

Создание макросов и пользовательских функций на VBA

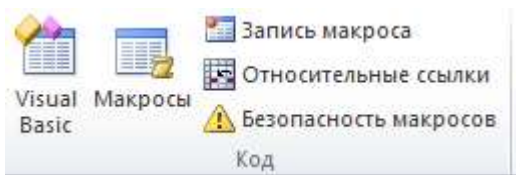
Макрос - это запрограммированная последовательность действий (программа, процедура), записанная на языке программирования Visual Basic for Applications (VBA). Мы можем запускать макрос сколько угодно раз, заставляя Excel выполнять последовательность любых нужных нам действий, которые нам не хочется выполнять вручную.

Существует множество языков программирования (Pascal, Fortran, C++, C#, Java, ASP, PHP...), но для всех программ пакета Microsoft Office стандартом является именно встроенный язык VBA. Команды этого языка понимает любое офисное приложение, будь то Excel, Word, Outlook или Access.

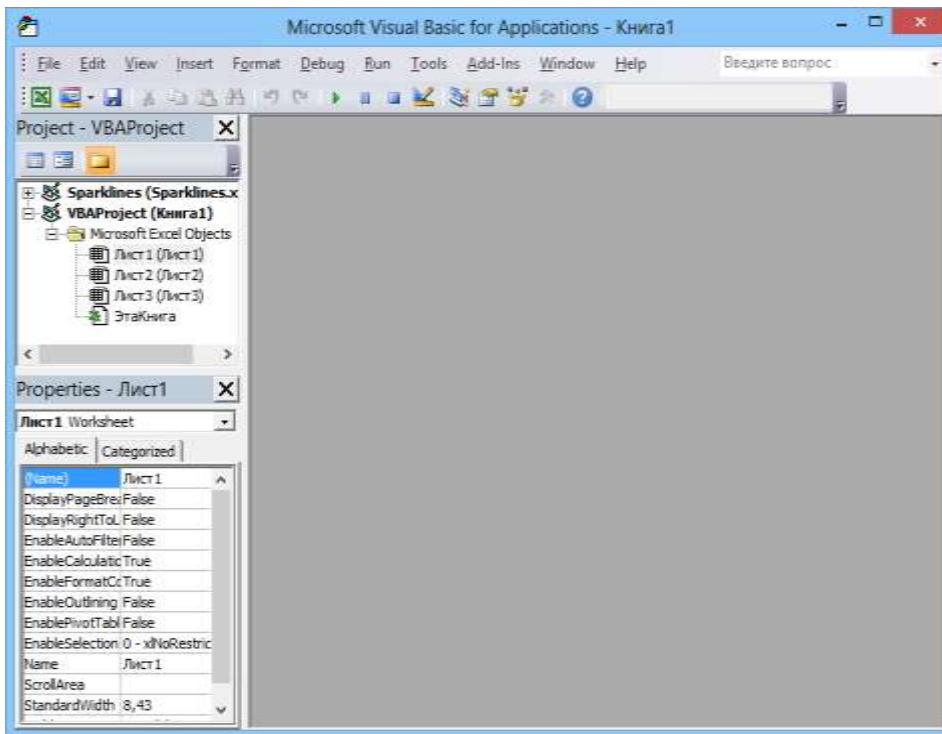
1 Создание макросов в редакторе Visual Basic

Для ввода команд и формирования программы, т.е. создания макроса необходимо открыть специальное окно - редактор программ на VBA, встроенный в Microsoft Excel.

- В новых версиях (Excel 2010 и новее) нужно сначала отобразить вкладку **Разработчик (Developer)**. Выбираем **Файл - Параметры - Настройка ленты (File - Options - Customize Ribbon)** и включаем в правой части окна флажок **Разработчик (Developer)**. Теперь на появившейся вкладке нам будут доступны основные инструменты для работы с макросами, в том числе и нужная нам кнопка **Редактор Visual Basic (Visual Basic Editor)**

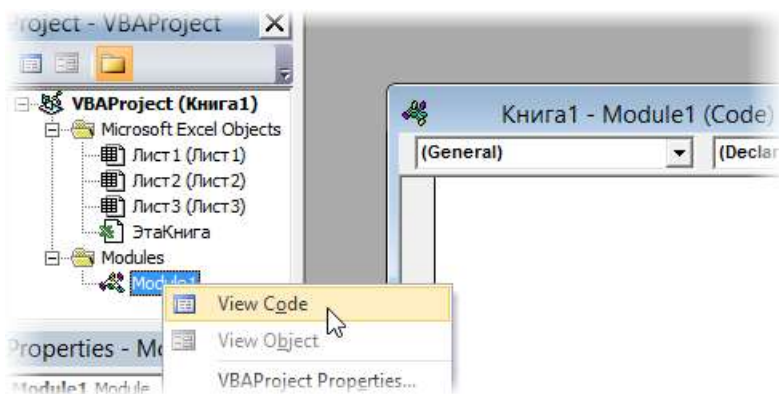


К сожалению, интерфейс редактора VBA и файлы справки не переводятся компанией Microsoft на русский язык, поэтому с английскими командами в меню и окнах придется смириться:



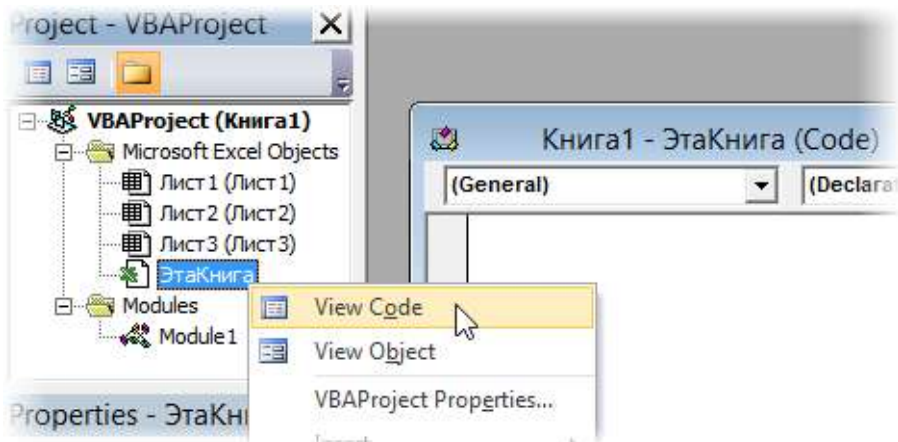
Макросы (т.е. наборы команд на языке VBA) хранятся в программных модулях. В любой книге Excel мы можем создать любое количество программных модулей и разместить там наши макросы. Один модуль может содержать любое количество макросов. Доступ ко всем модулям осуществляется с помощью окна Project Explorer в левом верхнем углу редактора (если его не видно, нажмите CTRL+R). Программные модули бывают нескольких типов для разных ситуаций:

Обычные модули - используются в большинстве случаев, когда речь идет о макросах. Для создания такого модуля выберите в меню Insert - Module. В появившееся окно нового пустого модуля можно вводить команды на VBA, набирая их с клавиатуры или копируя их из другого модуля:

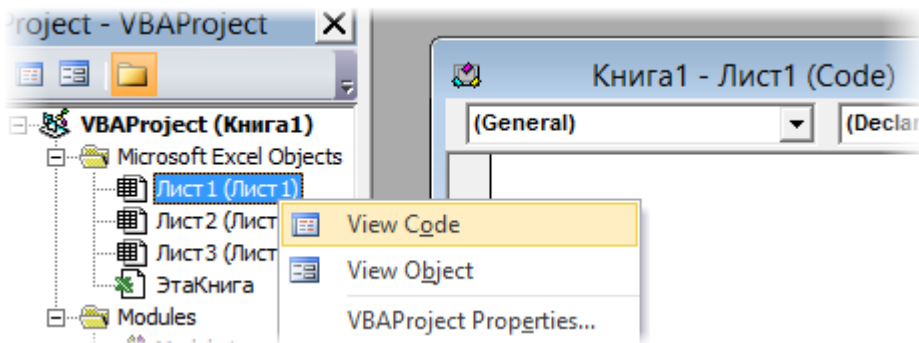


- **Модуль Эта книга** - также виден в левом верхнем углу редактора Visual Basic в окне, которое называется Project Explorer. В этот модуль обычно записываются макросы, которые выполняются при наступлении каких-либо событий в книге (открытие или

сохранение книги, печать файла и т.п.):



- **Модуль листа** - доступен через Project Explorer и через контекстное меню листа, т.е. правой кнопкой мыши по ярлычку листа - команда **Исходный текст (View Source)**. Сюда записывают макросы, которые должны выполняться при наступлении определенных событий на листе (изменение данных в ячейках, пересчет листа, копирование или удаление листа и т.д.)



Обычный макрос, введенный в стандартный модуль, выглядит примерно так:

```
Книга1 - Module1 (Code)
(General) Zamena
Sub Zamena ()
    Selection.Interior.ColorIndex = 6
    For Each cell In Selection
        cell.Formula = cell.Value
    Next cell
    MsgBox "Все формулы заменены на значения"
End Sub
```

Давайте разберем приведенный выше в качестве примера макрос Zamena:

- Любой макрос должен начинаться с оператора Sub, за которым идет имя макроса и список аргументов (входных значений) в скобках. Если аргументов нет, то скобки надо оставить пустыми.
- Любой макрос должен заканчиваться оператором End Sub.
- Все, что находится между Sub и End Sub - тело макроса, т.е. команды, которые будут выполняться при запуске макроса. В данном случае макрос выделяет ячейку заливаает выделенных диапазон (Selection) желтым цветом (код = 6) и затем проходит в цикле по всем ячейкам, заменяя формулы на значения. В конце выводится окно сообщения (MsgBox).

Ясно, что вот так сразу, без предварительной подготовки и опыта в программировании вообще и на VBA в частности, будет сложно сообразить какие именно команды и как надо вводить, чтобы макрос автоматически выполнял все действия, которые, например, Вы делаете для создания еженедельного отчета для руководства компании. Поэтому перейдем ко второму способу создания макросов.

2 Запись макросов макрорекордером

Макрорекордер - это небольшая программа, встроенная в Excel, которая переводит любое действие пользователя на язык программирования VBA и записывает получившуюся команду в программный модуль. Если мы включим макрорекордер на запись, а затем начнем создавать свой еженедельный отчет, то макрорекордер начнет записывать команды вслед за каждым нашим действием и, в итоге, мы получим макрос создающий отчет как если бы он был написан программистом.

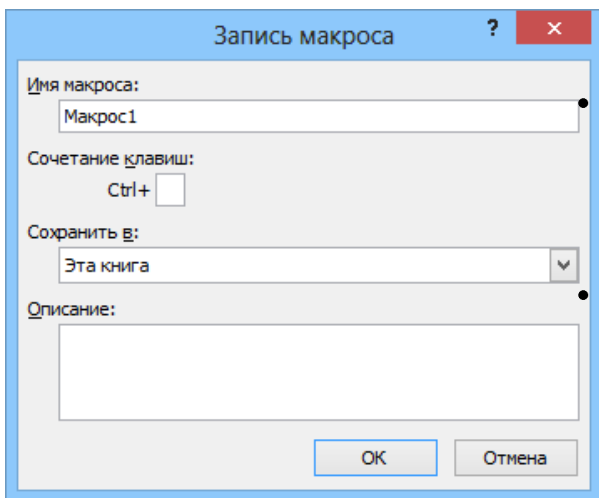
Такой способ создания макросов не требует знаний пользователя о программировании и VBA и позволяет пользоваться макросами как неким аналогом видеозаписи: включил запись, выполнил операцию, перемотал пленку и запустил выполнение тех же действий еще раз. Естественно у такого способа есть свои плюсы и минусы:

- Макрорекордер записывает только те действия, которые выполняются в пределах окна Microsoft Excel. Как только вы закрываете Excel или переключаетесь в другую программу - запись останавливается.
- Макрорекордер может записать только те действия, для которых есть команды меню или кнопки в Excel. Программист же может написать макрос, который делает то, что Excel никогда не умел (сортировку по цвету, например или что-то подобное).
- Если во время записи макроса макрорекордером вы ошиблись - ошибка будет записана. Однако смело можете давить на кнопку отмены последнего действия

(Undo) - во время записи макроса макрорекордером она не просто возвращает Вас в предыдущее состояние, но и стирает последнюю записанную команду на VBA.

Чтобы включить запись необходимо:

- в Excel 2010 и новее - нажать кнопку **Запись макроса (Record macro)** на вкладке **Разработчик (Developer)**
Затем необходимо настроить параметры записываемого макроса в окне **Запись макроса**:



Имя макроса - подойдет любое имя на русском или английском языке. Имя должно начинаться с буквы и не содержать пробелов и знаков препинания.

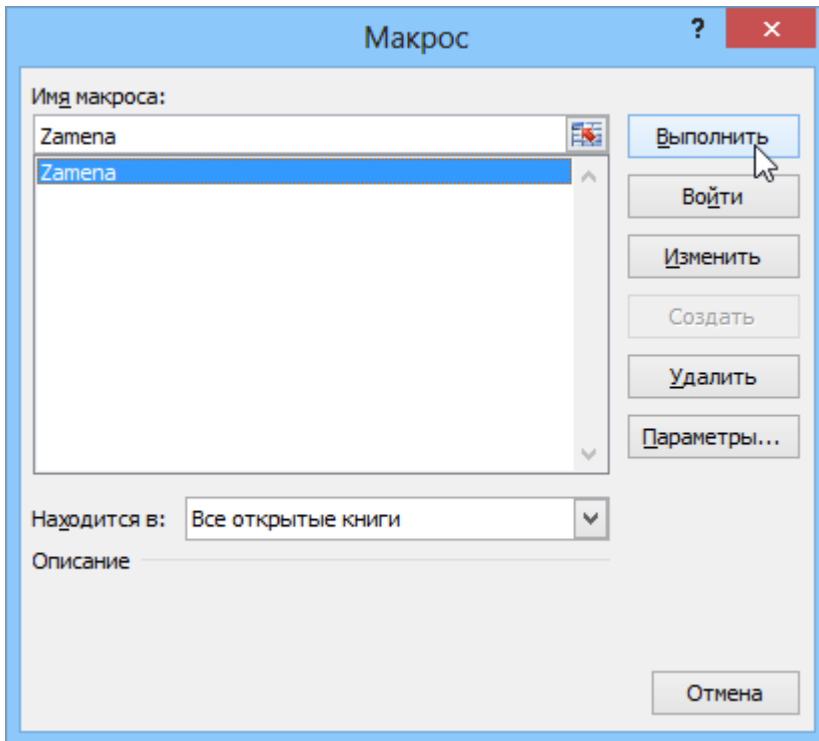
Сочетание клавиш - будет потом использоваться для быстрого запуска макроса. Если забудете сочетание или вообще его не введете, то макрос можно будет запустить через меню **Сервис -**

Макрос - Макросы - Выполнить (Tools - Macro - Macros - Run) или с помощью кнопки **Макросы (Macros)** на вкладке **Разработчик (Developer)** или нажав ALT+F8.

- **Сохранить в...** - здесь задается место, куда будет сохранен текст макроса, т.е. набор команд на VBA из которых и состоит макрос:
 - *Эта книга* - макрос сохраняется в модуль текущей книги и, как следствие, будет выполняться только пока эта книга открыта в Excel
 - *Новая книга* - макрос сохраняется в шаблон, на основе которого создается любая новая пустая книга в Excel, т.е. макрос будет содержаться во всех новых книгах, создаваемых на данном компьютере начиная с текущего момента
 - *Личная книга макросов* - это специальная книга Excel с именем *Personal.xls*, которая используется как хранилище макросов. Все макросы из *Personal.xls* загружаются в память при старте Excel и могут быть запущены в любой момент и в любой книге.
После включения записи и выполнения действий, которые необходимо записать, запись можно остановить командой **Остановить запись (Stop Recording)**.

3 Запуск и редактирование макросов

Управление всеми доступными макросами производится в окне, которое можно открыть с помощью кнопки **Макросы (Macros)** на вкладке **Разработчик (Developer)** или - в старых версиях Excel - через меню **Сервис - Макрос - Макросы (Tools - Macro - Macros)**:



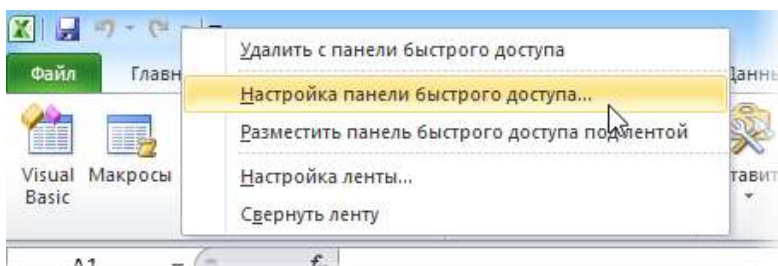
- Любой выделенный в списке макрос можно запустить кнопкой **Выполнить (Run)**.
- Кнопка **Параметры (Options)** позволяет посмотреть и отредактировать сочетание клавиш для быстрого запуска макроса.
- Кнопка **Изменить (Edit)** открывает редактор Visual Basic (см. выше) и позволяет посмотреть и отредактировать текст макроса на VBA.

4 Создание кнопки для запуска макросов

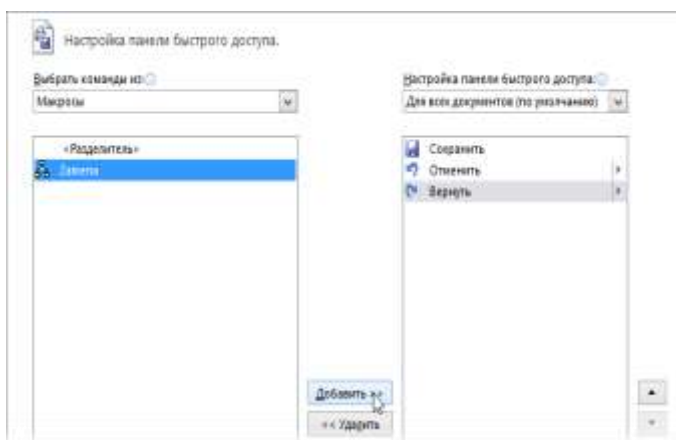
Чтобы не запоминать сочетание клавиш для запуска макроса, лучше создать кнопку и назначить ей нужный макрос. Кнопка может быть нескольких типов:

4.1 Кнопка на панели быстрого доступа в Excel 2010 и новее

Щелкните правой кнопкой мыши по панели быстрого доступа в левом верхнем углу окна Excel и выберите команду **Настройка панели быстрого доступа(Customise Quick Access Toolbar)**:



Затем в открывшемся окне выберите категорию **Макросы** и при помощи кнопки **Добавить (Add)** перенесите выбранный макрос в правую половину окна, т.е. на панель быстрого доступа:

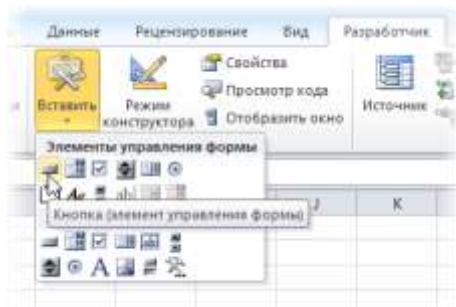


4.2 Кнопка на листе

Этот способ подходит для любой версии Excel. Мы добавим кнопку запуска макроса прямо на рабочий лист, как графический объект. Для этого:

- В Excel 2010 и новее - откройте выпадающий список **Вставить (Insert)** на вкладке **Разработчик (Developer)**

Выберите объект **Кнопка (Button)**:

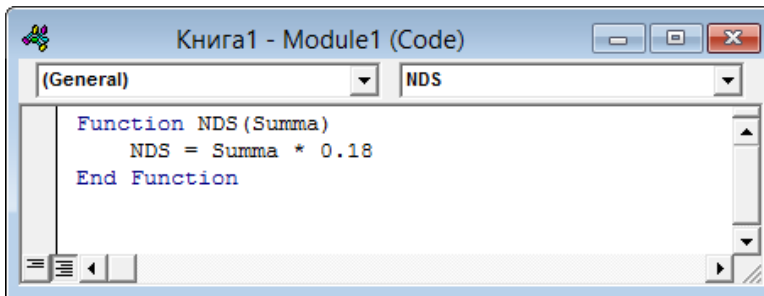


Затем нарисуйте кнопку на листе, удерживая левую кнопку мыши. Автоматически появится окно, где нужно выбрать макрос, который должен запускаться при щелчке по нарисованной кнопке.

5 Создание пользовательских функций на VBA

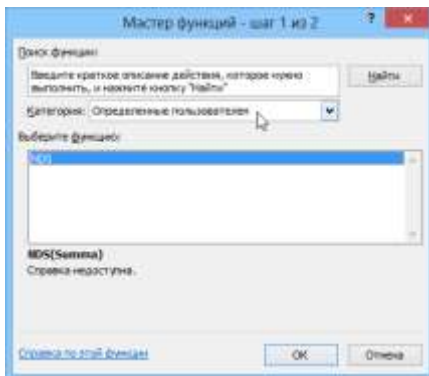
Создание пользовательских функций или, как их иногда еще называют, UDF-функций (User Defined Functions) принципиально не отличается от создания макроса в обычном программном модуле. Разница только в том, что макрос выполняет последовательность действий с объектами книги (ячейками, формулами и значениями, листами, диаграммами и т.д.), а пользовательская функция - только с теми значениями, которые мы передадим ей как аргументы (исходные данные для расчета).

Чтобы создать пользовательскую функцию для расчета, например, налога на добавленную стоимость (НДС) откроем редактор VBA, добавим новый модуль через меню **Insert - Module** и введем туда текст нашей функции:

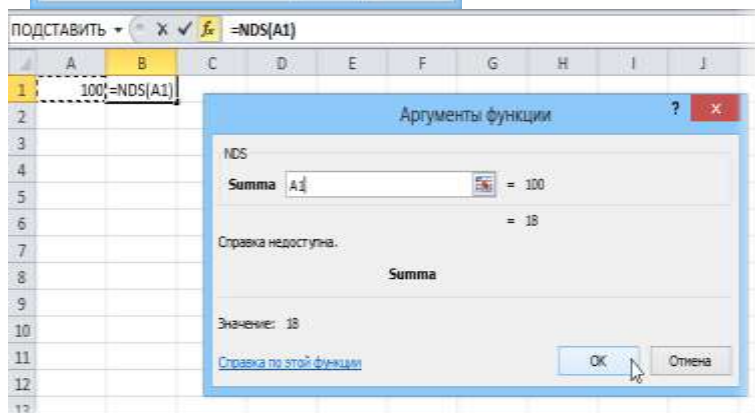


```
Function NDS (Summa)
    NDS = Summa * 0.18
End Function
```

Обратите внимание, что в отличие от макросов функции имеют заголовок **Function** вместо **Sub** и непустой список аргументов (в нашем случае это *Summa*). После ввода кода наша функция становится доступна в обычном окне Мастера функций (**Вставка - Функция**) в категории **Определенные пользователем (User Defined)**:



После выбора функции выделяем ячейки с аргументами (с суммой, для которой надо посчитать НДС) как в случае с обычной функцией:



6 Описание метода для решения нелинейных уравнений

Для решения нелинейных уравнений в Excel используется **итерационный метод**. Метод доступен в виде встроенного модуля (**Данные - Анализ «если что» -Подбор параметра...**) (Рис 6.1).

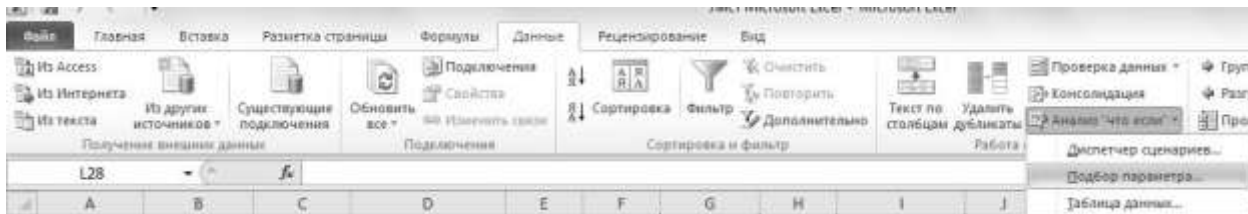


Рисунок 6.1- Лента инструментов вкладки Данные

Пример задачи нахождения корней уравнения для функции $x^2 - \sin 5x = 0$, принадлежащих отрезку $[-0,1; 0,6]$.

- ✚ Заполните столбцы значениями аргумента (X) и функции на указанном диапазоне (рис.6.2).
- ✚ Постройте точечную диаграмму (**Вставка – Диаграмма - Точечная**), которая необходима для уточнения количества и мест расположения решений на указанном отрезке. В случаях, если функция имеет ограниченную область определения или решений несколько, определитесь с «точками начала итераций» (точками начала перебора, к которым искомые корни расположены достаточно близко).

Точка (0,0) определена точно, а вот второй корень уравнения, располагающийся между значениями аргумента 0,5 и 0,6, требует уточнения. Точками начала итераций будут $x=0,5$ и $x=0,6$.

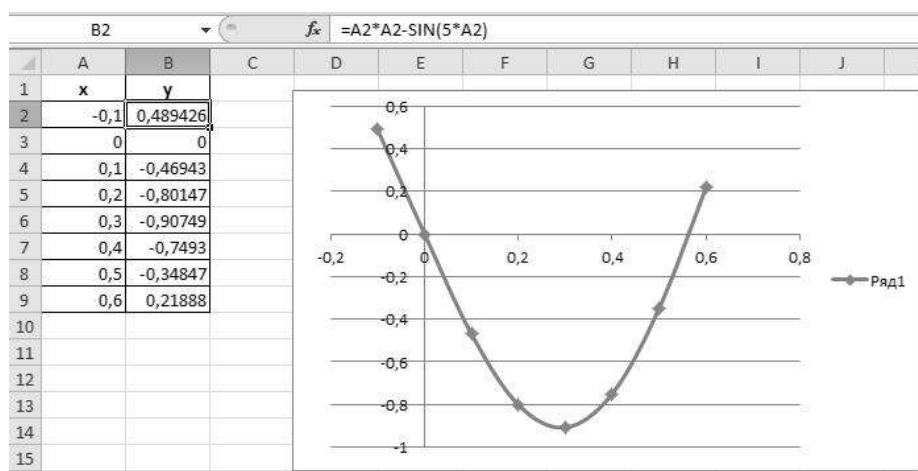


Рисунок 6.2 - График функции $x^2 - \sin 5x = 0$

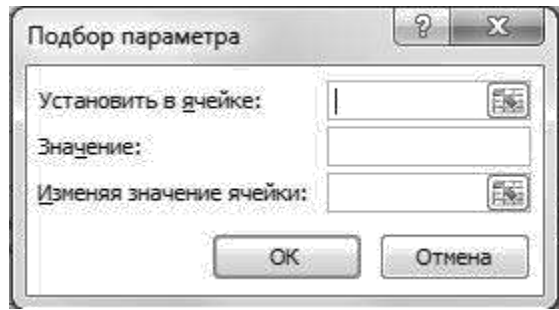


Рисунок 6.3 - Мастер Подбор параметра

Внесите в свободные ячейки электронной таблицы значения «точек начала итераций». Рядом скопируйте формулы для расчета значений функции в данных точках.

Обратитесь к мастеру **«Подбор параметра»** (**Данные - Анализ «если что» - Подбор параметра...**) и введите необходимую информацию:

- ✚ Адрес ячейки, в которой рассчитывается значение функции
- ✚ Требуемое значение функции (в нашем случае 0)
- ✚ Адрес ячейки, в которой уточняется значение корня уравнения

Подобным образом последовательно уточните все корни уравнения, соответствующие условию задачи.

X	Y
0	0
0,563671	8,43E-05

Для уравнений, у которых не указан интервал, подберите его самостоятельно, обращая внимание на точки, где функция меняет свой знак.