

Лабораторная работа № 2

ЗНАКОМСТВО СО СРЕДОЙ ПРОГРАММИРОВАНИЯ VBA

Цель работы: освоить принципы решения задач с использованием основных алгоритмических конструкций в *MS Excel* с помощью *VBA*.

Задачи работы

1. Изучить:

- интерфейс среды разработки *VBA*;
- команды *VBA* для организации ввода и вывода данных;
- команды *VBA* для преобразования типов данных.

2. Научиться:

- использовать команды ввода и вывода;
- работать с ячейками *Excel*;
- представлять линейный алгоритм в виде компьютерной программе на языке *VBA*;
- сохранять созданный проект *VBA*.

Перечень обеспечивающих средств

Задания лабораторной работы выполняются в программном обеспечении *MS Excel* 2013 и выше.

Общие теоретические сведения

Visual Basic for Applications (*VBA*, *Visual Basic* для приложений) – упрощенная реализация языка программирования *Visual Basic*, встроенная в линейку продуктов *Microsoft Office*, а также во многие другие программные пакеты, такие как *AutoCAD*, *SolidWorks*, *CorelDRAW*, *WordPerfect* и *ESRI ArcGIS*.

Для написания программного кода *VBA* достаточно открыть одно из приложений *MS Office*. В лабораторной работе программный код будет создаваться в *Excel*.

Код *VBA* хранится в проектах. Проекты содержат модули, а модули включают процедуры и функции.

Сохранение проекта производится только в файлах с поддержкой макросов: для *MS Excel* – в формате **.xlsm*.

Для начала работы в среде *VBA* необходимо в *MS Excel* включить на ленте вкладку Разработчик. Для этого в меню Файл нужно перейти в Параметры. В пункте Настроить ленту активировать вкладку Разработчик.

На вкладке Разработчик для работы нам понадобятся (рис. 1):

- кнопка *Visual Basic*, которая открывает редактор *VBA*;
- кнопка Вставить, позволяющая вставить элементы управления в документ;
- кнопка Режим конструктора для включения / отключения режима конструктора в проекте.

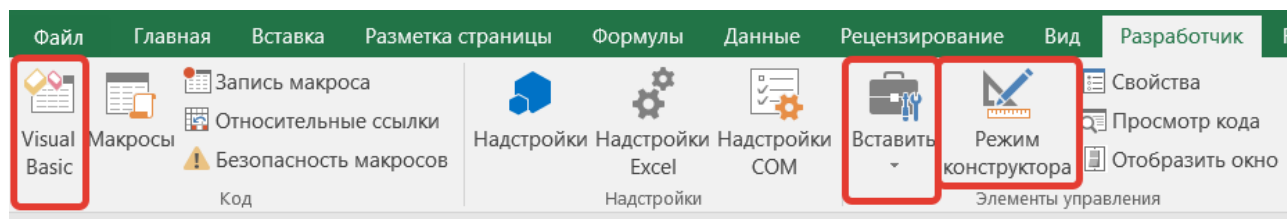


Рис. 1. Кнопки, необходимые для выполнения заданий

Рассмотрим пример создания программы для вычисления суммы двух чисел.

Сначала создадим объект для активации нашей будущей программы. Это будет кнопка на листе *Excel*. Чтобы создать кнопку на листе нужно перейти на вкладку Разработчик, открыть раскрывающийся список, нажав на элемент Вставить, и выбрать элемент Кнопка в группе Элементы *ActiveX* (рис. 2).

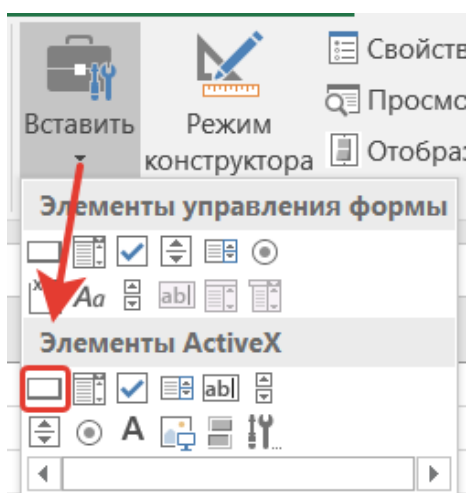


Рис. 2. Создание кнопки на листе *Excel*

Далее выделим небольшую область на листе, где будет расположена кнопка. В результате получим кнопку с текстом *CommandButton1* (рис. 3).

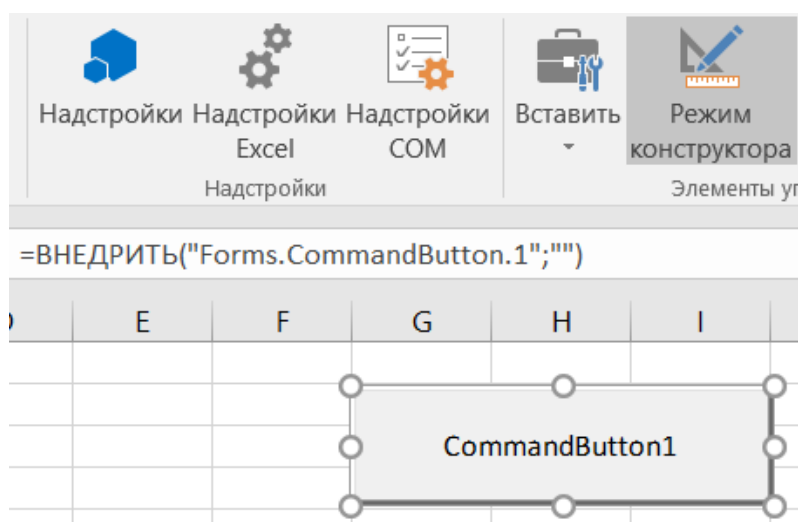


Рис. 3. Кнопка *CommandButton1* на листе *Excel*

Работа с элементами управления производится в двух режимах: в режиме конструктора, когда кнопка Режим конструктора на ленте активна, и в режиме использования, когда кнопка Режим конструктора не активна.

В активном Режиме конструктора можно изменять местоположение кнопки и ее свойства (размеры, надпись на кнопке и др.). Для изменения свойств кнопки нужно выбрать команду Свойства в группе Элементы управления на вкладке Разработчик. Откроется окно *Properties* (рис. 4).

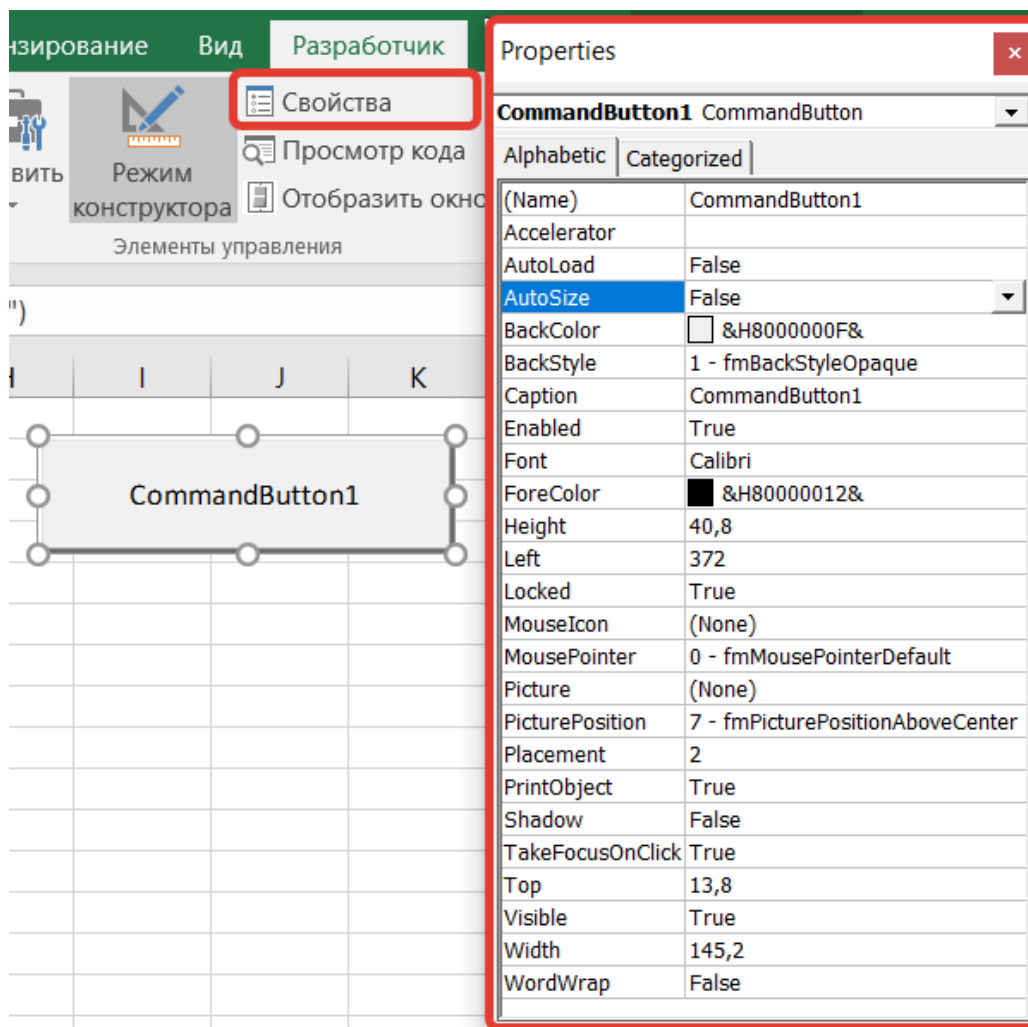


Рис. 4. Окно *Properties*

Рассмотрим свойства, которые отвечают за размеры кнопки и надпись на кнопке. Установим размеры кнопки: 150 пикселей по ширине и 50 пикселей по высоте. Для этого изменим значения свойств *Width* (ширина) и *Height* (высота). Установим на кнопке надпись в виде текста: Пример 1. Для этого изменим значение свойства *Caption*. Установим размер шрифта надписи 16 пунктов с помощью свойства *Font*.

Окно *Properties* после установки всех необходимых значений свойств можно закрыть.

Деактивируем команду Режим конструктора на вкладке Разработчик. Затем нажмем на нашу кнопку на листе *Excel*. При нажатии на кнопку ничего не происходит, так как мы не написали программный код, который будет исполняться при нажатии на кнопку.

Чтобы при нажатии на кнопку выполнялись какие-либо действия, нужно создать событие и написать программный код.

Снова активируем Режим конструктора и нажмем 2 раза на кнопку, откроется окно Редактор кода (рис. 5).

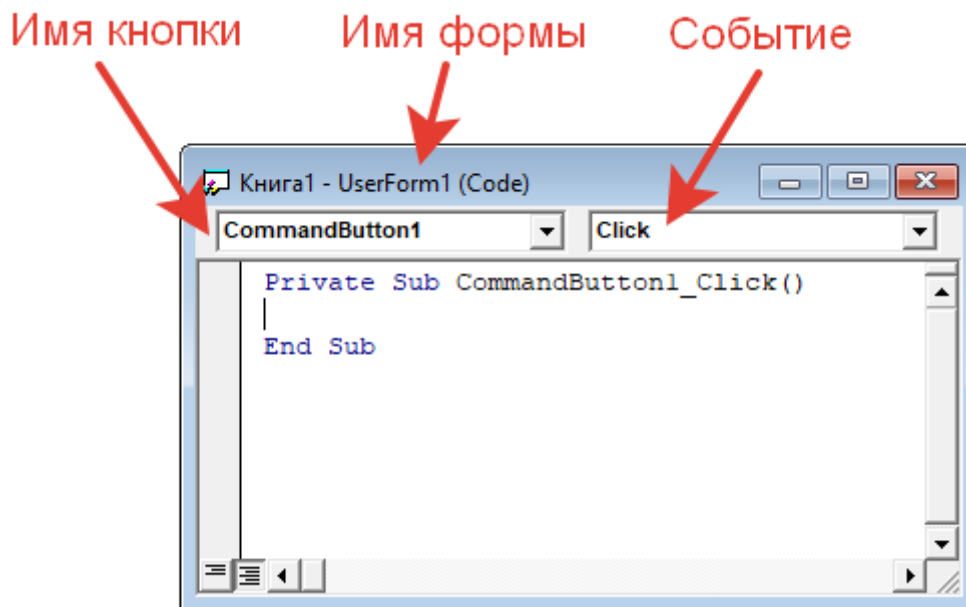


Рис. 5. Окно редактора кода

В редакторе кода можно выбирать из списка имя объекта, для которого создается событие (в нашем случае – это кнопка *CommandButton1*), в другом списке находятся названия событий. При двойном нажатии на кнопку автоматически формируется событие *Click*. То есть программный код будет выполняться при нажатии на кнопку.

Первая строка кода – это начало процедуры обработчика события. Третья строка – завершение процедуры.

Между ними находится курсор, указывая, что в этом месте мы должны писать программный код для события *Click*.

Несколько правил оформления кода:

- строки кода внутри программных конструкций принято писать с отступом. Это облегчает восприятие программы;
- длинные строки можно разрывать на более короткие, ставя в конце этих строк знак «_» (символ подчеркивания);

– для удобства можно разбивать различные блоки программы пустыми строками;

– все служебные команды можно писать строчными буквами. Редактор кода автоматически преобразует их в нужный вид. Например, если написать *msgbox*, то редактор кода преобразует эту команду в *MsgBox*.

Познакомимся, как в программном коде оформляются комментарии. Комментарии поясняют работу программы и не исполняются.

Любой комментарий начинается с апострофа (одинарная кавычка) и отображается в редакторе кода зеленым цветом (рис. 6).

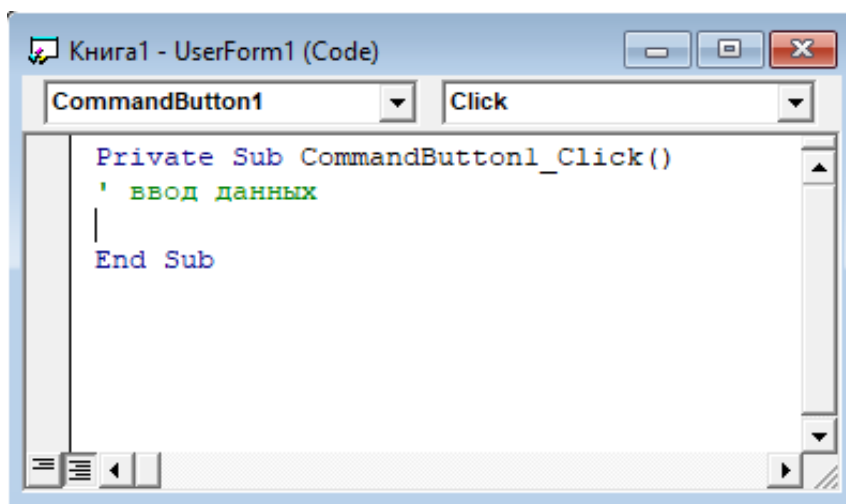


Рис. 6. Оформление комментария

Ввод и вывод данных при работе в *MS Excel* можно организовать тремя способами.

1. Использовать для ввода и вывода данных системные окна *VBA*.
2. Использовать для исходных данных и для вывода результата ячейки *Excel*.
3. С помощью объектов *TextBox* (для ввода данных) и *Label* (для вывода данных).

В этой лабораторной работе мы будем использовать только 1 и 2 способы ввода и вывода данных.

Сначала рассмотрим команды ввода и вывода данных с использованием системных окон. Составим программу, которая запрашивает у пользователя два

числа с помощью функции *InputBox*, складывает их и выводит результат с помощью функции *MsgBox*.

Команда ввода *InputBox* открывает специальное окно для ввода данных. Введенные данные должны храниться в определенной переменной, поэтому нужно обязательно указывать имя переменной. Имя переменной должно отвечать тем же требованиям, что и имя формы. Желательно давать переменным осмысленные имена. Программный код представлен на рис. 7.

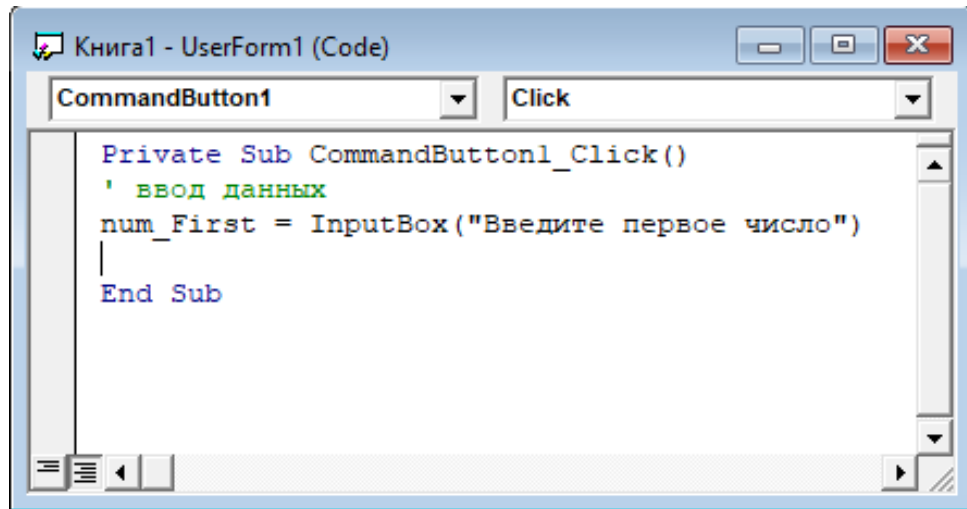


Рис. 7. Ввод данных

Запустим программу и введем любое число (рис. 8).

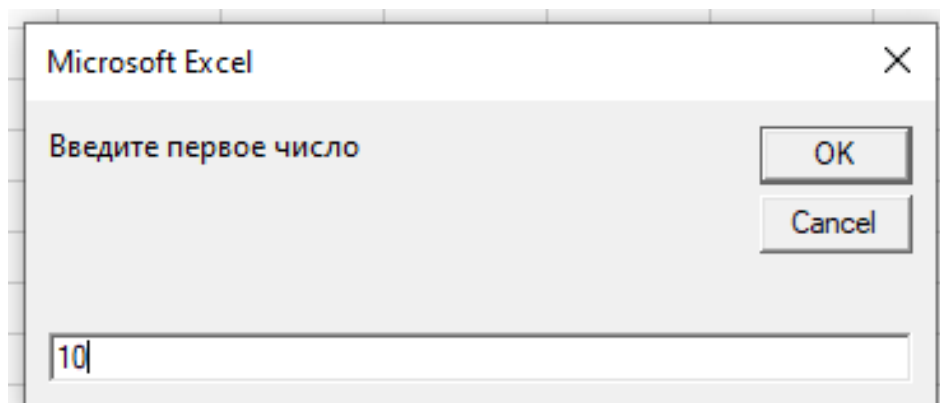


Рис. 8. Ввод данных при запуске программы

После нажатия кнопки *OK* переменной *num_First* будет присвоено значение 10.

Проведем аналогичные действия для ввода второго числа в переменную *num_Second*.

Затем напишем команду для вычисления суммы первого и второго числа (рис. 9).

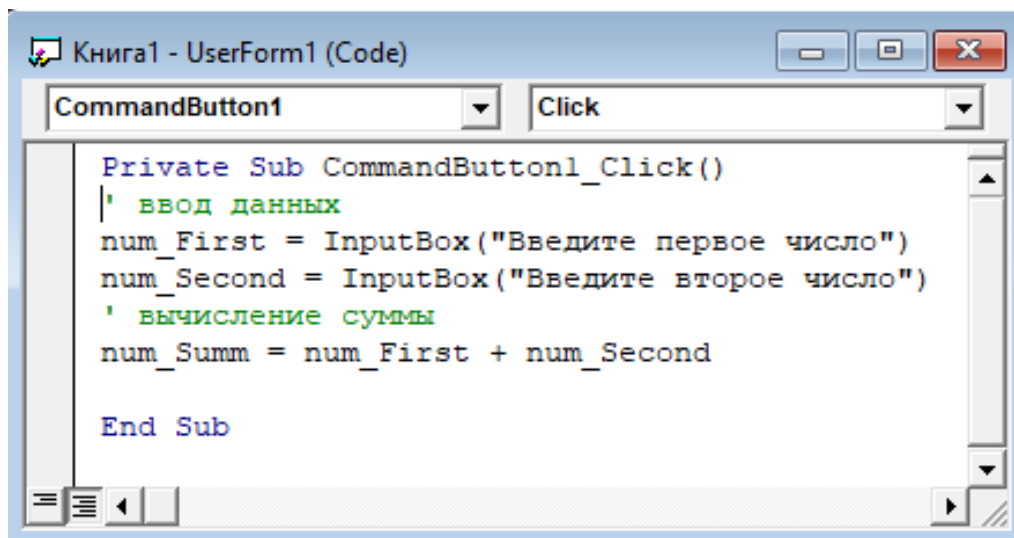


Рис. 9. Ввод данных и вычисление суммы

Напишем команду для вывода результата вычислений. Для вывода результата в системном окне используется команда *MsgBox* (рис. 10).

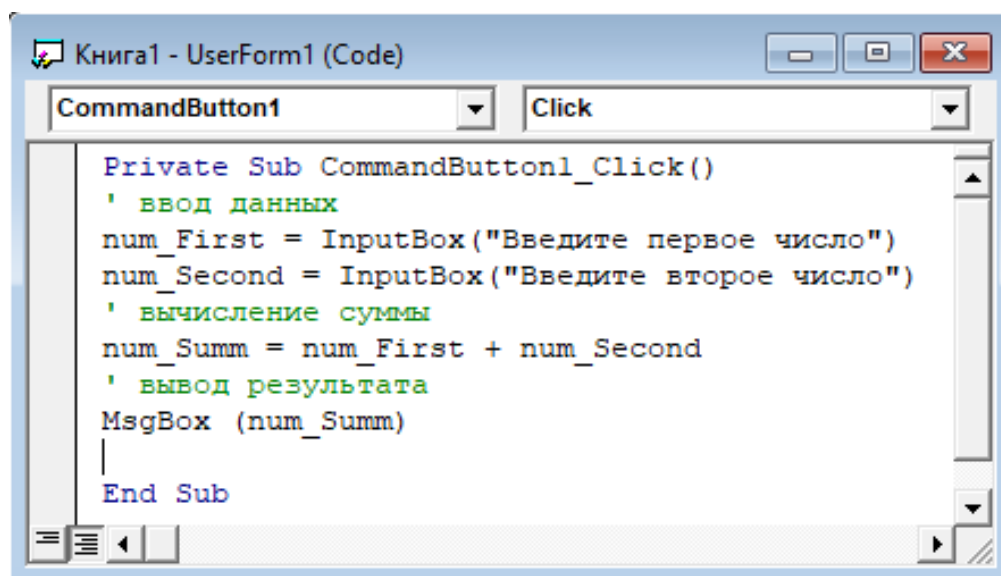


Рис. 10. Вывод результата

Запустим программу. Введем числа 10 и 20. Посмотрим результат (рис. 11).

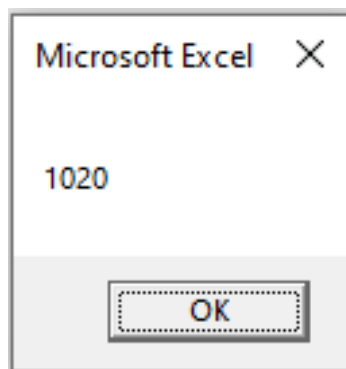


Рис. 11. Запуск программы

Мы видим, что результат получился не совсем корректным с точки зрения правил сложения математики. Однако, с точки зрения программирования в VBA результат корректен. Дело в том, что в процессе ввода данных любые введенные символы относятся к типу «текстовые данные» (тип *String*), даже несмотря на то, что мы вводили числа. Символ «+» для текстовых данных обозначает слияние. Так мы получили соединение двух символов «10» и еще двух символов «20». В итоге получили текст «1020».

Чтобы программа работала корректно с точки зрения правил математики, нужно явно указать, что вводимые данные являются числами (тип *Double*). Для этого нужно преобразовать тип данных, используя функция *Val* (рис. 12).

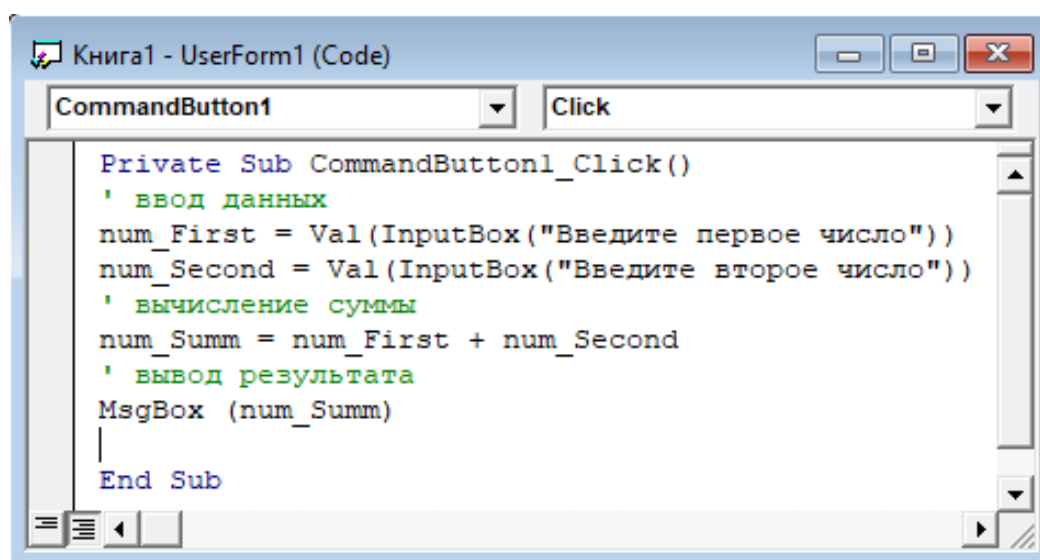


Рис. 12. Преобразование типов данных

Теперь после выполнения программного кода мы получим числовой результат «30».

Добавим поясняющий текст в команду вывода (рис. 13), запустим программу и посмотрим результат (рис. 14).

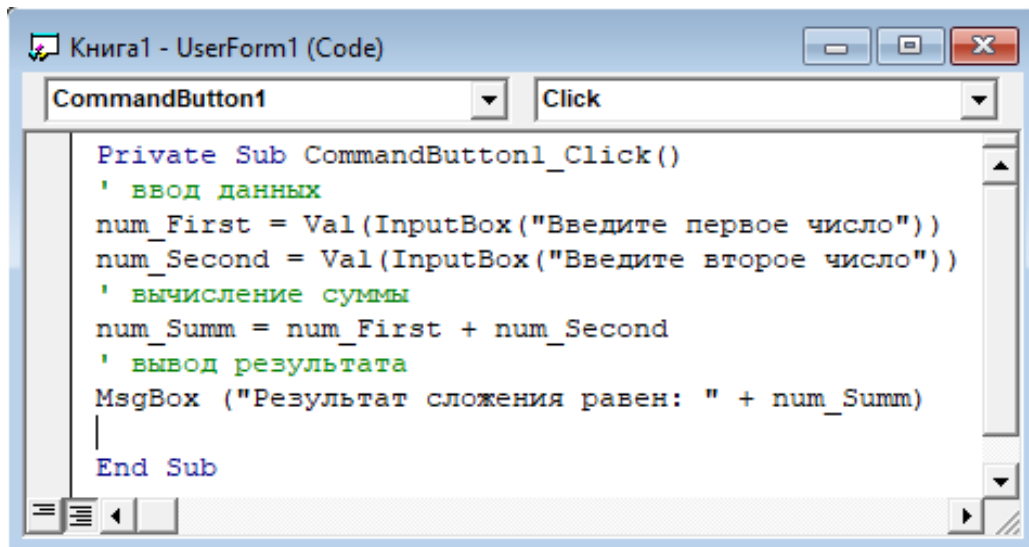


Рис. 13. Поясняющий текст в команде вывода

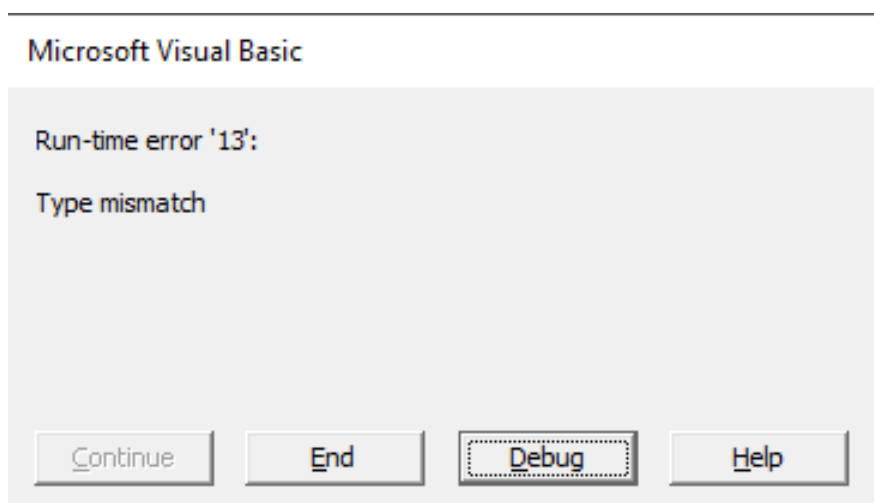


Рис. 14. Ошибка во время выполнения программы

Мы столкнулись с ошибкой *Type mismatch* (несоответствие типов).

Если нажать на кнопку *End*, программа будет завершена, и мы вернемся в окно редактора кода.

При нажатии на кнопку *Debug* мы также вернемся в редактор кода, но желтым цветом будет подсвечена строка с ошибкой (рис. 15).

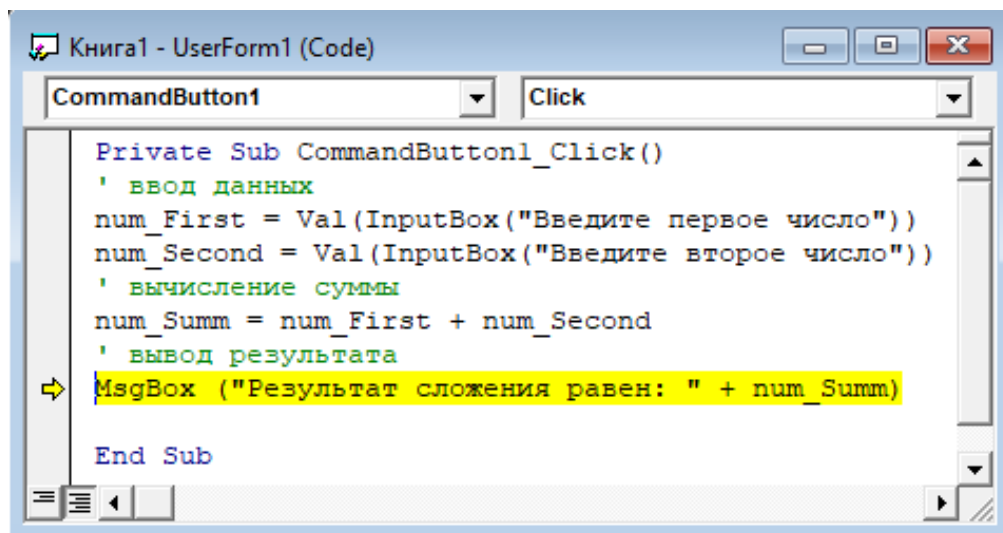


Рис. 15. Режим *Debug*

Ошибка состоит в следующем. Внутри команды вывода с помощью символа «+» соединяются данные разных типов. В первой части в кавычках – данные типа *String*, а во второй части (переменная *num_Summ*) – данные типа *Double*. Таким образом, выполнить такую команду невозможно.

Для исправления ошибки нужно привести все части к одному типу данных. В этом случае это будет тип *String*.

Для преобразования данных из типа *Double* в тип *String* используется функция *Str*. Итоговый программный код представлен на рис. 16. Запустим программу, нажав на кнопку на листе *Excel*, введем корректные данные (числа) и получим результат. Сохраним файл *Excel* в формате с поддержкой макросов.

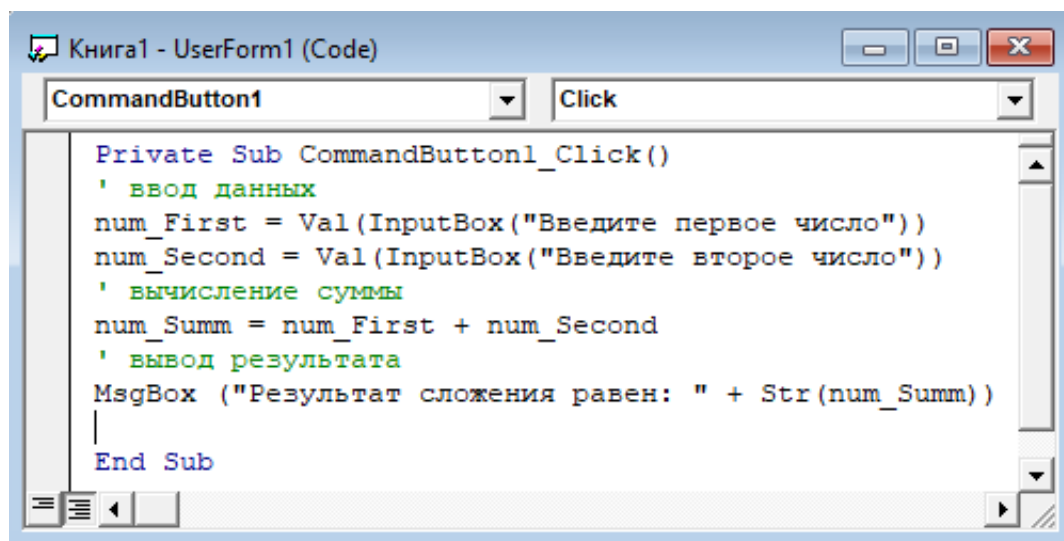


Рис. 16. Приведение данных к одному типу *String* в команде вывода

Рассмотрим второй способ ввода и вывода данных – с использованием ячеек *MS Excel*.

Условие задачи остается то же самое: ввести два числа, вычислить сумму и вывести результат. Исходные данные пользователь будет вводить в ячейки, результат также выводится в ячейке, кнопка для запуска программы будет располагаться на листе *Excel*.

На листе 1 в *Excel* напишем поясняющий текст: в ячейке A1 – Число 1; в ячейке A2 – Число 2; в ячейке A4 – Сумма.

Разместим на листе кнопку с надписью Вычислить сумму.

Интерфейс нашей программы представлен на рис. 17.

	A	B	C
1	Число 1		
2	Число 2		
3			
4	Сумма		
5		Вычислить сумму	
6			
7			

Рис. 17. Интерфейс программы

Убедимся, что Режим конструктора активирован.

Дважды щелкнем на кнопке, чтобы создать событие *Click*. Откроется редактор кода.

Напишем программный код (рис. 18).

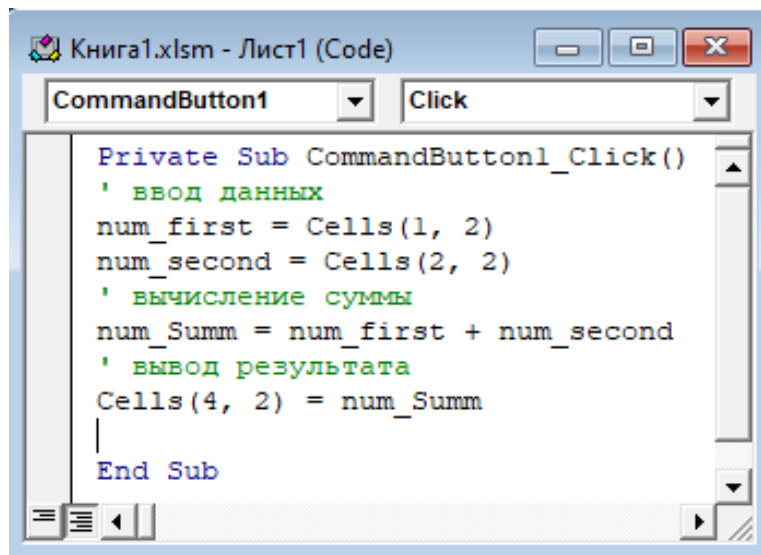


Рис. 18. Программный код для вычисления суммы чисел

Для обращения к данным в ячейках используется функция *Cells*. Аргументами функции являются порядковый номер строки и столбца. Таким образом, мы задали, что первое число находится в ячейке *B1* (1 строка и 2 столбец), второе число – в ячейке *B2* (2 строка и 2 столбец), а результат выводится в ячейку *B4* (4 строка и 2 столбец).

Закроем редактор кода и запустим программу. Для этого деактивируем Режим конструктора. Введем в ячейки *B1* и *B2* числа 10 и 20, нажмем на кнопку и проверим результат (должно получиться 30).

Программа работает корректно. Сохраним файл *Excel* в формате с поддержкой макросов.

Задания

Задание 1. Написать программу для решения математической задачи (номер задачи соответствует номеру варианта). В первом случае для ввода и вывода данных использовать функции *InputBox* и *MsgBox*, во втором случае организовать ввод и вывод данных с помощью ячеек *Excel*.

В программном коде должны быть комментарии. В первом комментарии укажите свою фамилию.

Для числа π использовать функцию рабочего листа *WorksheetFunction.Pi*.

Для вычисления квадратного корня использовать функцию *sqr*.

1. Дана площадь круга. Найти длину окружности, ограничивающей этот круг.

2. Вычислить медианы треугольника со сторонами a, b, c по формулам:

$$m_a = 0,5\sqrt{2b^2 + 2c^2 - a^2};$$

$$m_b = 0,5\sqrt{2a^2 + 2c^2 - b^2};$$

$$m_c = 0,5\sqrt{2b^2 + 2a^2 - c^2};$$

3. Вычислить площадь круга и длину окружности по введенному значению радиуса.

4. Вычислить площадь S и периметр L эллипса по введенным значениям полуосей a и b :

$$S := \pi \cdot a \cdot b;$$

$$L = 2 \cdot \pi \cdot \sqrt{\frac{1}{2}(a^2 + b^2)}.$$

5. Дано целое четырехзначное число. Используя операции *div* и *mod*, найти сумму его цифр.

6. Вычислить объем V и площадь боковой поверхности цилиндра S по введенным значениям радиуса основания R и высоты цилиндра H .

$$V = \pi \cdot R^2 \cdot H;$$

$$S = 2 \cdot \pi \cdot R \cdot H.$$

7. Вычислить объем V и площадь поверхности S сферы по введенному значению радиуса r :

$$V = \frac{4}{3} \cdot \pi \cdot r^3;$$

$$S = 4 \cdot \pi \cdot r^2.$$

8. Дана сторона равностороннего треугольника. Найти площадь этого треугольника и радиусы вписанной и описанной окружностей.

9. Даны координаты трех вершин треугольника $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Найти его периметр и площадь.

10. Дана длина окружности. Найти площадь круга, ограниченного этой окружностью.

Задание 2. Написать программу для решения математической задачи (номер задачи соответствует номеру варианта). В первом случае для ввода и вывода данных использовать функции *InputBox* и *MsgBox*, во втором случае организовать ввод и вывод данных с помощью ячеек *Excel*.

В программном коде должны быть комментарии. В первом комментарии укажите свою фамилию.

1. Найти сумму вычисленных объемов двух шаров.
2. Найти произведение вычисленных площадей трех ромбов.
3. Найти квадрат суммы вычисленных гипотенуз трех треугольников.
4. Найти произведение вычисленных площадей трех параллелограммов.
5. Возвести в куб вычисленные объемы двух цилиндров.
6. Найти сумму вычисленных площадей трех прямоугольников.
7. Вычислить значение квадрата произведения площадей трех овалов.
8. Найти произведение вычисленных площадей трех кругов.
9. Найти сумму кубов объемов трех цилиндров.
10. Возвести в квадрат произведение площадей трех трапеций.

Контрольные вопросы

1. Как включить вкладку Разработчик на ленте?
2. Как правильно сохранять созданные проекты *VBA*?
3. Как правильно оформляются комментарии в программном коде?
4. Где устанавливаются свойства объектов?
5. Для чего используются функции *InputBox* и *MsgBox*?
6. В чем назначение функций *Val* и *Str*?
7. Как происходит обращение к ячейкам *Excel*?

Содержание отчета

1. Титульный лист
2. Цель работы.
3. Номер варианта.

4. Формулировка задания.

5. Скриншоты разработанного интерфейса программ, программного кода и процесса работы программы с двумя наборами исходных данных (2 раза запустить на выполнение каждую программу).

6. Ответы на контрольные вопросы.

7. Общий вывод о проделанной работе.