).
- **Вариант В:** Написать программу, которая проверяет, содержит ли введенная строка символ «@» (имитация проверки e-mail), и выводит сообщение «Корректно» или «Ошибка».

Лабораторная работа 3. Выбор средств организации циклов

1. Цель работы

1. Изучить основные виды циклических конструкций в языке VBA (For...Next, Do...Loop, For Each...Next).
2. Научиться анализировать алгоритмическую задачу и выбирать наиболее подходящий тип цикла для её реализации.
3. Освоить методы управления ходом выполнения цикла (досрочный выход, продолжение итерации).
4. Получить навыки отладки циклических алгоритмов и предотвращения бесконечных циклов.

2. Краткие теоретические сведения

Цикл — это управляющая конструкция, позволяющая многократно выполнять одну и ту же последовательность команд (тело цикла). Выбор типа цикла зависит от логики задачи: известно ли количество повторений заранее или оно зависит от условия.

Цикл со счетчиком (For...Next)

Используется, когда **количество повторений известно заранее** или может быть вычислено до начала цикла. *Синтаксис:*


```

For Счетчик = НачальноеЗначение To КонечноеЗначение [step Шаг]
|   ' Тело цикла
Next [Счетчик]

```

Особенности: Шаг может быть отрицательным. Изменение переменной-счетчика внутри тела цикла не рекомендуется.

Цикл с условием (Do...Loop)

Используется, когда **количество повторений неизвестно** и зависит от выполнения некоторого условия в процессе работы. *Варианты:*

1. Проверка условия перед выполнением:

```

Do While Условие
|   ' Тело цикла
Loop

```

(Цикл может не выполниться ни разу, если условие ложно сразу)

2. Проверка условия после выполнения:

```

Do
|   ' Тело цикла
Loop Until Условие

```

(Тело цикла выполнится минимум один раз)

Цикл по коллекции (For Each...Next)

Используется для перебора элементов коллекции или массива (например, ячеек в диапазоне, листов в книге). *Синтаксис:*

```

For Each Элемент In Коллекция
|   ' Тело цикла
Next [Элемент]

```

Особенности: Не требует счетчика, безопаснее при работе с объектами Excel, часто работает быстрее цикла For при переборе диапазонов.

Управление циклом

- Exit For / Exit Do — немедленный выход из цикла.
- Continue For (в VBA нет прямого аналога, используется переход меткой или логика If) — пропуск остатка текущей итерации.
- **Бесконечный цикл:** Ситуация, когда условие выхода никогда не становится истинным. Требуется принудительной остановки (Ctrl + Break).

3. Задание на лабораторную работу

Студенту необходимо разработать три процедуры, демонстрирующие разные типы циклов, и обосновать выбор конструкции.

Задание 1. Цикл со счетчиком (Заполнение диапазона). Написать макрос, который заполняет диапазон ячеек (например, A1:A10) числами от 1 до 10. Каждую вторую ячейку закрасить цветом. *Требование:* Использовать цикл For...Next.

Задание 2. Цикл с условием (Валидация ввода). Написать макрос, который запрашивает у пользователя пароль через InputBox. Цикл должен повторяться до тех пор, пока пользователь не введет правильное значение (например, "1234") или не нажмет «Отмена». *Требование:* Использовать цикл Do...Loop. Предусмотреть счетчик попыток (максимум 3 попытки).

Задание 3. Цикл по коллекции (Обработка диапазона). Написать макрос, который проверяет заполненный числовой диапазон (например, B1:B20). Все отрицательные числа должны быть заменены на 0, а ячейки с ними закрашены красным. *Требование:* Использовать цикл For Each...Next.

Задание 4. Сравнительный анализ. В отчете ответить на вопрос: почему в Задании 3 нецелесообразно использовать цикл For...Next с индексом, если диапазон может изменяться?

4. Порядок выполнения работы

Подготовка

1. Откройте MS Excel, создайте новый файл.
2. Откройте редактор VBA (Alt + F11), вставьте новый модуль.
3. Добавьте строку Option Explicit для обязательного объявления переменных.

Выполнение Задания 1 (For...Next)

1. Объявите переменную-счетчик i.
2. Используйте цикл от 1 до 10.
3. Внутри цикла присваивайте значение ячейке Cells(i, 1).
4. Используйте оператор If и оператор Mod (остаток от деления) для проверки четности номера строки и изменения цвета (Interior.Color).

```
For i = 1 To 10
    Cells(i, 1).Value = i
    If i Mod 2 = 0 Then
        Cells(i, 1).Interior.Color = vbYellow
    End If
Next i
```

Выполнение Задания 2 (Do...Loop)

1. Объявите переменные для ввода (strPass), счетчика попыток (iTry).
2. Используйте конструкцию Do While или Do...Loop Until.
3. Внутри цикла проверяйте ввод. Если нажата «Отмена» (пустая строка), используйте Exit Do.
4. Если попытки исчерпаны, выведите сообщение «Доступ запрещен».

```
Do While iTry < 3
    strPass = InputBox("Введите пароль")
    If strPass = "" Then Exit Do ' Нажата Отмена
    If strPass = "1234" Then
        MsgBox "Доступ разрешен"
        Exit Do
    Else
        iTry = iTry + 1
        MsgBox "Неверно. Попытка " & iTry
    End If
Loop
```

Выполнение Задания 3 (For Each...Next)

1. Объявите переменную типа Range (например, cell).
2. Задайте диапазон для перебора (например, Range("B1:B20")).
3. В цикле проверяйте значение cell.Value.
4. Если меньше 0, меняйте значение и цвет.

```
For Each cell In Range("B1:B20")
    If cell.Value < 0 Then
        cell.Value = 0
        cell.Interior.Color = vbRed
    End If
Next cell
```

Отладка

1. Заполните диапазон B1:B20 произвольными числами (включая отрицательные) для тестирования Задания 3.
2. Запустите макросы по очереди.
3. Используйте пошаговое выполнение (F8), чтобы увидеть, как меняется переменная-счетчик и как программа входит/выходит из циклов.

5. Требования к отчету

Отчет должен содержать:

1. Титульный лист.

2. Цель работы.
3. Листинг кода всех трех процедур с комментариями.
4. Скриншоты результатов работы каждого макроса (заполненный столбец, окно ввода пароля, обработанный диапазон).
5. **Таблица сравнения циклов** (заполняется студентом):

Тип цикла	Когда используется	Преимущество	Недостаток
For...Next			
Do...Loop			
For Each...Next			

6. Письменный ответ на вопрос из Задания 4 (почему For Each лучше для диапазонов).
7. Выводы о проделанной работе.

6. Варианты индивидуальных заданий

- **Вариант 1:** Найти сумму всех четных чисел в диапазоне от 1 до N (вводится пользователем). Использовать For.
- **Вариант 2:** Вычислить факториал числа N. Использовать Do...Loop.
- **Вариант 3:** Найти все ячейки в выделенном диапазоне, содержащие текст "Ошибка", и удалить эти строки. Использовать For Each (с осторожностью при удалении) или обратный цикл For.
- **Вариант 4:** Реализовать игру «Угадай число». Компьютер загадывает число, пользователь угадывает. Цикл работает, пока число не угадано.

Лабораторная работа 4. Вставка пользовательских форм

1. Цель работы

1. Изучить назначение и возможности пользовательских форм (UserForm) в среде VBA.
2. Освоить интерфейс редактора форм и панель инструментов (Toolbox).
3. Научиться размещать элементы управления (Label, TextBox, CommandButton, ComboBox) и настраивать их свойства.
4. Изучить обработку событий элементов управления (Click, Initialize, Change).
5. Получить навыки передачи данных между формой и рабочим листом Excel.

2. Краткие теоретические сведения

Пользовательская форма (UserForm)

UserForm — это объект VBA, представляющий собой диалоговое окно, которое создается разработчиком для взаимодействия с пользователем. Формы позволяют создавать удобный графический интерфейс (GUI) для ввода данных, выбора параметров и вывода сообщений.

Элементы управления (Controls)

Основные элементы, используемые при создании форм:

- **Label (Надпись):** Для вывода статического текста (подписи к полям).
- **TextBox (Поле ввода):** Для ввода текстовой или числовой информации пользователем.
- **CommandButton (Кнопка):** Для запуска действий (например, «ОК», «Отмена»).
- **ComboBox (Выпадающий список):** Для выбора одного значения из предустановленного списка.
- **CheckBox (Флажок):** Для выбора варианта «Да/Нет» или включения опции.
- **Frame (Рамка):** Для визуального группирования элементов.

Свойства, События и Методы

- **Свойства (Properties):** Характеристики элемента (имя Name, заголовок Caption, значение Value/Text, видимость Visible).

- **События (Events):** Действия, на которые реагирует элемент (нажатие кнопки Click, загрузка формы Initialize, изменение текста Change).
- **Методы (Methods):** Команды, выполняемые над формой (Show — показать, Hide — скрыть, Unload — выгрузить из памяти).

Режимы отображения форм

- **Модальное окно (Modal):** Пользователь не может работать с Excel, пока не закроет форму (по умолчанию).
- **Немодальное окно (Modeless):** Пользователь может переключаться между формой и листом Excel (UserForm.Show vbModeless).

3. Задание на лабораторную работу

Студенту необходимо разработать приложение для ввода данных о товаре с использованием пользовательской формы.

Задание 1. Проектирование формы. Создать форму «Карточка товара», содержащую:

- Поля ввода: «Наименование товара» (TextBox), «Цена» (TextBox), «Количество» (TextBox).
- Выпадающий список: «Категория» (ComboBox) с предустановленными значениями (Электроника, Одежда, Продукты).
- Кнопки: «Добавить» (CommandButton), «Очистить» (CommandButton), «Заккрыть» (CommandButton).
- Подписи ко всем полям (Label).

Задание 2. Программирование логики.

1. При запуске формы (Initialize) очищать поля и заполнять список категорий.
2. При нажатии кнопки «Добавить»:
 - Проверить заполненность полей (валидация).
 - Проверить, что Цена и Количество — числа.
 - Записать данные на активный лист Excel в следующую свободную строку.
 - Очистить поля для ввода следующего товара.
3. При нажатии кнопки «Заккрыть» — выгрузить форму из памяти.
4. При нажатии кнопки «Очистить» — стереть введенные данные.

Задание 3. Запуск формы. Создать стандартный модуль с процедурой, которая открывает форму по кнопке на листе Excel или через макрос.

4. Порядок выполнения работы

Создание формы

1. Откройте редактор VBA (Alt + F11).
2. В меню выберите Insert -> UserForm. Появится пустое окно формы и панель Toolbox.
3. Перетащите элементы из Toolbox на форму согласно Заданию 1.
4. **Важно:** Измените свойства элементов в окне Properties (F4):
 - Для формы: Name = frmProduct, Caption = «Карточка товара».
 - Для кнопок: Name = btnAdd, btnClear, btnClose; Caption = «Добавить», «Очистить», «Заккрыть».
 - Для полей ввода: Name = txtName, txtPrice, txtQty.
 - Для списка: Name = cmbCategory.

Написание кода формы

Дважды щелкните по форме, чтобы открыть редактор кода.

1. **Событие инициализации:**

```

Private Sub UserForm_Initialize()
    ' Очистка полей
    txtName.Value = ""
    txtPrice.Value = ""
    txtQty.Value = ""

    ' Заполнение списка категорий
    With cmbCategory
        .AddItem "Электроника"
        .AddItem "Одежда"
        .AddItem "Продукты"
    End With
End Sub

```

2. Событие кнопки «Добавить»:

```

Private Sub btnAdd_Click()
    Dim nextRow As Long

    ' Проверка заполненности
    If txtName.Value = "" Or txtPrice.Value = "" Then
        MsgBox "Заполните все поля!", vbExclamation
        Exit Sub
    End If

    ' Проверка на число
    If Not IsNumeric(txtPrice.Value) Or Not IsNumeric(txtQty.Value) Then
        MsgBox "Цена и количество должны быть числами!", vbExclamation
        Exit Sub
    End If

    ' Поиск следующей свободной строки (предположим, лист Sheet1)
    nextRow = Sheets("Sheet1").Cells(Rows.Count, 1).End(xlUp).Row + 1

    ' Запись данных
    With Sheets("Sheet1")
        .Cells(nextRow, 1).Value = txtName.Value
        .Cells(nextRow, 2).Value = cmbCategory.Value
        .Cells(nextRow, 3).Value = txtPrice.Value
        .Cells(nextRow, 4).Value = txtQty.Value
    End With

    MsgBox "Товар добавлен!", vbInformation
    txtName.Value = "" ' Очистка после ввода
End Sub

```

3. Событие кнопки «Закрыть»:

```

Private Sub btnClose_Click()
    Unload Me
End Sub

```

4. Событие кнопки «Очистить»:

```

Private Sub btnClear_Click()
    txtName.Value = ""
    txtPrice.Value = ""
    txtQty.Value = ""
    cmbCategory.ListIndex = -1
    txtName.SetFocus ' Возврат курсора
End Sub

```

Запуск формы

1. Вставьте обычный модуль (Insert -> Module).
2. Напишите процедуру запуска:

```

Sub ShowForm()
    frmProduct.Show
End Sub

```

3. Вернитесь в Excel, назначьте этот макрос на кнопку на листе или запустите через меню Макросы.

Тестирование

1. Попробуйте ввести нечисловые значения в поля цены.
2. Попробуйте оставить поля пустыми.
3. Убедитесь, что данные корректно записываются в таблицу.

5. Требования к отчету

Отчет должен содержать:

1. Титульный лист.
2. Цель работы.
3. Скриншот формы в режиме конструктора (Design Mode) с видимой сеткой элементов.
4. Таблицу свойств основных элементов (Элемент, Name, Caption/Value).
5. Листинг кода формы и модуля запуска с комментариями.
6. Скриншоты работы программы:
 - Сообщение об ошибке валидации.
 - Заполненная форма.
 - Результат на листе Excel после добавления данных.
7. Выводы: преимущества использования форм перед InputBox, сложности, с которыми столкнулись.

6. Варианты индивидуальных заданий

- **Вариант А (Калькулятор):** Форма с двумя полями ввода и кнопками операций (+, -, *, /). Результат выводится в метку (Label) на форме.
- **Вариант Б (Авторизация):** Форма ввода логина и пароля. Пароль должен скрываться символами (свойство PasswordChar). При успешном входе открывается основной лист, при неудачном — сообщение.
- **Вариант В (Анкета):** Форма с использованием CheckBox (хобби) и OptionButton (пол). Результаты записываются в одну ячейку через запятую.
- **Вариант Г (Поиск):** Форма с полем ввода и кнопкой «Найти». Ищет введенное значение на листе и выделяет найденную ячейку.

Лабораторная работа 5. Данные для дашборда, анализ числовых данных показателей для построения дашбордов

1. Цель работы

1. Изучить принципы проектирования эффективных дашбордов и ключевые показатели эффективности (KPI).
2. Освоить методы предварительной обработки и очистки данных для визуализации.
3. Научиться рассчитывать агрегированные метрики и производные показатели для дашбордов.
4. Получить навыки анализа числовых данных: выявление трендов, аномалий, сравнение периодов.
5. Подготовить структурированный набор данных, готовый к визуализации в инструментах BI (Excel, Power BI, Tableau).

2. Краткие теоретические сведения

Понятие дашборда и его назначение

Дашборд (Dashboard) — это инструмент визуализации данных, представляющий ключевые метрики и показатели в удобном для восприятия формате на одной экранной панели.

Типы дашбордов:

Тип	Назначение	Пример использования
Операционный	Мониторинг процессов в реальном времени	Отслеживание заказов, нагрузка на кол-центр
Аналитический	Глубокий анализ данных, выявление причин	Анализ продаж по регионам, каналам, товарам
Стратегический	Отслеживание выполнения целей (KPI)	Финансовые показатели компании, доля рынка

Тактический	Контроль выполнения проектов/задач	Статус задач, загрузка ресурсов
-------------	------------------------------------	---------------------------------

Ключевые показатели (KPI) для дашбордов

KPI (Key Performance Indicator) — измеримая величина, демонстрирующая эффективность достижения целей.

Примеры метрик для бизнес-дашбордов:

- **Финансовые:** Выручка (Revenue), Прибыль (Profit), Маржинальность (%), Средний чек (AOV).
- **Продажи:** Количество сделок, Конверсия (%), LTV (пожизненная ценность клиента), CAC (стоимость привлечения).
- **Операционные:** Время обработки заказа, Процент брака, Загрузка оборудования.
- **Производные показатели:**
 - *Динамика (MoM, YoY):* (Текущий период - Прошлый период) / Прошлый период * 100%
 - *Доля в общем:* Значение категории / Сумма всех категорий * 100%
 - *План/Факт:* Фактическое значение / Плановое значение * 100%

Подготовка данных для дашборда

Качество дашборда напрямую зависит от качества данных. Этапы подготовки:

1. **Очистка данных:**
 - Удаление дубликатов.
 - Обработка пропущенных значений (замена средним, удаление, интерполяция).
 - Исправление форматов (даты, числа, текст).
 - Устранение выбросов (аномальных значений).
2. **Структурирование:**
 - Приведение к «длинному» формату (tidy data): одна строка — одно наблюдение.
 - Создание иерархий (Дата → Год/Квартал/Месяц; Категория → Подкатегория).
 - Нормализация текстовых полей (единый регистр, удаление лишних пробелов).
3. **Обогащение данных:**
 - Расчет новых полей (например, «Прибыль = Выручка - Себестоимость»).
 - Группировка (создание категорий по диапазонам: «0-1000», «1001-5000»).
 - Добавление справочников (регионы, менеджеры).

Принципы визуального восприятия

- **F-паттерн:** Важнейшие показатели размещаются в левом верхнем углу.
- **Правило 5±2:** Не более 5-7 ключевых метрик на одном экране.
- **Цветовая семантика:** Зеленый — позитив/рост, Красный — негатив/падение, Серый — нейтрально.
- **Контекст:** Число без сравнения (с планом, прошлым периодом) малоинформативно.

3. Задание на лабораторную работу

Студенту предоставляется исходный набор данных (например, «Продажи интернет-магазина» за год). Необходимо подготовить данные для создания дашборда руководителя отдела продаж.

Исходные данные (пример структуры):

Дата	Заказ	Товар	Категория	Регион	Менеджер	Количество	Цена_за_ед	Себестоимость
------	-------	-------	-----------	--------	----------	------------	------------	---------------

Задание 1. Очистка и проверка данных.

1. Проверить данные на дубликаты, пропуски, некорректные форматы.
2. Устранить найденные ошибки и зафиксировать их в отчете.

3. Привести текстовые поля к единому формату (регистр, пробелы).

Задание 2. Расчет показателей. Создать новые расчетные столбцы:

1. Выручка = Количество × Цена_за_ед.
2. Прибыль = (Цена_за_ед - Себестоимость) × Количество.
3. Маржа_% = Прибыль / Выручка × 100.
4. Период = извлечь из даты Год, Квартал, Месяц.

Задание 3. Агрегация и анализ. С помощью сводных таблиц (Pivot Tables) рассчитать:

1. **Итоговые метрики:** Общая выручка, общая прибыль, средняя маржа.
2. **Динамика:** Ежемесячная выручка и темп роста к предыдущему месяцу (MoM).
3. **Структура:** Доля выручки по категориям товаров и по регионам.
4. **Рейтинги:** Топ-5 товаров по прибыли, Топ-3 менеджера по выручке.
5. **Сравнение:** Факт выручки к плану (план задается отдельно или как константа).

Задание 4. Подготовка макета дашборда.

1. Сформулировать 3-5 ключевых вопросов, на которые должен отвечать дашборд.
2. Выбрать типы визуализации для каждого показателя (карточка, линейный график, столбчатая диаграмма, таблица).
3. Создать эскиз (wireframe) расположения элементов на листе.
4. Подготовить итоговый лист «Dashboard_Data» с агрегированными данными, готовыми к построению графиков.

4. Порядок выполнения работы

Подготовка рабочего места

1. Откройте файл с исходными данными в MS Excel.
2. Создайте копию листа для работы (чтобы сохранить оригинал).
3. Включите отображение формул (Ctrl + ~) для проверки расчетов.

Выполнение Задания 1 (Очистка данных)

1. **Поиск дубликатов:** Выделите диапазон → Данные → Удалить дубликаты.
2. **Поиск пропусков:** Используйте фильтр или условное форматирование для выделения пустых ячеек.
3. **Проверка форматов:**
 - Убедитесь, что столбец «Дата» имеет формат Дата.
 - Числовые столбцы — формат Числовой (без текста).
 - Используйте функцию ЕЧИСЛО() или ЕТЕКСТ() для проверки.
4. **Нормализация текста:**
excel

=ПРОПНАЧ(СЖПРОБЕЛЫ(A2)) ' Trim + Proper аналог

5. Зафиксируйте количество найденных и исправленных ошибок в отчете.

Выполнение Задания 2 (Расчет показателей)

Добавьте расчетные столбцы в исходную таблицу:

Формула (пример для Excel)	Описание
=[@Количество]*[@Цена_за_ед]	Выручка по строке
=([@Цена_за_ед]-[@Себестоимость])*[@Количество]	Прибыль по строке
=ЕСЛИ([@Выручка]=0;0;[@Прибыль]/[@Выручка])	Маржа (с защитой от деления на 0)
=ГОД([@Дата])	Год
=МЕСЯЦ([@Дата])	Номер месяца
=ТЕКСТ([@Дата];"ММММ")	Название месяца

Совет: Преобразуйте диапазон в «Умную таблицу» (Ctrl+T) — формулы протянутся автоматически.

Выполнение Задания 3 (Агрегация через сводные таблицы)

1. **Итоговые метрики:**
 - Создайте сводную таблицу.
 - Перетащите Прибыль в «Значения» (сумма).
 - Результат скопируйте на отдельный лист «Dashboard_Data».
2. **Динамика по месяцам:**
 - Строки: Дата (сгруппировать по месяцам).
 - Значения: Выручка (сумма).
 - Добавьте вычисляемое поле для MoM:

excel

1 =(Текущий_месяц - Предыдущий_месяц) / Предыдущий_месяц

Или используйте «Дополнительные вычисления» → «Разница» в сводной таблице.

3. **Структура по категориям:**
 - Строки: Категория.
 - Значения: Выручка (сумма).
 - Дополнительные вычисления → «Доля от суммы по столбцу».
4. **Рейтинг товаров:**
 - Строки: Товар.
 - Значения: Прибыль.
 - Фильтр: «Топ-10» → изменить на 5.

Выполнение Задания 4 (Макет и подготовка)

1. Сформулируйте вопросы дашборда:
 - «Какова общая выручка и динамика к прошлому месяцу?»
 - «Какие категории приносят наибольшую прибыль?»
 - «Кто из менеджеров выполняет план?»
2. Выберите типы графиков:
 - Карточки (Big Numbers) — для итоговых метрик.
 - Линейный график — для динамики во времени.
 - Столбчатая диаграмма — для сравнения категорий.
 - Таблица — для рейтинга менеджеров.
3. На новом листе расположите агрегированные данные из сводных таблиц в удобном для построения графиков формате.

5. Требования к отчету

Отчет должен содержать:

1. Титульный лист.
2. Цель работы.
3. **Описание исходных данных:** объем (строк/столбцов), описание полей, тип данных.
4. **Журнал очистки данных:** таблица найденных проблем и способов их решения.

Проблема	Пример	Способ исправления	Результат
Пропуски в «Регион»	5 строк	Заполнено значением «Не указан»	Исправлено

5. **Формулы расчета показателей:** скриншоты или листинг формул с пояснениями.
6. **Результаты агрегации:** скриншоты сводных таблиц с ключевыми метриками.
7. **Эскиз дашборда:** схема расположения элементов (можно от руки или в любом редакторе).
8. **Итоговый лист «Dashboard_Data»:** скриншот подготовленной таблицы.
9. **Выводы:**
 - Какие показатели оказались наиболее информативными?
 - С какими трудностями при подготовке данных столкнулись?
 - Как очищенные данные улучшат качество будущего дашборда?

6. Варианты индивидуальных заданий

Наборы данных и предметная область могут варьироваться:

Вариант	Предметная область	Ключевые метрики	Особенности задания
1	Продажи ритейла	Выручка, средний чек, конверсия	Анализ по филиалам и категориям товаров
2	Веб-аналитика	Посещения, отказы, время на сайте, цели	Работа с данными из Яндекс.Метрики/Google Analytics
3	Финансы компании	EBITDA, рентабельность, ликвидность	Расчет финансовых коэффициентов, сравнение с планом
4	Производство	Объем выпуска, брак, простой, OEE	Анализ причин простоев, динамика качества
5	Маркетинг	CTR, CPC, CPA, ROI кампаний	Сравнение эффективности каналов привлечения
6	Логистика	Время доставки, стоимость перевозки, загрузка	Анализ отклонений от нормативов
7	Персонал (HR)	Текучесть, время закрытия вакансий, удовлетворенность	Расчет индексов, работа с анонимизированными данными

КРИТЕРИИ ОЦЕНКИ

Оценка за лабораторную работу выставляется по 5-балльной шкале:

«5» (Отлично): Работа выполнена в полном объеме, отчет оформлен грамотно, студент свободно отвечает на контрольные вопросы, предложены собственные идеи по оптимизации.

«4» (Хорошо): Работа выполнена полностью, но есть мелкие недочеты в оформлении или ответах на вопросы.

«3» (Удовлетворительно): Работа выполнена частично, отчет содержит ошибки, ответы на вопросы неполные.

«2» (Неудовлетворительно): Работа не выполнена или отчет не сдан в срок.

Список использованных источников

1. Гуриков С.Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1913856>
2. Зыкова Г.В. Основы программирования на языке Python : учебно-методическое пособие / Г. В. Зыкова, А. С. Попов, Т. Н. Сапуглецева ; под. ред. Г. В. Зыковой. - 2-е изд., стер. - Москва : ФЛИНТА, 2020. - 135 с. - ISBN 978-5-9765-4430-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1860057>
3. Козлов А.Ю. Статистический анализ данных в MS Excel : учебное пособие / А.Ю. Козлов, В.С. Мхитарян, В.Ф. Шишов. — Москва : ИНФРА-М, 2023. — 320 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/2842. - ISBN 978-5-16-004579-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1907518>
4. Зиборов В.В. Visual Basic 2010 на примерах : [для программистов] .— СПб. : БХВ Петербург, 2010 .— 329 с. : ил., табл.+1CD.
5. Черников, Б. В Информационные технологии управления : учебник / Б.В. Черников. — 2-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2023. — 368 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0782-5. - Текст : электронный.
6. Ниматулаев, М. М. Информационные технологии в профессиональной деятельности : учебник / М.М. Ниматулаев. — Москва : ИНФРА-М, 2023. — 250 с. — (Высшее образование: Специалитет). - ISBN 978-5-16-016545-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1903327>