

| | |
|---|-----|
| Глава 24. Дополнительная информация | 573 |
| 24.1. Тестирование и отладка | 574 |
| 24.2. Работа с редактором | 578 |



Глава 24. Дополнительная информация

Я уже рассказал достаточно много и до сих пор не упомянул ни слова об оболочке Delphi. В большинстве книг это первое, с чего начинают обучение программированию, что на мой взгляд грубейшая ошибка. Нет смысла обучать человека тому, чего он абсолютно не собирается применять, потому что не знает зачем и когда.

Сначала нужно заинтересовать человека, что я и пытался сделать на протяжении всей книги. Теперь же, когда у тебя достаточно знаний для написания собственных проектов, пора показать тебе, как работать с оболочкой Delphi, как тестировать и отлаживать написанные тобой приложения. В программах регулярно возникают ошибки даже у именитых профессионалов и для выявления ошибок необходимо много сил и терпения. Я не буду говорить, что Delphi упрощает отладку, потому что это и так понятно. Я лучше всё это покажу на практике.





24.1. Тестирование и отладка

Для начала мы разберёмся с отладкой твоих программ. Отладка – пошаговое выполнение операций программы. В этом режиме, каждая строчка кода выполняется по команде и сразу после её выполнения Delphi останавливает её работу и ждёт следующей команды. Пока программа остановлена, ты можешь просмотреть значения всех переменных на данном этапе и даже изменить их.

Точка прерывания отладки - строка кода, на которой программа должна остановить своё выполнение и перейти в Delphi для продолжения выполнения в пошаговом режиме.

Для того, чтобы поставить точку прерывания, нужно выделить строку и нажать F5. Эта строка должна окраситься в красный цвет. Если она сразу или после компиляции оказалась другого цвета, то на этой строке программа не может останавливаться. На рисунке 24.1.1 ты можешь видеть пример строки, на которой установлена точка прерывания.

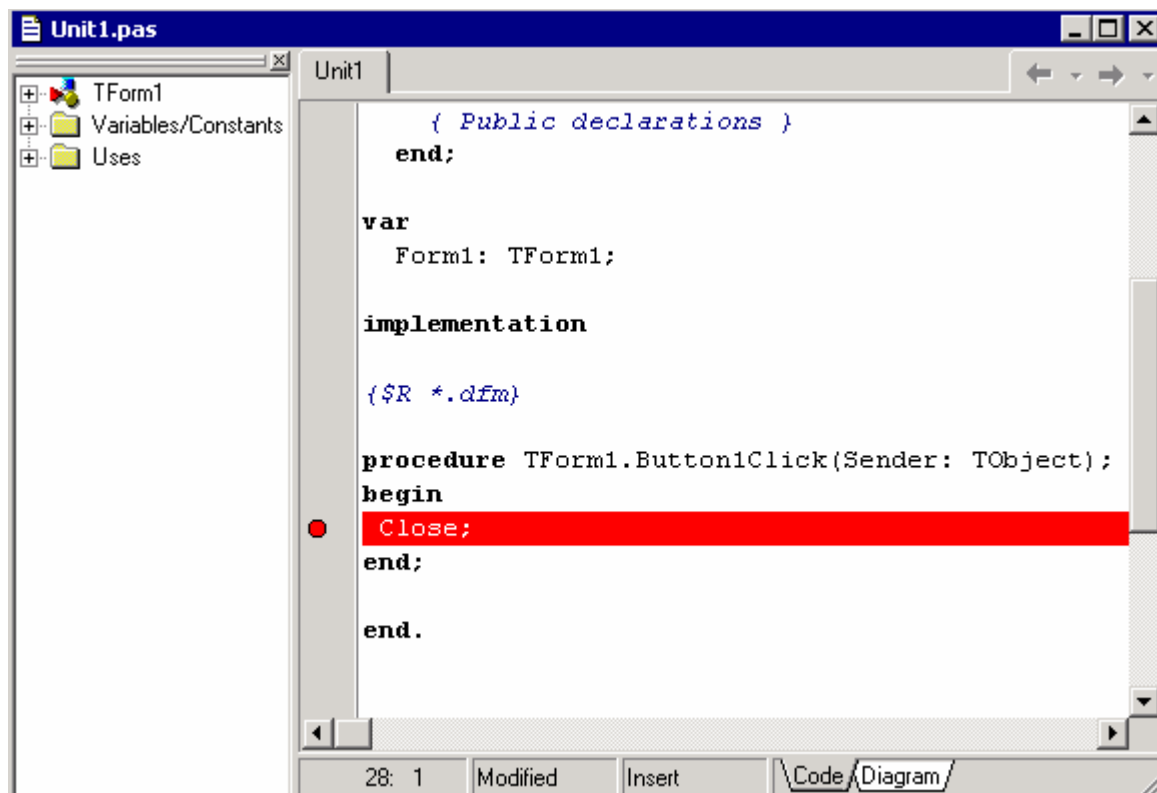


Рис 24.1.1. Точка прерывания.

Если слева от строки стоит синяя точка, то эта строка 100% может стать точкой прерывания. Если такой точки нет, то и прерывания не может быть. Эти точки видны не всегда. Они могут пропадать и появляться только после очередной компиляции программы.

Давай напишем маленький пример и попробуем разобраться с отладкой на нём. Создай новое приложение и установи на форму одну кнопку. По её нажатию напиши следующий код:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
i, j:Integer;
begin
i:=10;
j:=20;
i:=i+j;
end;
```

Теперь откомпилируй программу (Ctrl+F9). После компиляции ты увидишь три сообщения типа *Value assigned to 'i' never used* (рисунок 24.1.2). Эти сообщения говорят о том, что значения присвоенные указанным переменным не используются. В таких случаях Delphi оптимизирует код программы и раз значения не используются, то и незачем компилировать этот код.

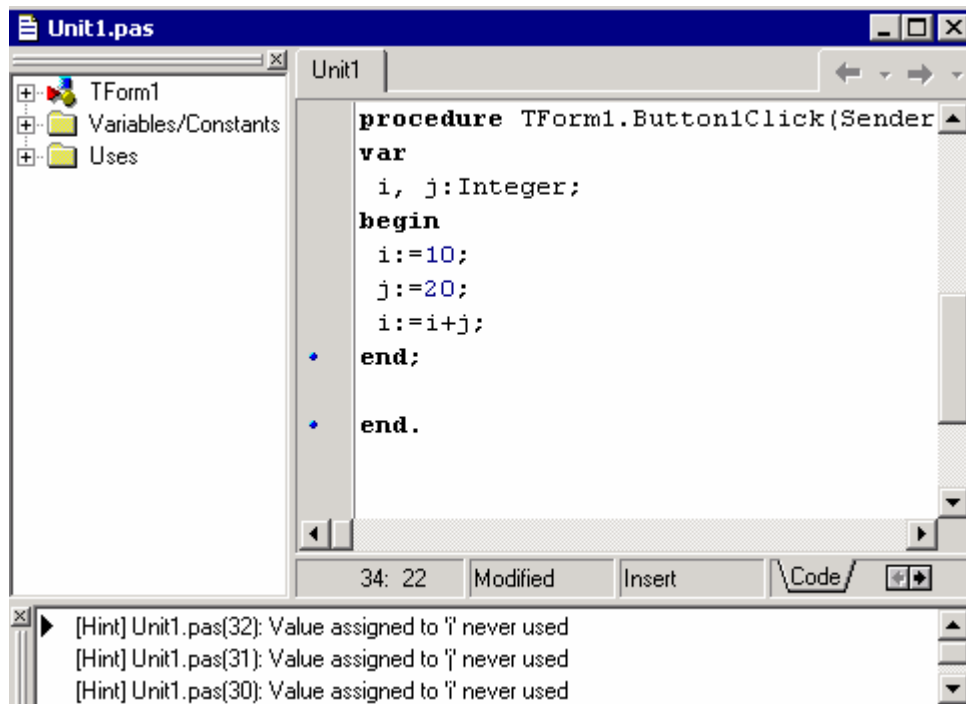


Рис 24.1.2. Список сообщений внизу окна редактора кода.

Обрати внимание, что слева от наших строк кода не появились синие точки. Это значит, что мы не сможем поставить на них прерывания. Точнее сказать сможем, но они не будут работать. Это связано с тем, что Delphi оптимизировал эти строки, потому что они явно не влияют на ход программы. Мы производим расчёты, но они никуда не выводятся. Так что по синим точкам ты можешь проверить, какой код был оптимизирован.

Давай откорректируем наш код до следующего вида:

```
procedure TForm1.Button1Click(Sender: TObject);
var
i, j:Integer;
begin
i:=10;
j:=20;
i:=i+j;
if i>0 then exit;
end;
```

Здесь я всего лишь добавил в самом конце процедуры проверку переменной *i*. Если она больше нуля, то произойдёт выход процедуры. Если так подумать, то этот код тоже не влияет на ход работы программы, потому что даже если *i* будет меньше нуля, процедура всё равно заканчивается и произойдёт выход. Но если теперь откомпилировать программу, то сообщений не будет и слева от строк кода появятся синие точки. Вот такой код уже имеет смысл отлаживать.

Запусти программу и как только ты нажмёшь на кнопку, то выполнение остановится и управление получит Delphi. Чтобы продолжить выполнения программы до следующей строки можно нажать F8.

Если выделенная строка – твоя процедура или функция, то можно нажать F7, чтобы отладчик перешёл внутрь этой процедуры и продолжил её выполнение построчно. Если ты нажмёшь F8, то отладчик выполнит процедуру без тебя и перейдёт дальше. В этом случае ты не сможешь посмотреть, что происходит внутри процедуры.

Если ты хочешь, чтобы отладочный режим закончился и программа продолжила выполняться самостоятельно, то нажми F9. После этого программа продолжит своё выполнение с последней остановленной точки. Если ты хочешь совсем остановить работу программы, то нажми Ctrl+F2 или выбери из меню *Run* пункт *Program Reset* и программа будет выгружена из памяти.

Если ты хочешь посмотреть, какое значение хранится в переменной, то надо её выделить и нажать Ctrl+F7 или выбрать *Evaluate/Modify* из меню *Run*. Перед тобой откроется окно, как на рисунке 24.1.2. Если ты видишь, что значение неправильное и хочешь его изменить, то в строке *New value* нужно ввести нужное значение и нажать кнопку *Modify*. Чтобы пересчитать значение переменной нужно нажать кнопку *Evaluate*.

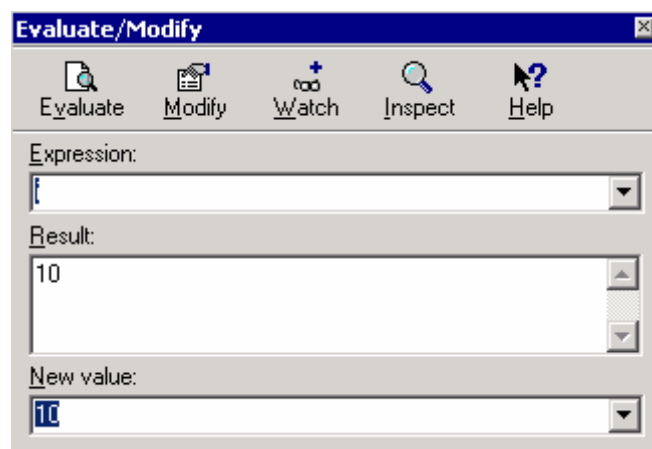


Рисунок 24.1.2 Окно просмотра значений переменной

В окне *Evaluate/Modify* можно вводить не только переменные, но и целые выражения. Например, поставь точку прерывания в примере выше на первой строчке кода процедуры *Button1Click*. Теперь запусти программу и нажми кнопку. Выполнение программы должно прекратиться и управление перейдёт в Delphi. Нажми F8, чтобы выполнить первую строку кода и перейти на вторую. Текущая строка должна быть выделена синим цветом. Теперь нажми Ctrl+F7 и в строке *Expression* окна *Evaluate/Modify* введи следующее выражение: *i+10*. Нажми *Enter* или кнопку *Evaluate*, чтобы просчитать это выражение. В строке *Result* должно появиться значение 20, потому что переменная *i* равна 10 плюс ещё 10 получаем результат 20.

Введи в строку *Expression* просто переменную *i*. Перечитай её значение нажатием кнопки *Evaluate*. Теперь с строке *New Value* введи значение 15 и нажми кнопку *Modify*. Теперь переменная *i* должна равняться 15. Проверь это нажатием кнопки *Evaluate*.

Попробуем ввести в строке *Expression* выражение ***i=10***. Здесь используется операция сравнения переменной *i* и числа 10. Если переменная равна 10, то мы должны увидеть в строке *Result* значение *true*, иначе *false*.

Теперь надо сказать пару слов о видимости переменных. Останови работу программы. Для этого можешь нажать Ctrl+F2, чтобы программа выгрузилась без сохранения данных или продолжи программу нажатием F9 для окончания отладочного режима и закрой окно программы. Снова запусти программу нажатием F9 из Delphi. Нажми кнопку в окне программы и снова Delphi прервёт выполнение и перейдёт в отладочный режим. Выдели переменную *i* и нажми Ctrl+F7, чтобы увидеть значение переменной. Вместо её значения, в строке *Result* ты увидишь следующий текст: *Variable 'i' inaccessible here due to optimization*. Этот текст гласит примерно следующее (в моём вольном переводе): *переменная i недоступна в этом месте потому что оптимизирована*. В данном случае программа остановила своё выполнение на строке:

i:=10;

здесь мы присваиваем переменной *i* значение 10, до этого кода переменная не имеет значения, поэтому мы и увидели такую надпись. Попробуй выполнить эту строку нажатием F8 и снова посмотреть значение переменной *i*. Теперь значение равно десяти.

Попробуй выполнить процедуру до самого конца. Остановись на следующей строчке:

if i>0 then exit

Теперь попробуй посмотреть значение переменной *j*. Ты снова должен увидеть сообщение о том, что переменная оптимизирована и не имеет значения. Это связано с тем, что начиная с текущей строчки кода и до конца процедуры уже нет обращений к значению переменной. Значит это значение не нужно и Delphi его снова оптимизировала.

Есть ещё один способ увидеть текущее значение переменной. Ты должен также выделить эту переменную и нажать Ctrl+F5 или выбрать *Add Watch* из меню *Run*. Переменная будет добавлена в специальное окно *Watch*, в котором будет постоянно отображаться её текущее значение (рисунок 24.1.3).



Рисунок 24.1.3. Окно *Watch*

Если ты делаешь всё, как я говорю, то у тебя программа должна быть остановлена на последней строке кода процедуры. Нажми F9 чтобы продолжить её выполнение, и ты увидишь окно своей программы. Снова нажми кнопку и программа снова остановит выполнение на первой строчке процедуры (если ты не снял точку останова). Добавь переменную *i* в окно *Watch*.

Попробуй дважды щёлкнуть по строке с переменной *i* в окне *Watch* и перед тобой откроется окно параметров просмотра (рис 24.1.4). В этом окне ты можешь сделать достаточно много настроек, но самое интересное – это в центре окна. Здесь находится большое количество элементов *RadioButton*. Выделяя один из них ты выбираешь тип твоей переменной. В зависимости от выбранного типа ты будешь по разному видеть её в окне *Watch*. Но чаще всего ты не будешь изменять эти настройки и достаточно значений по умолчанию.

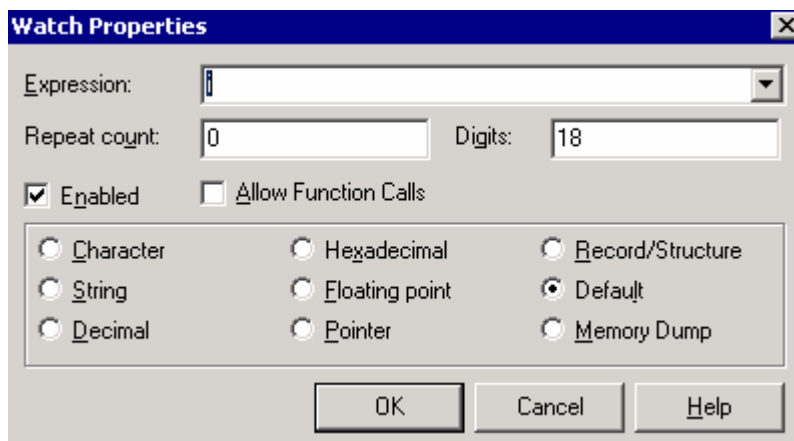



Рисунок 24.1.4. Окно *Watch*

Теперь попробуй расположить окно *Watch* так, чтобы оно всегда было видно и не мешало видеть код программы. Попробуй построчно выполнить код программы нажатием F8 и понаблюдать за изменением значения переменной *i*.

 На компакт диске, в директории \Примеры\Глава 24\Отладка ты можешь увидеть пример использованной здесь программы.

24.2. Работа с редактором

Теперь я хочу познакомить тебя с некоторыми приёмами по работе с редактором кода Delphi. В этой части ты узнаешь, как работать с закладками, как быстро создавать переменные, процедуры и функции и как искать нужный код. Если у тебя маленькая программа и модуль состоит из нескольких строк, то тут у тебя не будет проблем, потому что найти что-то нужное не так уж и сложно. А что если проект большой и модуль из 1000 строк? Вот тут возникает множество проблем, с которыми надо бороться.

Закладки

Встань на какую-нибудь строку текста (выделять не надо) и нажми Ctrl+Shift+ любая цифра. Эти клавиши поставят закладку на строке. Теперь перейди в другое место и нажми Ctrl и ту же цифру. После этого ты вернёшься в строку, где была закладка.

Когда ты устанавливаешь закладку на строку, то слева от строки кода появляется изображение кубика с цифрой внутри. Цифра указывает на номер закладки и именно эта цифра в сочетании с Ctrl моментально перенесёт тебя на выделенную строку. Пример редактора кода с закладкой ты можешь увидеть на рисунке 24.2.1.

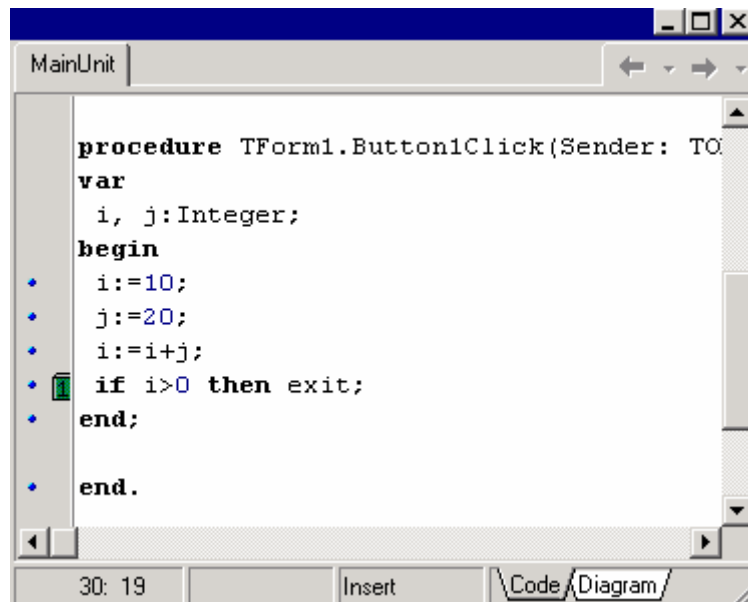


Рисунок 24.2.1. Окно редактора кода с закладкой

Закладки можно устанавливать и из контекстного меню. Щёлкни правой кнопкой мыши в редакторе кода и в появившемся меню наведи на пункт *Toggle Bookmarks*. Откроется подменю в котором ты можешь увидеть изображения кубиков с номерами и подписи. Выбирая любой из пунктов ты можешь установить соответствующую закладку.

Если ты установил закладку с номером 1 на одну строку и потом устанавливаешь на новую строку закладку с этим же номером, то из предыдущего места расположения закладка снимается и переносится в новое место. Чтобы просто снять закладку достаточно перейти на её строку и повторно установить там закладку с тем же номером. В этом случае закладка просто исчезнет.

Копирование строк

Встань в начало строки текста. Выдели строку с помощью нажатия Shift+стрелка вниз. Это выделит всю строку. Теперь удерживая Ctrl нажми K, а затем C (буквы латинские). Этот маленький трюк скопирует выделенную строку без использования буфера обмена (clipboard) и установит её чуть ниже копируемой

Code Explorer

Теперь посмотрим, как быстро создавать переменные. В окне кода слева, находится вытянутое окно *Code Explorer* в котором ты можешь видеть дерево из трёх пунктов:

1. *TForm1* – в этой ветке находится описание всех компонентов стоящих на твоей форме.
2. *Variables/Constants* – здесь хранятся переменные и константы.
3. *Uses* – здесь храниться список модулей, подключённых к программе.

Давай попробуем создать новую глобальную переменную в разделе **var**. Для этого щёлкни правой кнопкой по пункту *Variables/Constants* и выбери в появившемся меню пункт *New*. В дереве будет создан пункт, в котором нужно ввести следующий текст:

H: Integer

Таким образом мы создали переменную *H* в разделе **var** и при этом не имеет значения, в какой строчке кода мы находимся. На рисунке 24.2.2 ты можешь увидеть создание нового элемента.

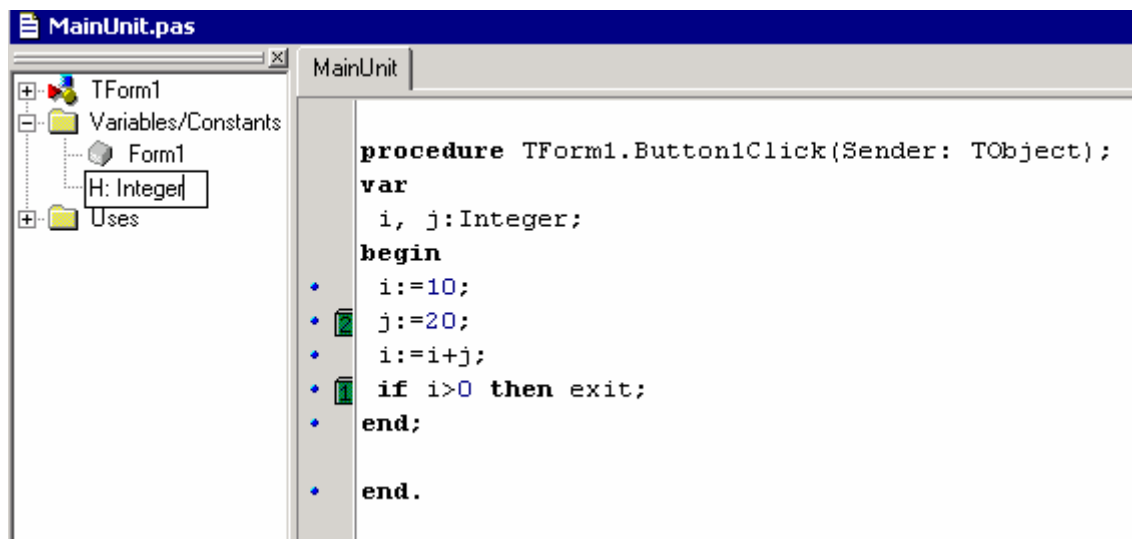


Рисунок 24.2.1. Окно редактора кода с закладкой

Такой приём очень удобен, когда у тебя очень большой модуль и не хочется переходить в самое начало, а потом возвращаться назад.

Точно так же можно создавать не только переменные, но и процедуры и функции для объекта главной формы. Для этого нужно щёлкать правой кнопкой по ветке *TForm1* (имя объекта твоего окна).