

Практическая работа № 1

Отладка приложения в Visual Studio 2010

Существуют два способа запуска режима отладки в *Visual Studio*, первый - *запуск* из меню *Debug* и второй - *запуск* с помощью соответствующей панели инструментов. Оба способа - предоставляют *доступ* к запуску сеансов отладки, пошаговому прохождению кода, управлению точками останова, а также и ко многим функциональным возможностям отладки в *Visual Studio*. Имеются два состояния меню отладки (*Debug*):

- Состояние покоя (неактивное);
- Режим отладки.

В состоянии покоя меню *Debug* (Рис. 1) предоставляет возможности для запуска сеанса отладки, прикрепления к выполняющемуся процессу и для доступа к некоторым из отладочных окон. В Таблице 1 перечислены все функциональные возможности, имеющиеся в меню *Debug* в состоянии покоя.

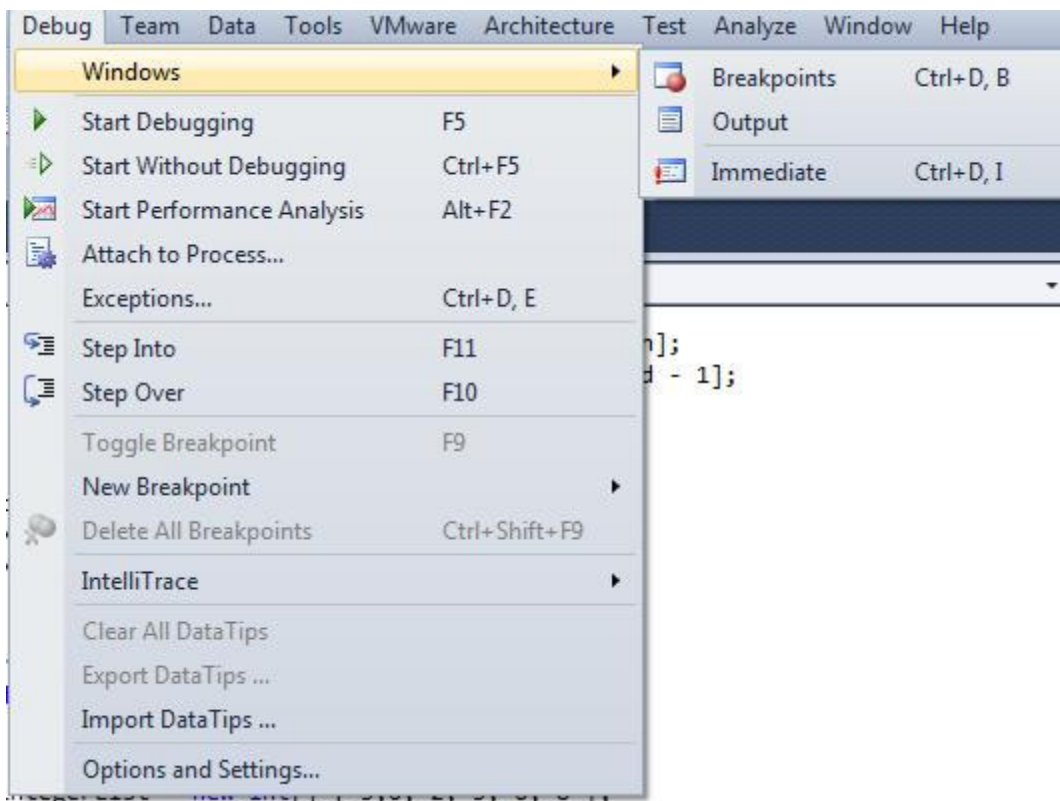


Рис. 1. Меню Debug в Visual Studio 2010 в режиме покоя

Таблица 1. Элементы меню Debug в состоянии покоя	
Элемент меню	Описание
Windows Breakpoints	Открывает в интегрированной среде окно Breakpoints, которое дает доступ ко всем точкам останова данного решения.
Windows Output	Показывает в интегрированной среде окно Output. Окно Output - это бегущий журнал множества сообщений, выдаваемых интегрированной средой, компилятором и отладчиком. Поэтому эта информация относится не только к сеансу отладки

Windows Immediate	Открывает в интегрированной среде окно Immediate, которое позволяет выполнять команды.
Start Debugging	Запускает приложение в режиме отладки
Start Without Debugging	Запускает приложение без подключения отладчика к выполняющемуся процессу. В этом режиме разработчик видит то, что увидит пользователь (вместо того, чтобы выходить в интегрированную среду при ошибках и в точках останова)
Attach to Process	Позволяет прикрепить отладчик к выполняющемуся процессу (исполняемому файлу). Например, если запущено приложение без отладки, то можете потом прикрепить к этому выполняющемуся процессу и начать отладку
Exceptions	Открывает диалоговое окно Exceptions, которое позволяет выбрать способ останова отладчика для каждого исключительного состояния
Step Into	Запускает приложение в режиме отладки. Для большинства проектов выбор команды Step Into означает вызов отладчика на первой выполняемой строке приложения. Таким образом, можно войти в приложение с первой строки
Step Over	Когда вы не находитесь в сеансе отладки, то команда Step Over просто запускает приложение точно так же, как это сделала бы кнопка Run
Toggle Breakpoint	Включает или выключает точку останова на текущей (активной) строке кода текстового редактора. Эта опция неактивна, если в интегрированной среде нет активного кодового окна
New Breakpoint Break at Function	Активирует диалоговое окно New Breakpoint позволяющее указать имя функции, для которой необходимо создать точку останова.
New Breakpoint New Data Breakpoint	Эта опция доступна только для приложений C++, позволяет определить точку останова, которая выходит в интегрированную среду тогда, когда изменяется значение по определенному адресу в памяти
Delete All Breakpoints	Удаляет все точки останова из текущего решения
Disable All Breakpoints	Деактивирует (без удаления) все точки останова текущего решения

Когда отладчик запущен, то состояние *меню Debug* изменяется, в нем становятся активными несколько дополнительных опции. Эти опции включают: функции перемещения по коду, перезапуск сеанса и *доступ* к дополнительным окнам отладки и т.д. На *Рис. 2* показано *меню Debug* во время сеанса отладки. В *Таблице 2* представлены элементы *меню Debug* в состоянии отладки.

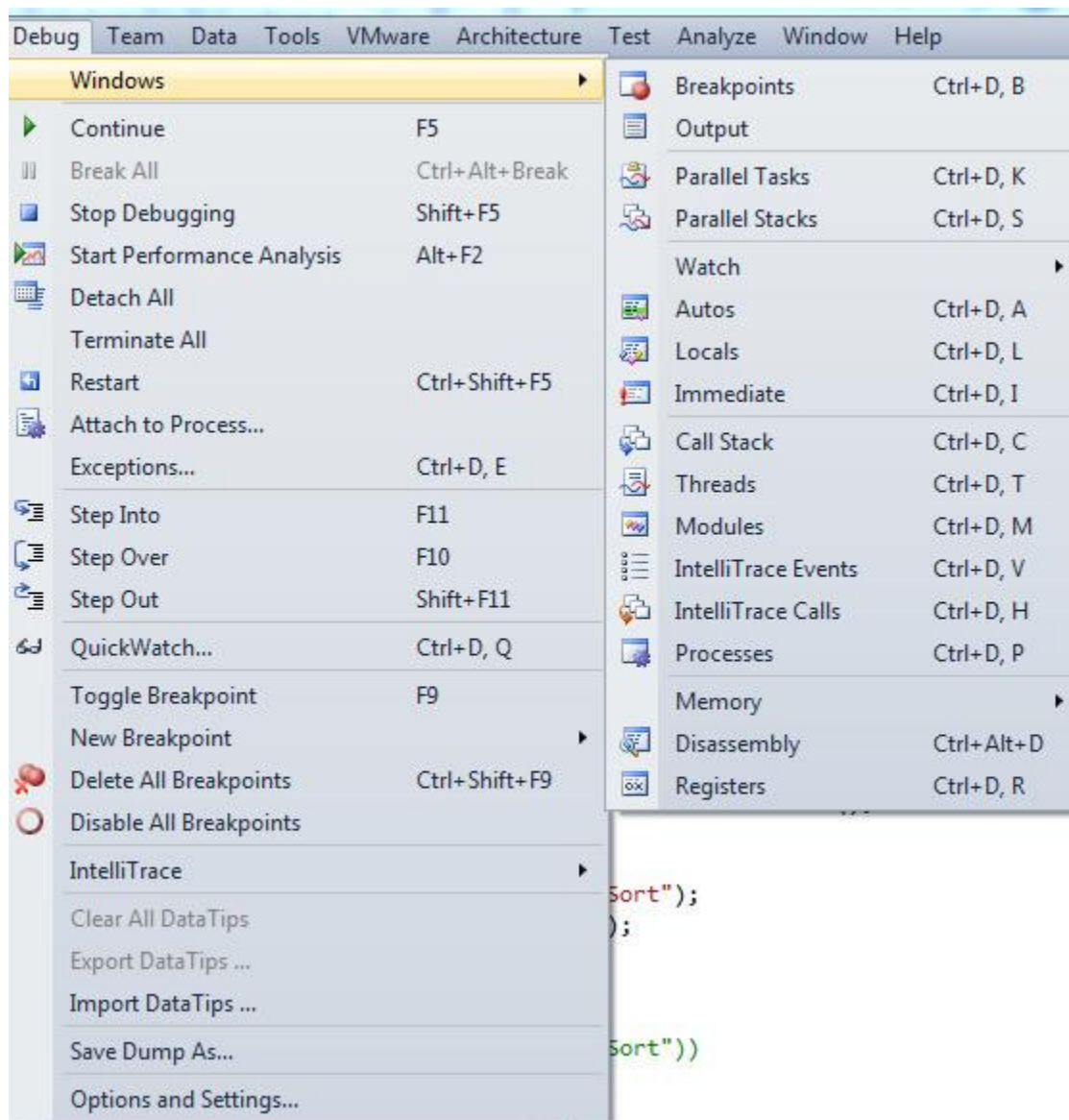


Рис. 2. Меню Debug в Visual Studio 2010 в режиме отладки

Таблица 2. Элементы меню Debug в режиме отладки

Элемент меню	Описание
Windows Breakpoints	Позволяет открыть окно Breakpoints во время сеанса отладки.
Windows Output	Открывает окно Output во время активного сеанса отладки для того, чтобы можно было читать выходные сообщения, выдаваемые компилятором и отладчиком.
Windows Parallel Task	Одно из нововведений в Visual Studio 2010. Подробнее о данном нововведении в подразделе лекции "Окно Parallel Task" .
Windows Parallel Stacks	Одно из нововведений в Visual Studio 2010. Подробнее о данном нововведении в подразделе лекции "Окно Parallel Stacks".
Windows Watch	Открывает одно из нескольких окон контрольных значений интегрированной среды. Окна контрольных значений представляют элементы и выражения, за которыми вы наблюдаете в течение сеанса отладки.

Windows Autos	Открывает окно Autos. Это окно показывает переменные (и их значения) в текущей и предыдущей строках кода.
Windows Locals	Открывает в интегрированной среде окно Locals, которое показывает переменные в локальной области действия (функции).
Windows Immediate	Открывает окно Immediate, в котором вы можете выполнить команду.
Windows Call Stack	Открывает список функций, которые имеются в стеке. Также указывает текущий кадр стека (функцию). Выделенный элемент - это то, что определяет содержимое окон Locals. Autos и окон контрольных значений.
Windows Threads	Показывает в интегрированной среде окно Threads. Здесь можно просматривать потоки отлаживаемого приложения и управлять ими.
Windows Modules	Показывает в интегрированной среде окно Modules. Это окно дает список dll и exe-файлов, используемых приложением.
Windows IntelliTrace Events	Одно из нововведений в Visual Studio 2010. По умолчанию IntelliTrace собирает информацию для выбранных событий IntelliTrace. Когда приложение входит в режим приостановки выполнения, можно воспользоваться представлением IntelliTrace Events, чтобы просмотреть собранные события диагностики.
Windows IntelliTrace Calls	Одно из нововведений в Visual Studio 2010. Когда IntelliTrace выполняет сбор сведений о вызовах, эти сведения можно просматривать в окне IntelliTrace Calls.
Windows Processes	Показывает в интегрированной среде окно Processes. Это окно отображает список процессов, к которым прикреплен сеанс отладки.
Windows Memory	Открывает окно Memory для просмотра используемой приложением памяти. Это работает только при включенной (в диалоговом окне Options) отладке на уровне адресов.
Windows Disassembly	Открывает окно Disassembly. Это окно показывает ассемблерный код, соответствующий командам компилятора. Данная функция работает только при включенной (в диалоговом окне Options) отладке на уровне адресов.
Windows Registers	Открывает окно Registers, чтобы вы могли видеть изменения значений регистров при прохождении по коду. Данная функция работает только при включенной (в диалоговом окне Options) отладке на уровне адресов.
Continue	Продолжает выполнение приложения после выхода в интегрированную среду разработки. Приложение продолжает выполняться с активной строки кода.
Break All	Позволяет прервать приложение вручную (без использования точки останова) во время сеанса отладки. Приложение прервется на следующей исполняемой строке. Эта возможность полезна в том случае, когда нужно получить доступ к отладочной информации.
Stop Debugging	Останавливает режим отладки. Прерывает также и отлаживаемый процесс.
Detach All	Открепляет отладчик от выполняющегося процесса. Это позволяет приложению продолжать выполнение после того, как отладчик свою работу закончил.
Terminate All	Останавливает отладку и прекращает все процессы, к которым вы прикреплены.
Restart	Останавливает и перезапускает его сеанс отладки. Аналогично последовательному нажатию кнопок Stop Debugging и Start Debugging.
Attach to Process	Позволяет прикрепить активный сеанс отладки к одному дополнительному процессу.
Exceptions	Активирует диалог Exceptions, который позволяет управлять выходом в IDE по конкретным типам исключительных состояний .NET Framework и других библиотек.

Step Into	Приводит к продвижению отладчика на одну строку.
Step Over	Работает точно так же, как Step Into, но с одной важной разницей: если используете пропуск функции по Step Over, то строка вызова функции будет выполнена (и функция тоже), и отладчик установит следующую строку за вызовом функции в качестве следующей отлаживаемой строки.
Step Out	Указывает отладчику выполнить текущую функцию, а затем выйти назад в отладчик (после выполнения функции). Эта функция полезна тогда, когда осуществляется вход в функцию, а затем нужно, чтобы эта функция выполнялась и вернула в отладчик после ее завершения.
QuickWatch	Активирует окно QuickWatch. Это окно отображает значение одной, конкретной переменной (или выражения).
Toggle Breakpoint	Включает и выключает активную точку останова.
New Breakpoint	Активирует диалоговое окно New Breakpoint.
Delete All Break - 'points	Удаляет все точки останова в вашем решении

Установка точки останова

Точка останова (breakpoint) - это сигнал, который указывает отладчику временно остановить выполнение программы в определенной точке. Приостановка выполнения программы в точке останова называется режимом приостановки. Вход в режим приостановки выполнения не приводит к прекращению или завершению работы программы, поэтому выполнение программы может быть продолжено в любое время. В Visual Studio можно помещать на любую строку кода, которая выполняется. Существуют три способа расстановки точек останова в Visual Studio 2010:

- С помощью клавиши F9;
- Через пункт меню Debug - Toggle Breakpoint;
- И самый простой способ - это щелкнуть дважды левой кнопкой мыши на нужной строке, в окне редактора кода внутри затененной области вдоль левого края окна документа.

Все эти способы приводят к размещению в конкретной строке точки останова, которая вызывает прерывание процесса выполнения и передачу управления отладчику. Как и в предыдущих версиях Visual Studio, точка останова обозначается большим кружком слева от соответствующей строки в окне редактора кода.

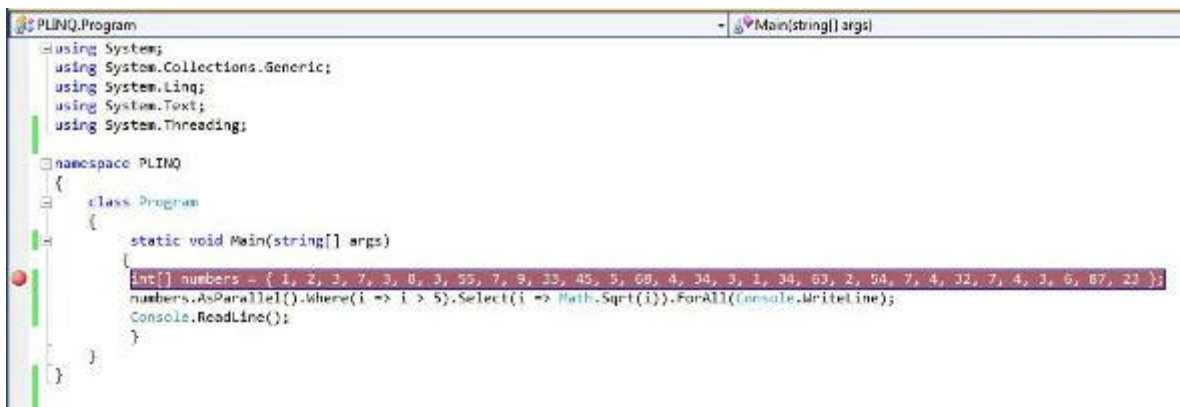


Рис. 3. Установка точки останова в Visual Studio 2010

Запуск отладчика

Для того что бы запустить режим отладки можно использовать несколько способов.

- Способ первый. С помощью специальной панели инструментов, как показано на Рис. 4;



Рис. 4. Запуск режима отладки с панели инструментов

- Способ второй. Из меню Debug с помощью пункта Start Debugging (Рис. 5);

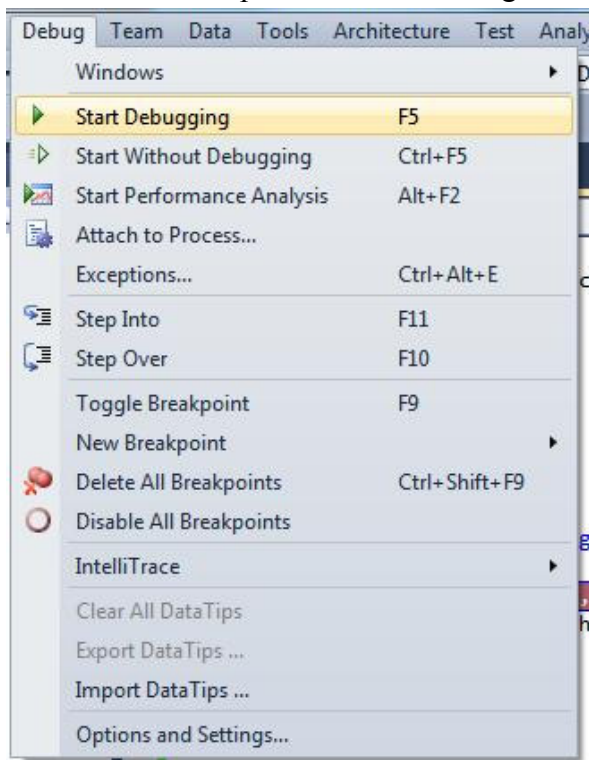


Рис. 5. Запуск режима отладки из меню Debug

- Способ третий. С помощью клавиши F5.

После запуска режима отладки, появится курсор отладки, который остановится напротив первой точки останова (Рис. 6).

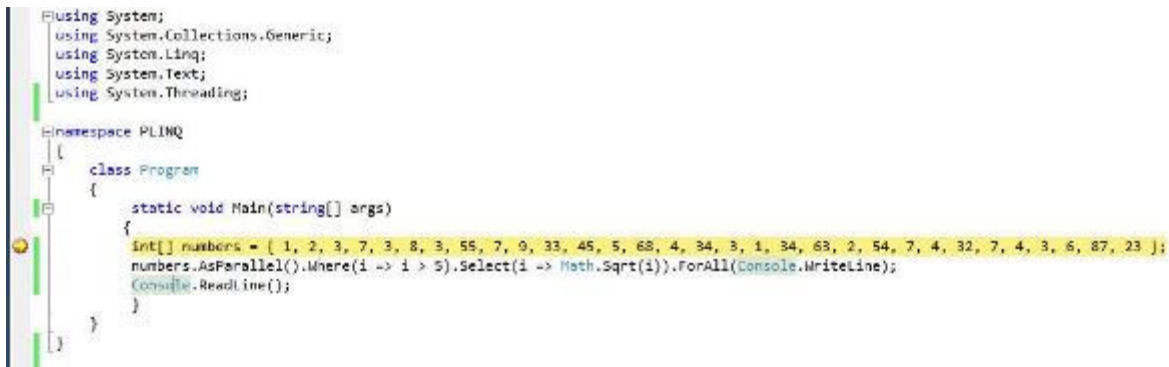


Рис. 6. Курсор отладки в Visual Studio

Пошаговое выполнение программы

Для пошаговой отладки используют специальную панель инструментов (Рис. 7).



Рис. 7. Специальная панель инструментов отладки

Или набор горячих клавиш (которые дублируют работу специальной панели инструментов), список которых представлен ниже:

- Continue (F5) - продолжить выполнение программы.
- Stop debugging (Shift+F5) - остановить отладку. При этом остановится и выполнение программы.
- Restart (Ctrl+Shift+F5) - перезапустить программу. Выполнение программы будет прервано и запущится заново;
- Show Next Statement (Alt + Num *) - показать следующий оператор, т.е. переместить курсор редактора кода в курсор пошагового выполнения;
- Step Into (F11) - выполнить очередной оператор. Если это метод, то перейти в начало этого метода, чтобы начать отладку;
- Step Over (F10) - выполнить очередной оператор. Если это метод, то он будет полностью выполнен, т.е. курсор выполнения не будет входить внутрь метода;
- Step out (Shift + F11) - выйти из метода.

При выполнении пошаговой отладки, разработчик может использовать следующие окна для просмотра значения переменных или если приложение многопоточное, то просматривать состояние потоков или переключаться между ними:

- Autos;
- Locals;
- Watch;
- Immediate;
- Threads;

- Parallel Task;
- Parallel Stacks.

Окно Autos

Окно Autos (Рис. 8) используется для того чтобы, просматривать значения, связанные с той строкой кода, на которой находится курсор отладки. Это окно отображает значения всех переменных и выражений, имеющих в текущей выполняющейся строке кода или в предыдущей строке кода. Содержит следующие столбцы:

- Name - название переменной;
- Value - значение переменной;
- Type - тип переменной.

Name	Value	Type
taskInsertionSort	Id = 1, Status = Created, Method = "Void <Main>b_00"	System.Threading.Tasks.Task
AsyncState	null	object
CancellationPending	false	bool
CreationOptions	None	System.Threading.Tasks.TaskCreationOptions
Exception	null	System.Exception
Id	1	int
Status	Created	System.Threading.Tasks.TaskStatus
Raw View		

Рис. 8. Окно Autos

Окно Locals

Окно Locals (Рис. 9) отображает все переменные и их значения для текущей области видимости отладчика, что дает представление обо всех переменных, которые используются в текущем выполняющемся методе. Переменные в этом окне автоматически настраиваются отладчиком. Данное окно содержит следующие столбцы:

- Name - название переменной;
- Value - значение переменной;
- Type - тип переменной.

Name	Value	Type
args	{string[]}	string[]
integerList	{int[]}	int[]
cancellationSource	{System.Threading.CancellationTokenSource}	System.Threading.CancellationTokenSource
insertionsort	{Sorts.SortResults}	Sorts.SortResults
bubblesort	{Sorts.SortResults}	Sorts.SortResults
taskInsertionSort	Id = 1, Status = Created, Method = "Void <Main>b_00"	System.Threading.Tasks.Task
taskBubbleSort	null	System.Threading.Tasks.Task
token	IsCancellationRequested = false	System.Threading.CancellationToken
insertionList	Count = 6	System.Collections.Generic.List<int>

Рис. 9. Окно Locals

Окно Watch

Окно Watch или окно контрольных значений (Рис. 10) - позволяет настраивать собственный список переменных и выражений, которые необходимо отслеживать. Всего доступно четыре окна Watch (Watch 1, Watch 2, Watch 3 и Watch 4), что позволяет выделить в четыре списка переменные и выражения, данную возможность удобно использовать в том случае, если каждый список относится к отдельной области видимости приложения. Переменные или выражение в окно Watch добавляются или из редактора кода, или из окна

QuickWatch. Если нужно добавить в окно Watch элемент из редактора кода, то нужно выделить нужную переменную или выражение, щелкнуть по ней правой кнопкой мыши и выбрать пункт Add Watch. Также можно перетаскивать, с помощью мыши, выделенный элемент в окно Watch. Данное окно содержит следующие столбцы:

- Name - название переменной;
- Value - значение переменной;
- Type - тип переменной.

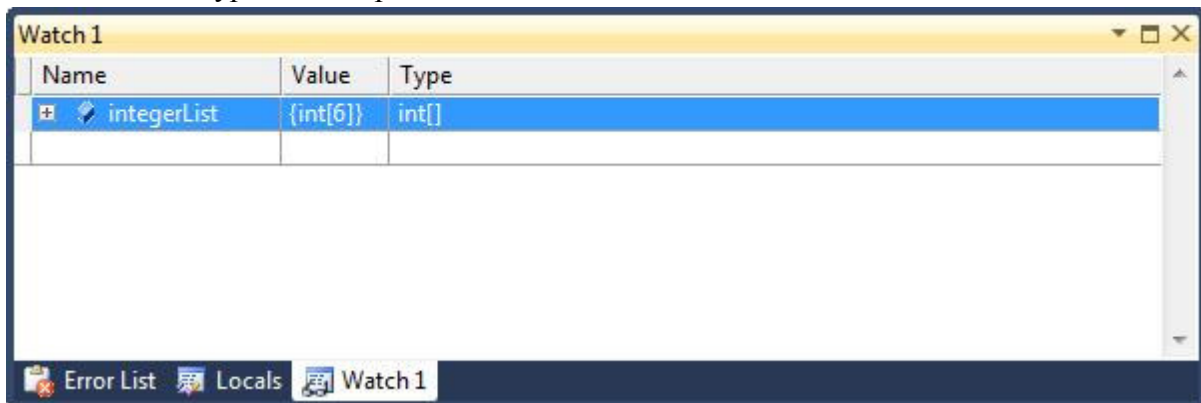


Рис. 10. Окно Watch

Окно Immediate

Окно Immediate или непосредственное выполнение (Рис. 11) - предназначено для ручного ввода и выполнения команд. Это окно появляется автоматически при прерывании работы программы в точках останова программы. Для выполнения команды или оператора необходимо написать команду и нажать клавишу <Enter>.

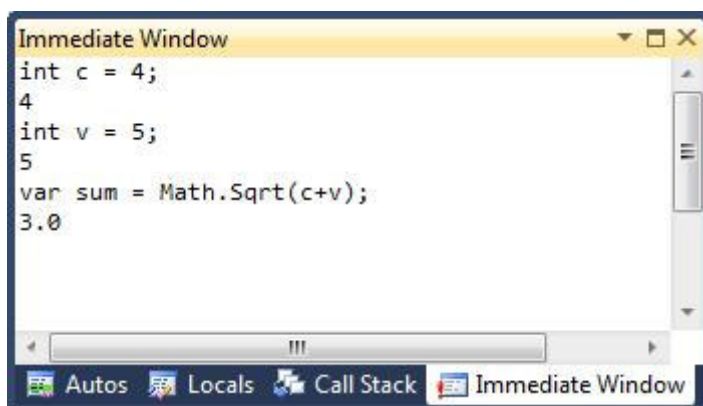


Рис. 11. Окно Immediate

Окно Threads

Окно Threads (Рис. 12) - позволяет просматривать и управлять всеми запущенными потокам на различных этапах отладки приложения.

Threads							
Search:		X Search Call Stack		Group by: Process Name			
	ID	Managed ID	Category	Name	Location	Priority	
Example 1.vshost.exe (id = 2092) : C:\parallel\Chapter 7\Example 1\Example 1\bin\Debug\Example 1.vshost.exe							
▼	5188	0	Worker Thread	<No Name>	<not available>	Highest	
▼	196	3	Worker Thread	<No Name>	<not available>	Normal	
▼	3784	6	Worker Thread	vshost.RunParkingWindow	▼ [Managed to Native Transition]	Normal	
▼	6940	7	Worker Thread	.NET SystemEvents	▼ [Managed to Native Transition]	Normal	
▼	4816	8	Main Thread	Main Thread	▼ Reporting_Example.Program.Main	Normal	
▼	2576	10	Worker Thread	Worker Thread	▼ Reporting_Example.XClass.ME	Normal	
▼	6668	13	Worker Thread	Worker Thread	▼ Reporting_Example.XClass.ME	Normal	
▼	6604	11	Worker Thread	Worker Thread	▼ Reporting_Example.XClass.MK	Normal	
▼	7080	9	Worker Thread	Worker Thread	▼ Reporting_Example.XClass.MM	Normal	
▼	6556	12	Worker Thread	Worker Thread	▼ Reporting_Example.XClass.MJ	Normal	
▼	7032	14	Worker Thread	<No Name>	<not available>	Normal	
▼	6056	15	Worker Thread	<No Name>	<not available>	Normal	
▼	6852	16	Worker Thread	<No Name>	<not available>	Normal	

Рис. 12. Окно Threads

По умолчанию в таблице перечисляются все потоки приложения, но можно фильтровать этот список, чтобы в нем показывались только нужные потоки. В каждом столбце содержится свой тип сведений. Сведения о столбцах окна Threads представлены в Табл. 3.

Таблица 3. Столбцы окна отладки Threads

Имя столбца	Описание
Флаги	Показывает, какие потоки помечены, и позволяет пометить потоки и снимать с них метки.
Значки	Желтая стрелка указывает активный поток. Контур стрелки указывает поток, где выполнение было передано в отладчик. Белая стрелка указывает прерванную задачу, т.е. задачу, которая была текущей во время вызова отладчика. Значок паузы указывает поток, замороженный пользователем.
ID	Столбец содержит идентификационные номера всех потоков.
Managed ID	В столбце содержатся управляемые идентификационные номера управляемых потоков.
Category	В данном столбце потоки классифицируются по категориям: потоки пользовательского интерфейса, обработчики удаленного вызова процедур (RPC) и рабочие потоки. Особая категория идентифицирует главный поток приложения.
Name	Столбец, в котором для каждого потока указывается имя, если оно имеется, или значение <No Name>.
Location	В данном столбце показывается, где поток выполняется. Можно развернуть это расположение, чтобы отобразить полный стек вызова для потока.

Priority	Столбец содержит приоритет потока, назначенный системой каждому потоку.
Affinity Mask	Дополнительный столбец, который обычно скрыт. В этом столбце показывается маска сходства процессора для каждого потока. В многопроцессорной системе маска сходства определяет, какой процессор, в каком потоке может работать.
Suspended	Столбец содержит счетчик приостановок. Этот счетчик определяет, может ли поток выполняться.

Окно Parallel Task

Окно Parallel Task (Рис. 13) или окно параллельных задач - выглядит как окно Threads (потоки), за исключением того, что отображает сведения о каждой задаче или объекте task_handle, вместо сведений о каждом потоке. Как и потоки, задачи представляют асинхронные операции, которые могут выполняться параллельно, однако несколько задач могут выполняться в одном потоке.

Parallel Tasks

	ID	Status	Location	Task	Parent	Thread Assignment
	1	Waiting	S.M		A(1)	4924 (Worker Thread)
	2	Running	S.R		A(2)	3412 (Worker Thread)
	3	Waiting-deadlocked	S.J		A(3)	2064 (Worker Thread)
	4	Waiting-deadlocked	S.R		A(4)	1516 (Worker Thread)
	6	Scheduled		2	A(6)	
	7	Scheduled		2	A(7)	
	8	Scheduled		2	A(8)	
	9	Scheduled		2	A(9)	
	10	Scheduled		2	A("10")	

Рис. 13. Окно Parallel Task

Сведения о столбцах окна Parallel Tasks представлены в Табл. 4.

Таблица 4. Столбцы окна Parallel Task	
Имя столбца	Описание
Флаги	Показывает, какие задачи помечены, и позволяет помечать задачи и снимать с них метки.
Значки	Рядом с текущей задачей отображается желтая стрелка. Текущая задача находится на самом верхнем уровне текущего потока. Белая стрелка указывает прерванную задачу, т.е. задачу, которая была текущей во время вызова

	отладчика. Значок паузы указывает задачу, замороженную пользователем.
ID	Столбец содержит предоставленный системой номер задачи. В машинном коде этот номер является адресом задачи.
Status	<p>Столбец отображает текущее состояние задачи:</p> <p>Запланированная задача - это задача, которая еще не выполнялась и, следовательно, не имеет стека вызова, назначенного потока и других соответствующих сведений.</p> <p>Запущенная задача - это задача, которая выполняла код, пока не была прервана в отладчике.</p> <p>Находящаяся в ожидании задача - это задача, заблокированная вследствие ожидания сигнала события, освобождения блокировки или завершения другой задачи.</p> <p>Заблокированная задача - это находящаяся в ожидании задача, чей поток заблокирован другим потоком.</p>
Location	Столбец, который отображает текущее расположение в стеке вызова задачи.
Task	Столбец отображает исходный метод и аргументы, которые были переданы в задачу при ее создании.
Parent	Столбец содержит идентификатор задачи, создавшей данную задачу. Если эта ячейка пуста, то у задачи нет родительской задачи. Это применимо только для управляемых программ.
Thread Assignment	Данный столбец содержит идентификатор и имя потока, в котором запущена задача.
AppDomain	Столбец содержит информацию о домене приложения (для управляемого кода), в котором выполняется задача.
task_group	Столбец отображает информацию о адрес объекта task_group, который запланировал задачу. Для асинхронных агентов и упрощенных задач этот столбец содержит значение 0.

Окно Parallel Stacks

Окно Parallel Stacks или параллельные стеки (*Рис. 14*) - это окно применяется при отладке многопоточных приложений, и содержит сведения о стеке вызова для всех потоков приложения. Оно также позволяет переходить в различные потоки и кадры стека в потоках.

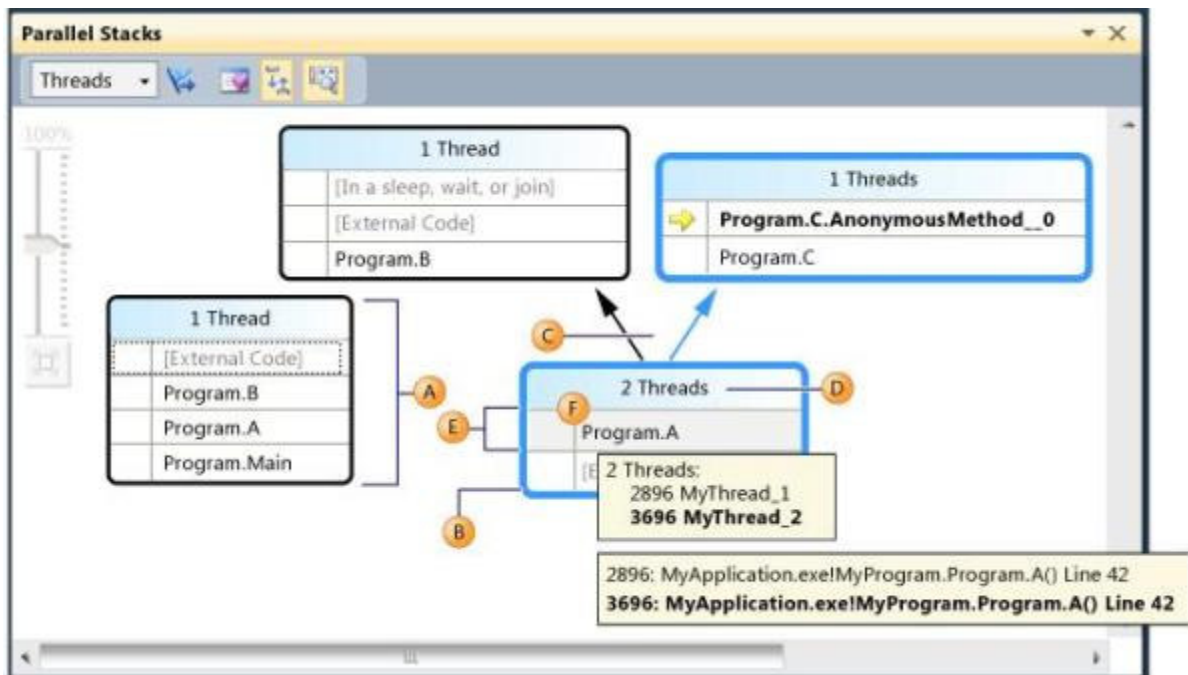





Рис. 14. Окно Parallel Stacks

На Рис. 14 - путь вызова текущего потока выделен, синим, а активный кадр стека обозначается желтой стрелкой. Текущий кадр стека можно изменить, выбрав другой метод в окне Parallel Stacks. При этом также может измениться текущий поток в зависимости от того, входит ли выбранный метод в текущий поток или является частью другого потока. В Табл. 5 представлены компоненты окна Parallel Stacks

Таблица 5. Компоненты окна Parallel Stacks		
	Имя элемента	Описание
A	Сегмент или узел стека вызова	Содержит последовательности контекстов методов для одного или нескольких потоков. Если узел не имеет линий со стрелками, то он представляет собой единый путь вызова для потоков.
B	Синее выделение	Указывает путь вызова текущего потока.
C	Линии со стрелками	Соединяют узлы и показывают единый путь вызова для потоков.
D	Всплывающая подсказка заголовка узла	Показывает идентификатор и пользовательское имя каждого потока, путь вызова которого использует данный узел.
E	Контекст метода	Представляет один или несколько кадров стека одного метода.
F	Всплывающая подсказка для контекста метода	Показывает подробные сведения всех кадров стека, которые представляются контекстом метода. Кадры стека для текущего потока отображаются жирным шрифтом.

В Табл. 6 описываются значки, которые предоставляют сведения об активных и текущих кадрах стека.

Таблица 6. Значки сведений об активных и текущих кадрах стека

Значок	Описание
	Указывает на то, что контекст метода содержит активный кадр стека текущего потока.
	Указывает на то, что контекст метода содержит активный кадр стека потока, который не является текущим.
	Указывает на то, что контекст метода содержит текущий кадр стека. Имя этого метода выделено жирным шрифтом во всех узлах.

На Рис. 15 и в Табл. 7 представлены элементы управления, доступные на панели управления "Параллельные стеки".

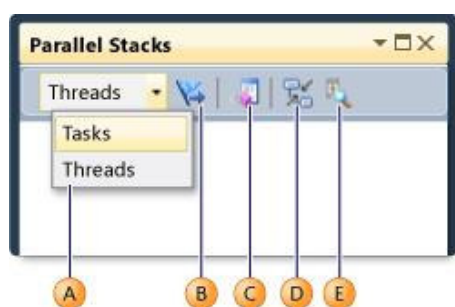


Рис. 15. Элементы управления окна Parallel Stacks

Таблица 7. Элементы управления окна Parallel Stacks		
	Элемент управления	Описание
A	Поле со списком "Потоки"/"Задачи"	Переключает отображение между стеками вызова для потоков и стеками вызова для задач.
B	Показывать только помеченные	Отображает стеки вызова только для потоков (или задач), помеченных в окнах "Потоки" или "Параллельные задачи".
C	Представление метода	Переключает представление стека и представление метода.
D	Автопрокрутка к текущему кадру стека	Автоматически прокручивает схему для отображения текущего кадра стека. Этот компонент применяется при изменении текущего кадра стека из других окон или при появлении точки останова в крупных диаграммах.
E	Переключить элемент управления масштабом	Отображает или скрывает элемент управления масштабом. Чтобы изменить масштаб, можно также нажать клавишу CTRL и повернуть колесо мыши вне зависимости от того, где находится элемент управления.

