

# Искусство отладки

Алексей Дрожжин

# Структура лекции

## Основные разделы и подразделы.

- Введение
- Подготовка и планирование
  - Неизбежность отладки,
  - Поддержка процессом,
  - Информационное и программное обеспечение
- Инструменты отладки ПО
  - Debugger,
  - Profiler,
  - Memory leaks detector,
  - UT Coverage
- Отладчик Visual Studio
  - Break points
  - Watch window
- Типовые дефекты
- Crash dumps
- Q&A
- Литература

# Введение

# Появление термина debugging.

Bug :

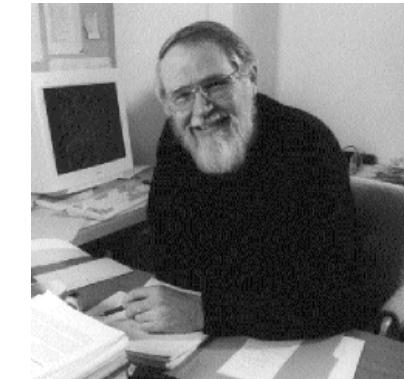
- Клоп
  - Букашка
  - Жук
  - Микроб



# Введение: Определение отладки (“debugging”)

Процесс определения и устранения причин ошибок в программе.

“ Отлаживать код вдвое сложнее, чем писать.  
Если Вы используете весь свой интеллект  
при написании программы, вы по определению  
недостаточно умны, чтобы её отладить.”



Брайан Керниган

# Введение: Эффективность навыка отладки

Исследования далёкого 1975 года.

Отобрали группу профессиональных программистов с 4-х летним стажем: 3 - “лучших”, 3 - “худших”.

Задача: найти и исправить программу, содержащую 12 дефектов.

	“Лучший”	“Худший”
Среднее время отладки (мин.)	5,0	14,1
Число не обнаруженных дефектов	0,7	1,7
Число внесенных дефектов	3,0	7,7

# Подготовка и планирование

- Неизбежность отладки,
- Поддержка процессом,
- Информационное и программное обеспечение
- Алгоритм поиска и устранения дефекта

# Подготовка и планирование: Неизбежность отладки

# Подготовка и планирование: Неизбежность отладки

- Человеческий фактор



# Подготовка и планирование: Неизбежность отладки

- Человеческий фактор
- Короткие/невозможные сроки



# Подготовка и планирование: Неизбежность отладки

- Человеческий фактор
- Короткие/невозможные сроки
- Непонимание требований



# Подготовка и планирование: Неизбежность отладки

- Человеческий фактор
- Короткие/невозможные сроки
- Непонимание требований
- Недостаток экспертизы/знаний



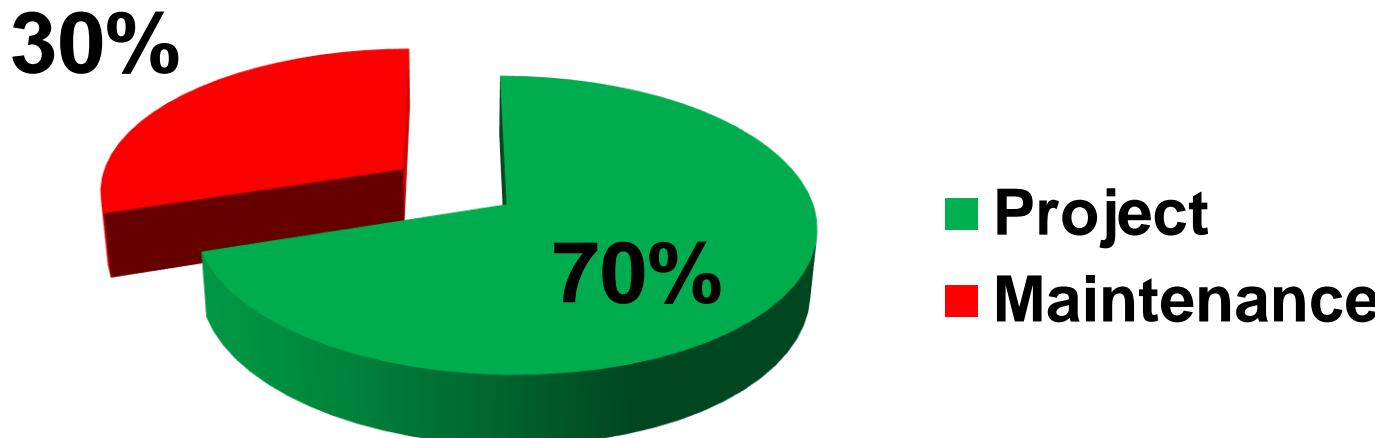
# Подготовка и планирование: Неизбежность отладки

- Человеческий фактор
- Короткие/невозможные сроки
- Непонимание требований
- Недостаток экспертизы/знаний
- Пренебрежение качеству



# Подготовка и планирование: Поддержка процессом

До 20-40% ресурсов тратится на поддержку существующего ПО, поиск и исправление дефектов.

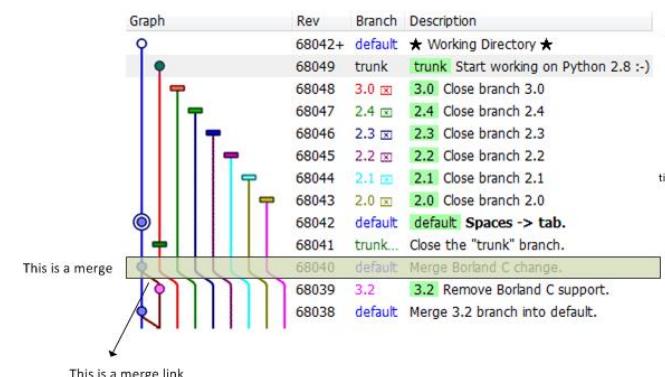


# Подготовка и планирование: Информационное и программное обеспечение

## Системы контроля версий.



14



# Подготовка и планирование: Информационное и программное обеспечение

## Системы отслеживания дефектов.



JIRA Dashboards - Projects - Issues - Agile - Capture - Create issue

Scrum: Teams in Space

SPRINT: Sprint 1 - QUICK FILTERS: Product UI Server Only My Issues Recently Updated

4 To Do    4 In Progress Min 3 Max 5    1 Code Review    4 Done

▼ TIS Developer Love 4 issues

- TIS-46 Update LocalTransportC to handle
- TIS-40 Update FlightController to handle
- TIS-8 Requesting available flights is now taking >
- TIS-67 Developer Toolbox does not display by default

▼ Everything Else 9 issues

- TIS-45 Email non registered users to sign
- TIS-43 Extend booking experience in UI to include
- TIS-44 Reward Customers an extra 5-10%
- TIS-49 Draft network plan for Mars Office
- TIS-68 Homepage footer uses an inline style -
- TIS-42 Extend booking experience in UI to include
- TIS-46 Add a pointer to main.css file to instruct users

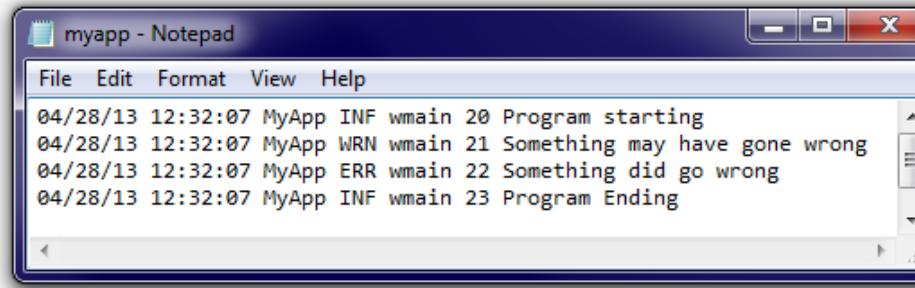
Developer Toolbox does not display by default

People  
Reporter: Jennifer Evans  
Assignee: Jennifer Evans

Development  
5 branches  
3 commits  
1 pull request OPEN  
1 build ✓

# Подготовка и планирование: Информационное и программное обеспечение

## Логирование.

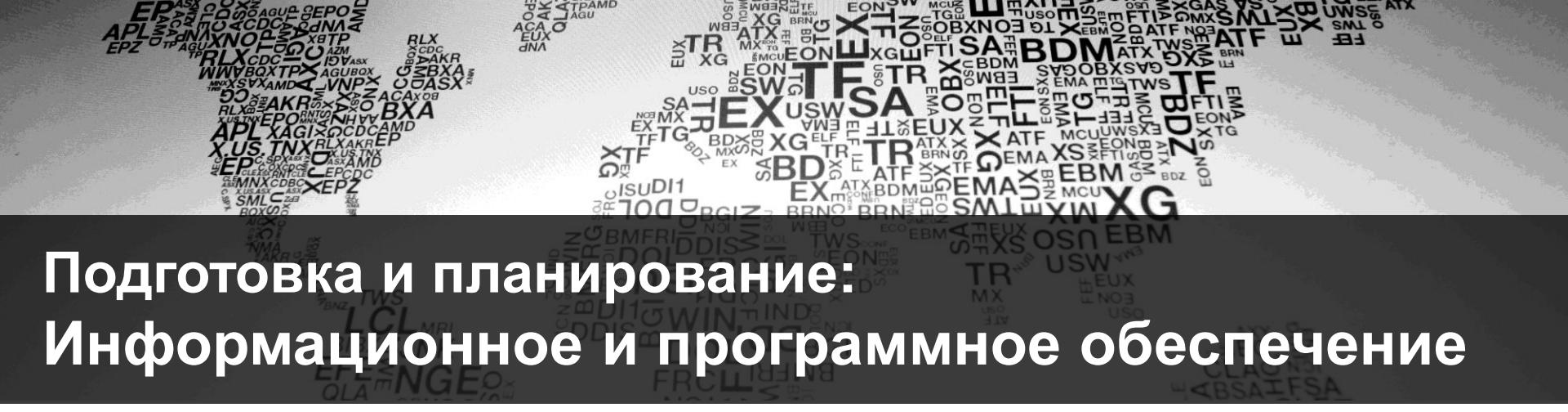


### Возможности хорошего логгера:

- Уровни логирования и фильтрация сообщений
- Ротация лог-файлов
- Возможность записи сообщений не только в файлы
- Потокобезопасность
- Асинхронное логирование
- Гибкое форматирование и конфигурация логов

# Подготовка и планирование: Информационное и программное обеспечение

## Плохой логгер



```
test.log - Notepad
File Edit Format View Help
[17-10-2019 18-42-54]: from_do_a0
17-10-2019 18-42-54from_do_b0
17-10-2019 18-42-54[from_do_b17-10-2019 18-42-54]: from_do_a1
[17-10-2019 18-42-54]: from_do_a2

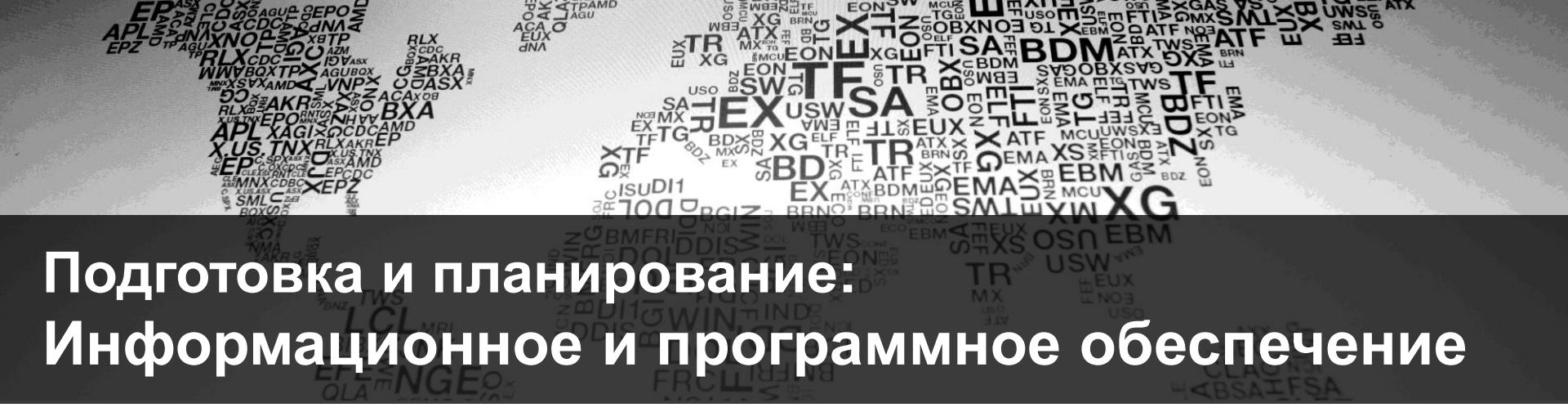
[17-10-2019 18-42-54]: from_do_a317-10-2019 18-42-54from_do_b2
17-10-2019 18-42-54
from_do_b3[
17-10-2019 18-42-54]: from_do_a4
[17-10-2019 18-42-54]: from_do_a5
17-10-2019 18-42-54from_do_b4
17-10-2019 18-42-54[from_do_b17-10-2019 18-42-54]: from_do_a6
5[
17-10-2019 18-42-54]: from_do_a7
[17-10-2019 18-42-54from_do_b6
17-10-2019 18-42-54]: from_do_a8
17-10-2019 18-42-54from_do_b7
[17-10-2019 18-42-54from_do_b817-10-2019 18-42-54]: from_do_a9
[17-10-2019 18-42-54]: from_do_a10

[17-10-2019 18-42-54from_do_b9
17-10-2019 18-42-5417-10-2019 18-42-54]: from_do_a11
from_do_b[1017-10-2019 18-42-54]: from_do_a12

[17-10-2019 18-42-54from_do_b11
17-10-2019 18-42-54]: from_do_a13
[17-10-2019 18-42-54from_do_b12
17-10-2019 18-42-5417-10-2019 18-42-54]: from_do_a14
from do b1117-10-2019 18-42-54]: from do a15
<
```

# Подготовка и планирование: Информационное и программное обеспечение

## Плохой логгер



```
test.log - Notepad
File Edit Format View Help
[17-10-2019 18-42-54]: from_do_a0
17-10-2019 18-42-54from_do_b0
17-10-2019 18-42-54[from_do_b17-10-2019 18-42-54]: from_do_a1
[17-10-2019 18-42-54]: from_do_a2

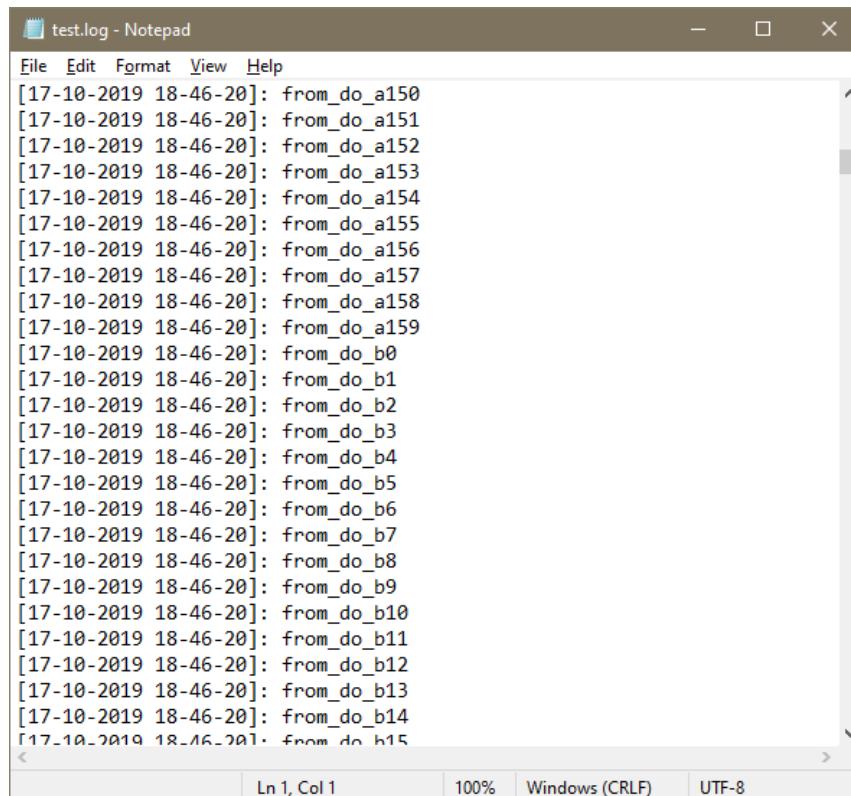
[17-10-2019 18-42-54]: from_do_a317-10-2019 18-42-54from_do_b2
17-10-2019 18-42-54
from_do_b3[
17-10-2019 18-42-54]: from_do_a4
[17-10-2019 18-42-54]: from_do_a5
17-10-2019 18-42-54from_do_b4
17-10-2019 18-42-54[from_do_b17-10-2019 18-42-54]: from_do_a6
5[
17-10-2019 18-42-54]: from_do_a7
[17-10-2019 18-42-54from_do_b6
17-10-2019 18-42-54]: from_do_a8
17-10-2019 18-42-54from_do_b7
[17-10-2019 18-42-54from_do_b817-10-2019 18-42-54]: from_do_a9
[17-10-2019 18-42-54]: from_do_a10

[17-10-2019 18-42-54from_do_b9
17-10-2019 18-42-5417-10-2019 18-42-54]: from_do_a11
from_do_b[1017-10-2019 18-42-54]: from_do_a12

[17-10-2019 18-42-54from_do_b11
17-10-2019 18-42-54]: from_do_a13
[17-10-2019 18-42-54from_do_b12
17-10-2019 18-42-5417-10-2019 18-42-54]: from_do_a14
from do b1117-10-2019 18-42-54]: from do a15
<
```

# Подготовка и планирование: Информационное и программное обеспечение

## Хороший логгер



The screenshot shows a Notepad window titled "test.log - Notepad". The window contains a list of log entries, each consisting of a timestamp and a message. The messages are all variations of "from\_do\_xxx" where xxx is a two-digit number ranging from 150 down to 15. The timestamp is consistently "[17-10-2019 18-46-20]". The Notepad interface includes standard menu options like File, Edit, Format, View, and Help, and status bar indicators for line count (Ln 1, Col 1), zoom level (100%), file type (Windows (CRLF)), and encoding (UTF-8).

```
[17-10-2019 18-46-20]: from_do_a150
[17-10-2019 18-46-20]: from_do_a151
[17-10-2019 18-46-20]: from_do_a152
[17-10-2019 18-46-20]: from_do_a153
[17-10-2019 18-46-20]: from_do_a154
[17-10-2019 18-46-20]: from_do_a155
[17-10-2019 18-46-20]: from_do_a156
[17-10-2019 18-46-20]: from_do_a157
[17-10-2019 18-46-20]: from_do_a158
[17-10-2019 18-46-20]: from_do_a159
[17-10-2019 18-46-20]: from_do_b0
[17-10-2019 18-46-20]: from_do_b1
[17-10-2019 18-46-20]: from_do_b2
[17-10-2019 18-46-20]: from_do_b3
[17-10-2019 18-46-20]: from_do_b4
[17-10-2019 18-46-20]: from_do_b5
[17-10-2019 18-46-20]: from_do_b6
[17-10-2019 18-46-20]: from_do_b7
[17-10-2019 18-46-20]: from_do_b8
[17-10-2019 18-46-20]: from_do_b9
[17-10-2019 18-46-20]: from_do_b10
[17-10-2019 18-46-20]: from_do_b11
[17-10-2019 18-46-20]: from_do_b12
[17-10-2019 18-46-20]: from_do_b13
[17-10-2019 18-46-20]: from_do_b14
[17-10-2019 18-46-20]: from_do_b15
```

# Подготовка и планирование: Информационное и программное обеспечение

## Уровни логирования.



**Debug** - сообщения отладки, профилирования



**Information** - обычные сообщения, информирующие о действиях системы.



**Warning** - произошло что-то странное. Следует разобраться в том, что произошло, что это означает, и отнести ситуацию либо к инфо-сообщению, либо к ошибке



**Error** - ошибка в работе системы, требующая вмешательства. Что-то не сохранилось, что-то отвалилось и т.д.



**Fatal** - особый класс ошибок. Такие ошибки приводят к неработоспособности системы в целом, или неработоспособности одной из подсистем.

# Подготовка и планирование: Алгоритм поиска и устранения дефекта

## Основные шаги:

Воспроизведение (reproducing)

Поиск бага (investigation)

Поиск решения (fixing)

Проверка решения (testing)

# Подготовка и планирование: Краткий повтор раздела

- Неизбежность отладки
  - человеческий фактор
  - сроки
  - требования
  - знания
  - качество
- Поддержка процессом
- Информационное и программное обеспечение
  - Системы контроля версий (git, mercurial, svn, cvs,...)
  - Системы отслеживания дефектов
  - Логирование
- Алгоритм поиска и устранения дефекта

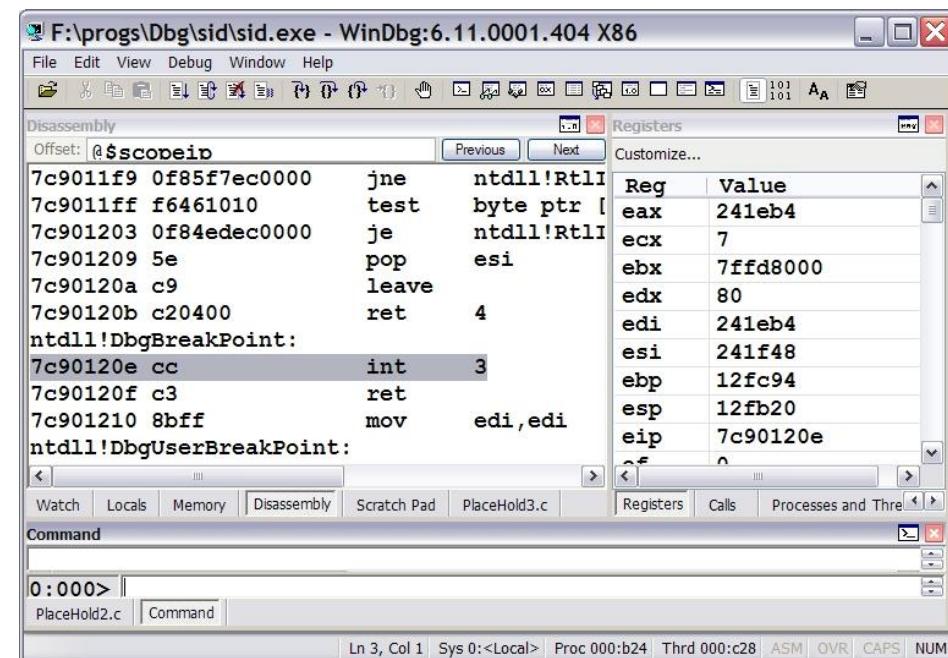
# Инструменты отладки ПО

- Debugger
- Profiler
- Memory leaks detector
- Testing

# Инструменты отладки ПО: Debugger (отладчик)

## Возможности:

- Пошаговая трассировка
- Отслеживание/установка /изменение значения переменных
- Установка/удаление условий остановки



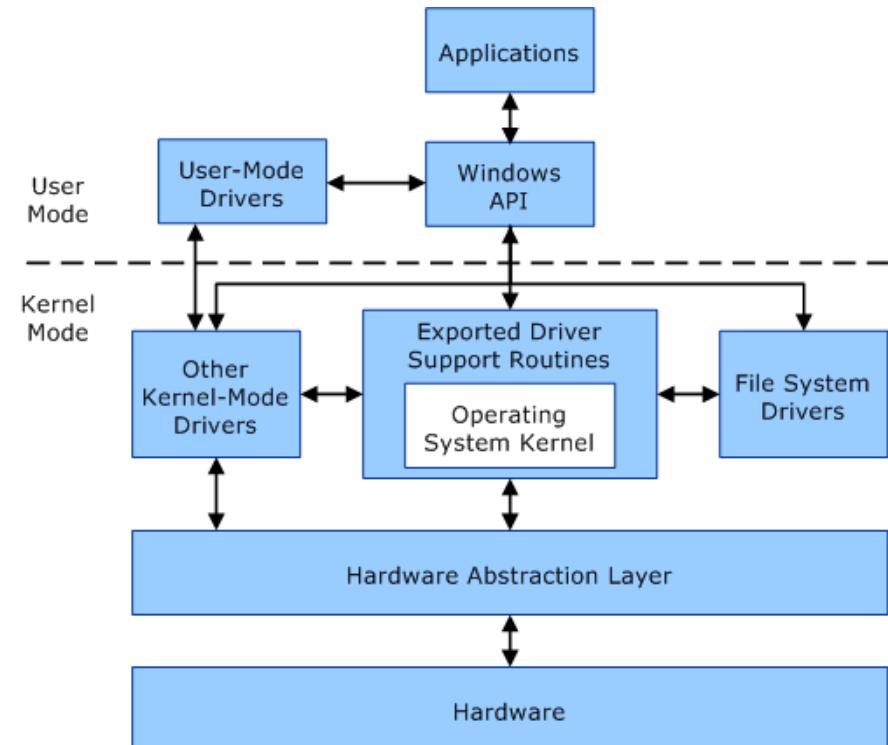
# Инструменты отладки ПО: Debugger (отладчик)

## Типы Windows-отладчиков

- Отладка в режиме пользователя
- Отладка в режиме ядра

## Представители:

- Visual Studio
- WinDbg
- SoftICE
- Dtrace
- OllyDbg



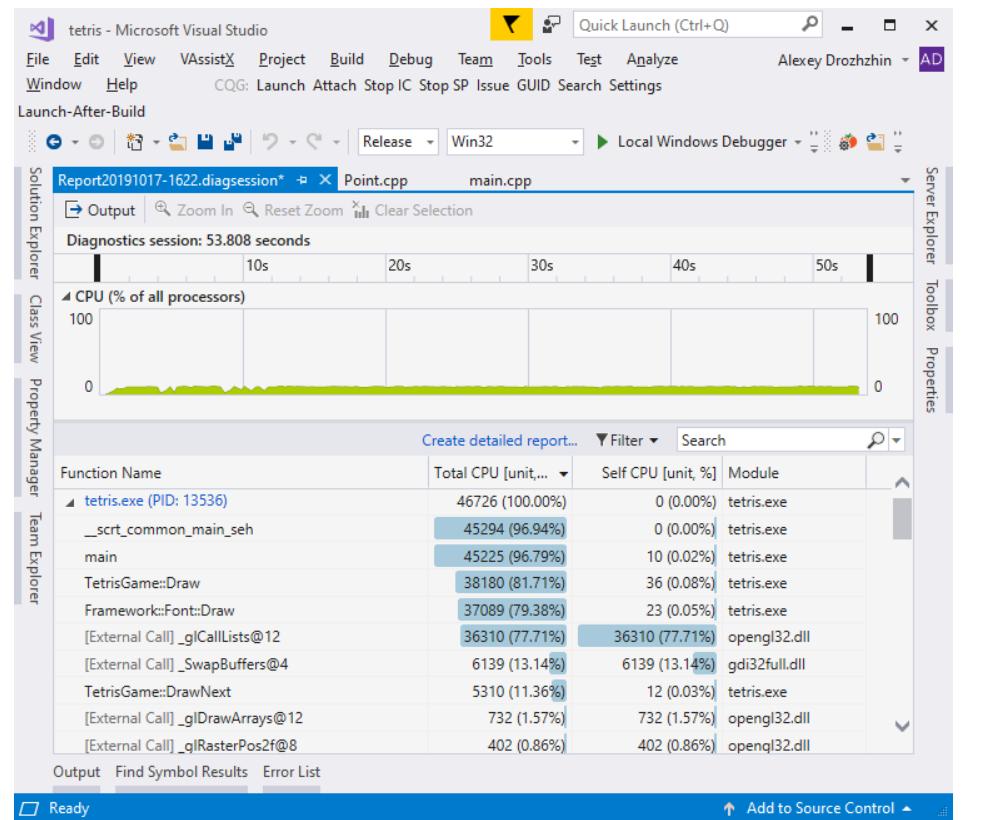
# Инструменты отладки ПО: Profiler (профилировщик)

Определяет:

- время выполнения отдельных фрагментов,
- число верно предсказанных условных переходов,
- количества вызовов той или иной точки программы.

Представители:

- Intel Vtune
- Visual Studio
- Xperf (+WPA)
- Valgrind



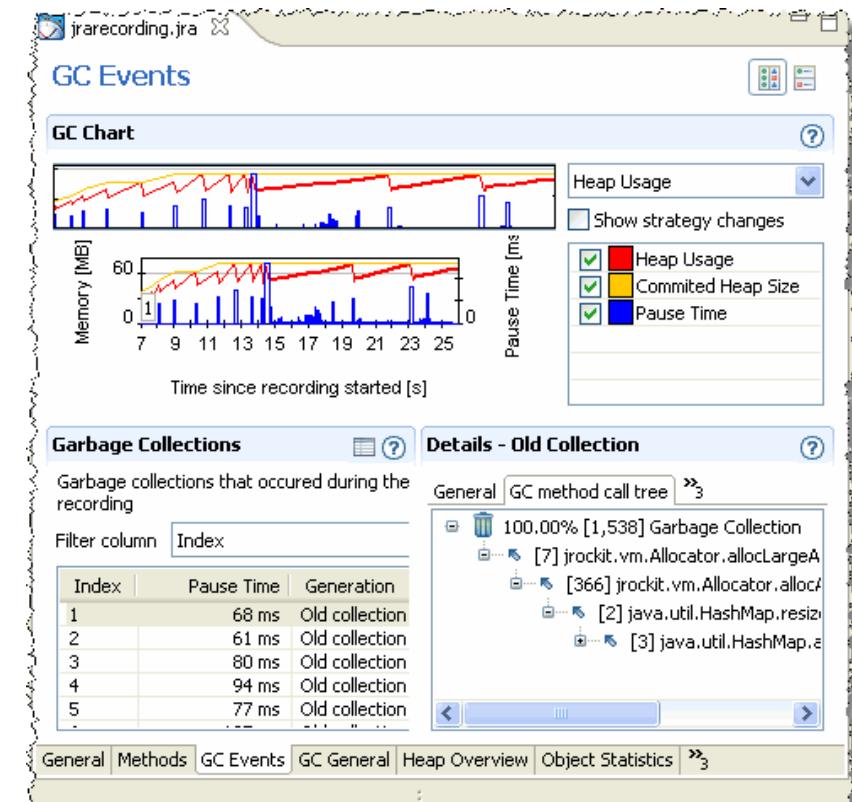
# Инструменты отладки ПО: Memory leaks detector

## Возможности:

- Выявление “утечки” памяти
  - Анализ использования памяти

## Представители:

- Visual Studio
  - Visual Leak Detector
  - Valgrind
  - Glow Code



# Инструменты отладки ПО: Testing

## Применение:

- *Юнит Тестирование* — тестирование отдельных компонентов системы
- *Интеграционное Тестирование* — комплексное тестирование системы после соединения всех отдельных компонентов
- *Нагрузочное Тестирование* — тестирование производительности системы
- *Стресс Тестирование* — тестирование отказоустойчивости системы в нештатных ситуациях
- *Регрессивное Тестирование* — тестирование уже протестированных участков исходного кода. После внесения изменений в программу может перестать работать то, что должно было продолжать работать.

# Инструменты отладки ПО: Краткий повтор раздела

- Debugger
- Profiler
- Memory leaks detector
- Testing

# Отладчик Visual Studio

- Как пользоваться
- Break points
- Watch window

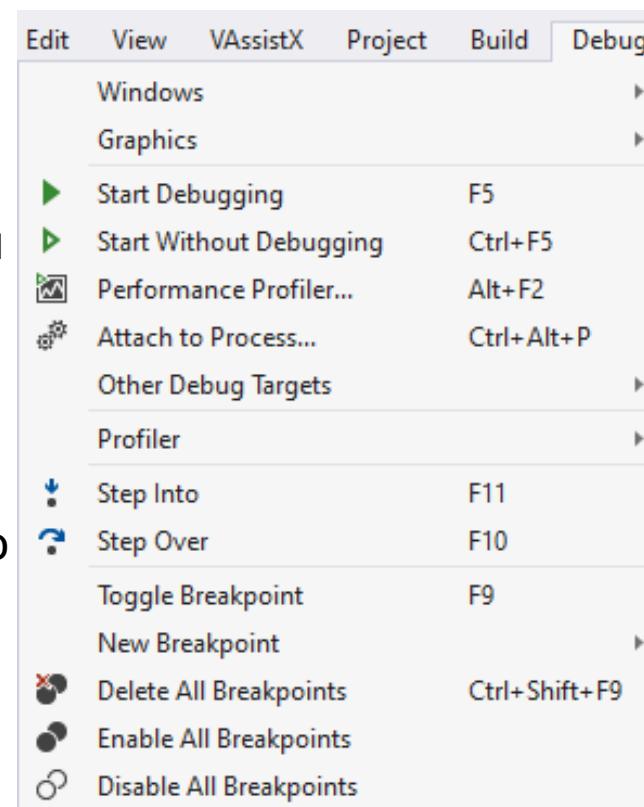
# Отладчик Visual Studio: Как пользоваться

Запуск с отладкой  
Запуск без отладки

Шаг с заходом в функцию  
Шаг без захода в функцию

Поставить Breakpoint

Управление Breakpoint



Attach Stop IC Stop SP Issue GUID Search Settings Launch-After-Build

Process: [0x5874] tetris.exe Lifecycle Events Thread: [0x3C38] Main Thread

Disassembly main.cpp Point.cpp

main void main()

```
1 #include "game\TetrisGame.h"
2
3
4 /*
5 * The entry point. I like to make it look clear and small, so pretty much everything is covered inside other
6 */
7 void main()
8 {
9     int b = 600;
10    int i = 800;
11    // All I have to do is make a TetrisGame and then Run() it!
12    TetrisGame *game = new TetrisGame(i, b);
13    i = 400;
14    game->Run();
15
16    // Don't forget to clean up tho
17    delete game; ▶
18 }
```

Run execution to here

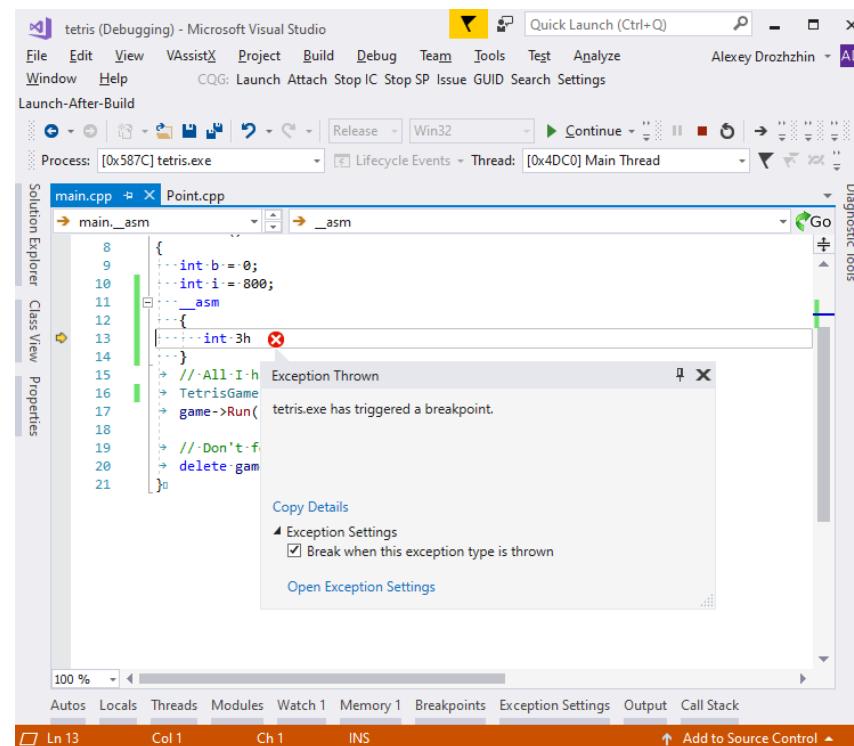
100 %

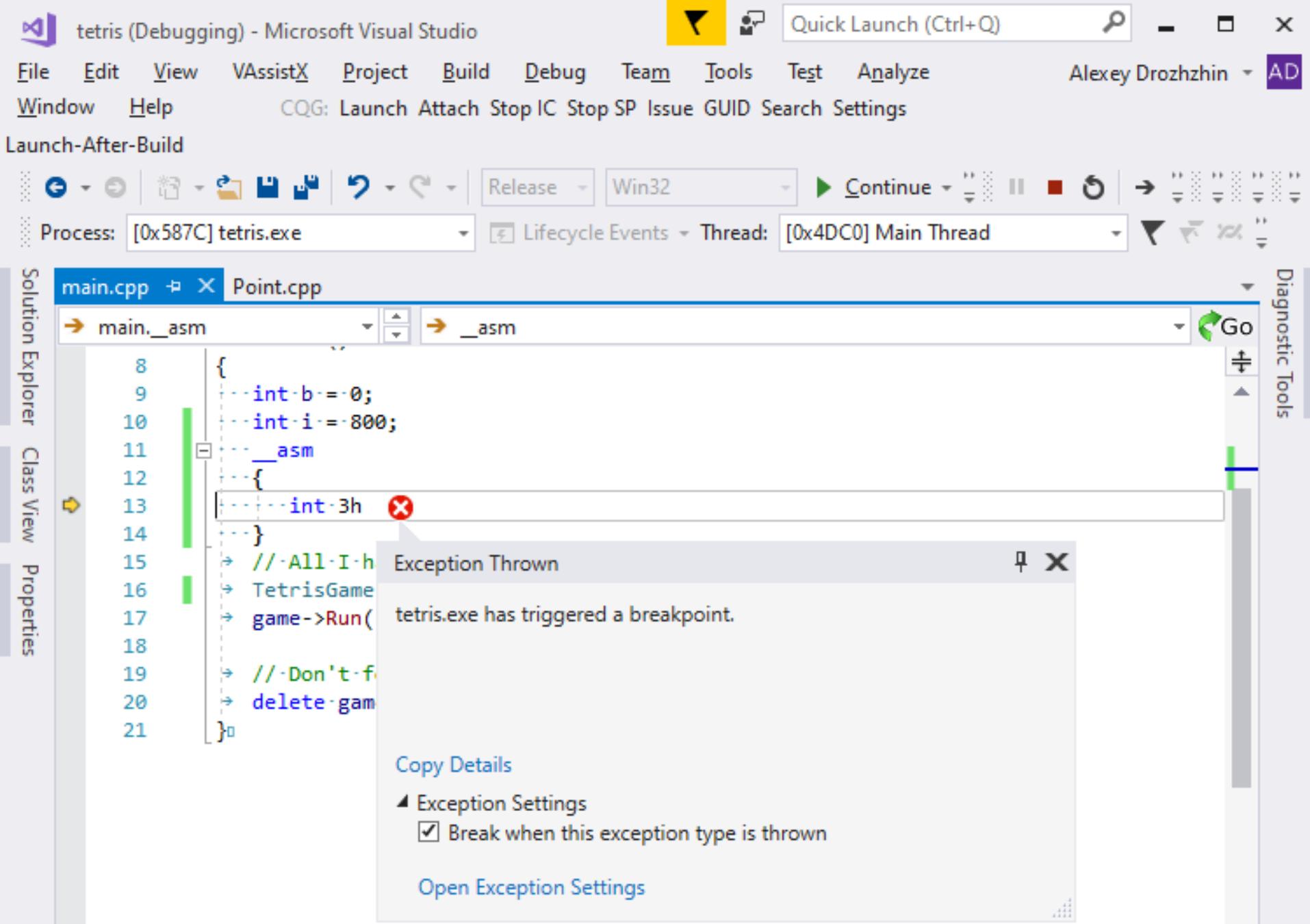
Autos Locals Threads Modules Memory 1 Breakpoints Exception Settings Output Call Stack

# Отладчик Visual Studio: Break points

```
9 // All I have to do is make a TetrisGame
10 int b = 0;
11 int i = 800;
12 // All I have to do is make a TetrisGame
13 TetrisGame *game = new TetrisGame(b, i);
14 game->Run();
```

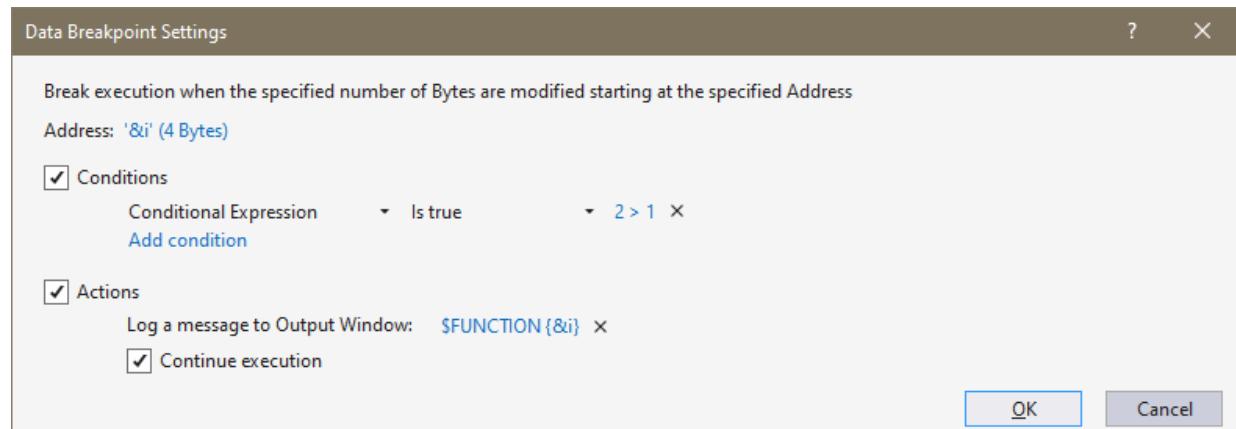
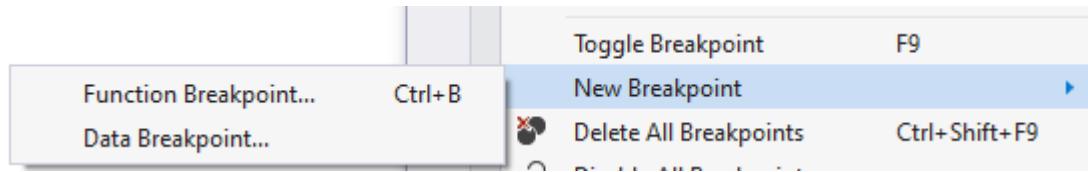
```
// STATUS_BREAKPOINT
// (0x80000003)
_asm
{
    int 3;
}
```





# Отладчик Visual Studio: Break points: основные типы

- Simple breakpoint
- Break at function
- Data breakpoint



The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes 'File', 'Edit', 'View', 'Project', 'Build', 'Tools', 'Help', and 'Switch To'. The toolbar contains icons for file operations like Open, Save, and Print, along with symbols for Debug, Win32, Continue, Stop, and Break.

The 'Process' dropdown shows '[0x1AD8] tetris.exe'. The 'Lifecycle Events' and 'Thread' dropdowns are set to their default values.

The Solution Explorer on the left lists 'main.cpp' and 'Point.cpp' under the 'Tetris' project.

The Class View shows the class hierarchy.

The Properties window is visible on the far left.

The Diagnostic Tools window is open on the right side.

The main code editor displays the 'main.cpp' file:

```
main.cpp  X Point.cpp
main void main()
1 #include "game\TetrisGame.h"
2
3
4 /*
5  * The entry point. I like to make it look clear and small, so pretty much everyth:
6 */
7 void main()
```

A green 'Go' button is located at the top right of the code editor.

The Output window below shows the assembly output for the 'tetris.exe' process:

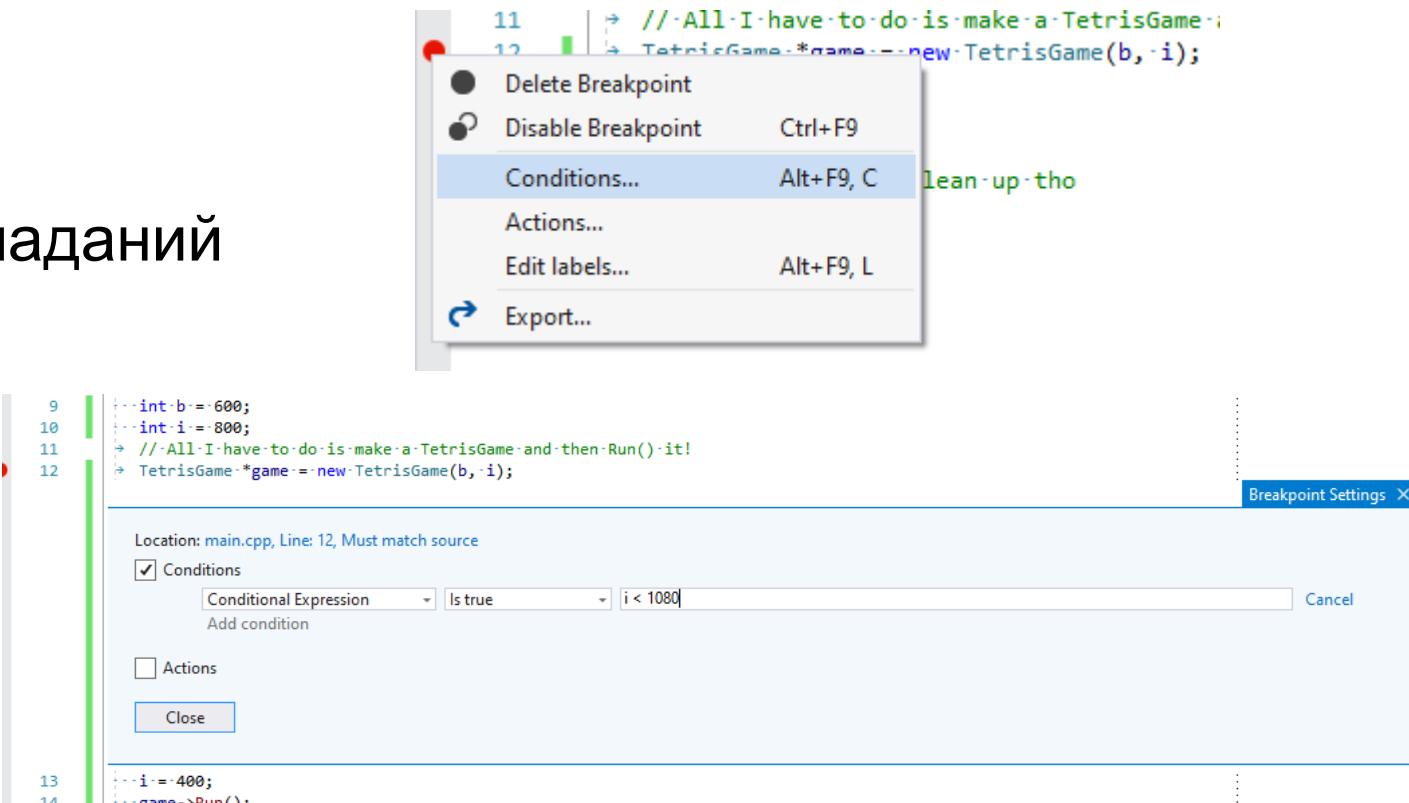
```
Show output from: Debug
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\version.dll'. Symbol loading disabled by In
'tetris.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\version.dll'
'tetris.exe' (Win32): Loaded 'C:\Windows\System32\DriverStore\FileRepository\igd1h64.inf_amd64
'tetris.exe' (Win32): Loaded 'C:\Windows\System32\DriverStore\FileRepository\igd1h64.inf_amd64
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\TextInputFramework.dll'. Symbol loading disa
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\CoreUIComponents.dll'. Symbol loading disabl
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\CoreMessaging.dll'. Symbol loading disabled
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ntmarta.dll'. Symbol loading disabled by In
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\WinTypes.dll'. Symbol loading disabled by In
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\WinTypes.dll'. Symbol loading disabled by In
'tetris.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\WinTypes.dll'
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\iertutil.dll'. Symbol loading disabled by In
'tetris.exe' (Win32): Loaded 'C:\Windows\SysWOW64\oleacc.dll'. Symbol loading disabled by Incl
main(void) 0x005cfb40 {0x000000190}
```

The bottom navigation bar includes 'Autos', 'Locals', 'Threads', 'Modules', 'Watch 1', 'Memory 1', 'Breakpoints', 'Exception Settings', 'Output', and 'Call Stack' tabs.

The status bar at the bottom shows 'Ready' and 'Add to Source Control' buttons.

# Отладчик Visual Studio: Break points: расширенные

- С условием  
(Condition...)
- По числу попаданий  
(Hit Count...)
- Фильтр  
(Filter...)
- По событию  
(When Hit...)

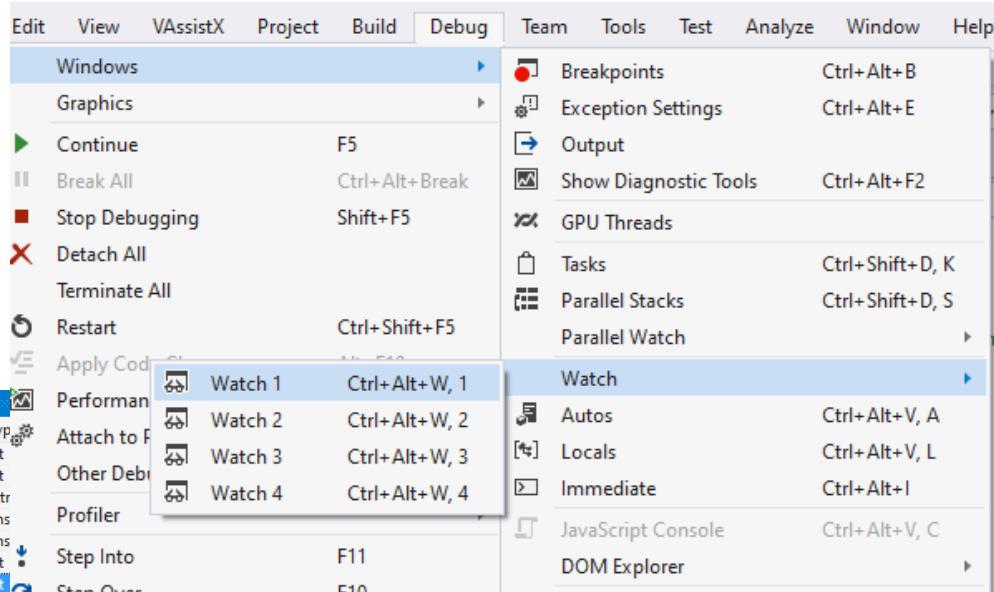


# Отладчик Visual Studio: Watch window

## Возможности:

- Просмотр значений переменных
- Редактирование переменных
- Просмотр псевдорегистров (@ERR,@EAX,@EBX, @ECX,...)
- Форматированный просмотр (d,l,u,o,x,e,c,s,hr,wm)

Watch 1		
Name	Value	Type
i_d	800	int
b_x	0x00000258	int
game	0x0082dee0 {rectBatch=0x0af11510 {...} linesBatch=0x0af113a8 {...} fo...}	Tree
eax,d	8576736	uns
esp,x	0x0053fc88	uns
game->score,d	0	int
game->state,bb	00000000000000000000000000000000	int



## Watch 1

# Отладчик Visual Studio: Краткий повтор раздела

- Как пользоваться
- Break points
- Watch window

# Типовые дефекты

- Народная мудрость
- Типы дефектов
  - Access violation
  - Memory leak
  - Heap corruption
  - Stack overflow
  - Deadlocks

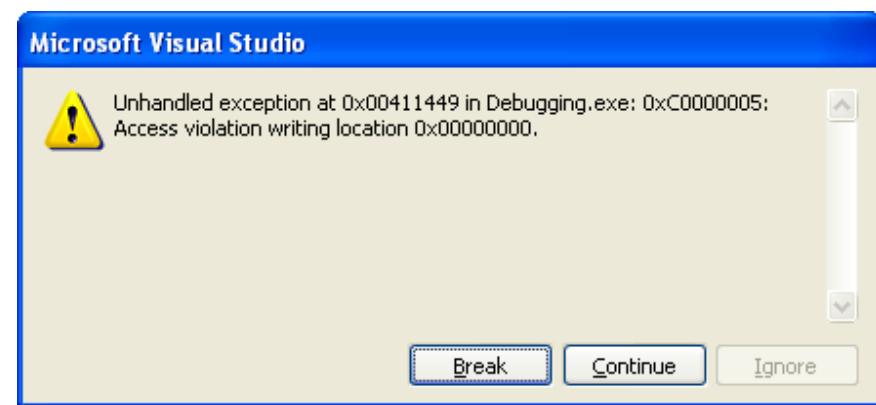
# Типовые дефекты: Народная мудрость

Обычно это так:

- Ошибка в собственном коде
- Ошибка имеет простое решение
- Ошибка возникла в последнем изменении кода
- Сегодня ошибку исправить проще, чем завтра

# Типовые дефекты: Типы дефектов

- Access violation
- Memory leak
- Heap corruption
- Stack overflow
- Deadlocks

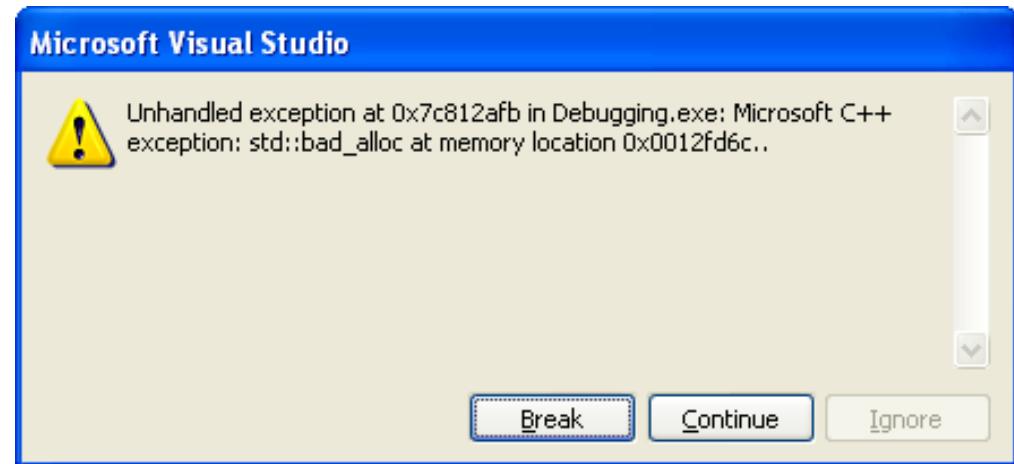


## Причины:

- Неинициализированные переменные и члены класса
- Висячие указатели

# Типовые дефекты: Типы дефектов

- Access violation
- Memory leak
- Heap corruption
- Stack overflow
- Deadlocks

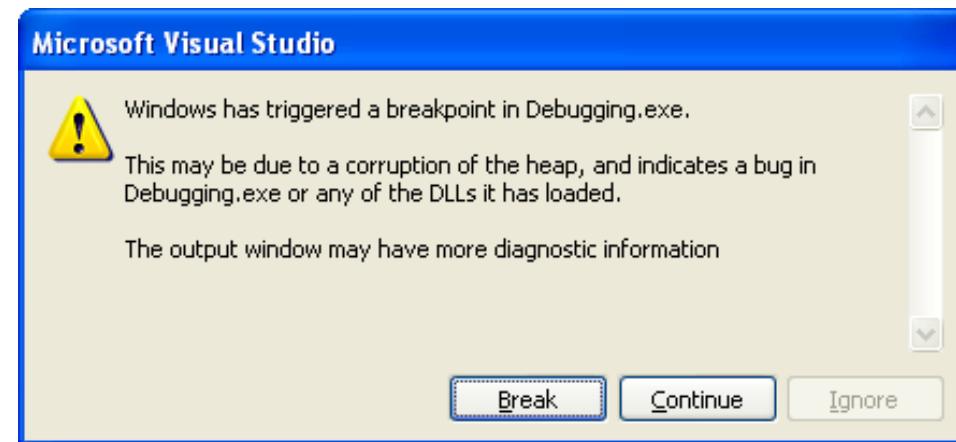


## Причины:

- Забытый delete
- Не виртуальный деструктор в базовом классе

# Типовые дефекты: Типы дефектов

- Access violation
- Memory leak
- Heap corruption
- Stack overflow
- Deadlocks

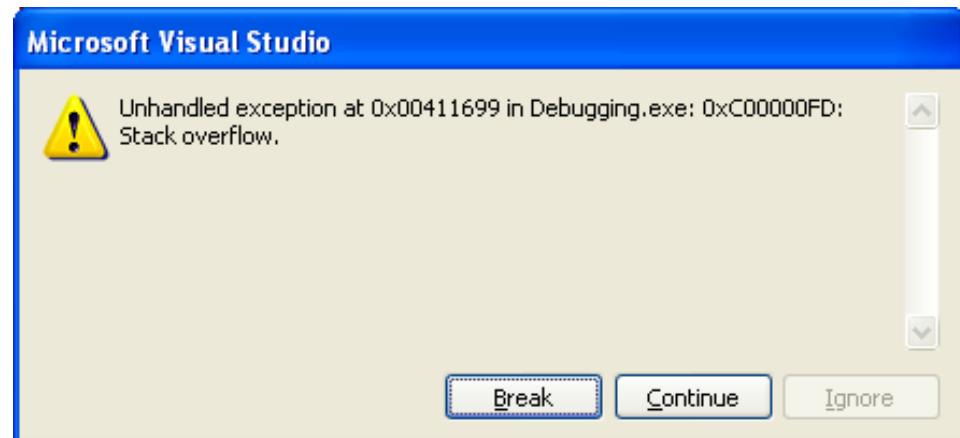


## Причины:

- Использование нескольких указателей на один адрес в куче
- Не корректное приведение типов

# Типовые дефекты: Типы дефектов

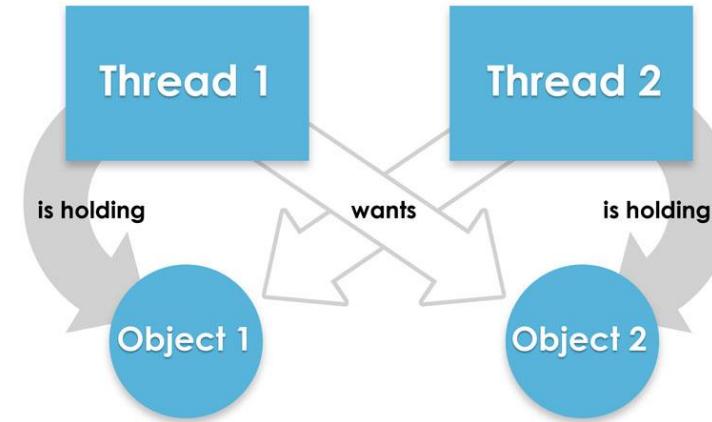
- Access violation
- Memory leak
- Heap corruption
- Stack overflow
- Deadlocks



**Причины:**  
• “Бесконечная” рекурсия  
• Слишком много переменных

# Типовые дефекты: Типы дефектов

- Access violation
- Memory leak
- Heap corruption
- Stack overflow
- Deadlocks



## Причины:

- Не правильная синхронизация доступа к объектам.

# Типовые дефекты: Краткий повтор раздела

- Народная мудрость
- Типы дефектов
  - Access violation
  - Memory leak
  - Heap corruption
  - Stack overflow
  - Deadlocks

# Crash dump



# Crash dump

- Назначение
- Создание
- Анализ
  - Без PDB
  - С PDB

# Crash dump: Назначение

Содержит информацию о состоянии программы в определённый момент времени:

- Снимок памяти
- Потоки приложения
  - Стек вызовов
  - Значения регистров ЦП
  - Значения локальных переменных
- Информация об ошибке/исключении

# Crash dump: Создание (вариант 1)

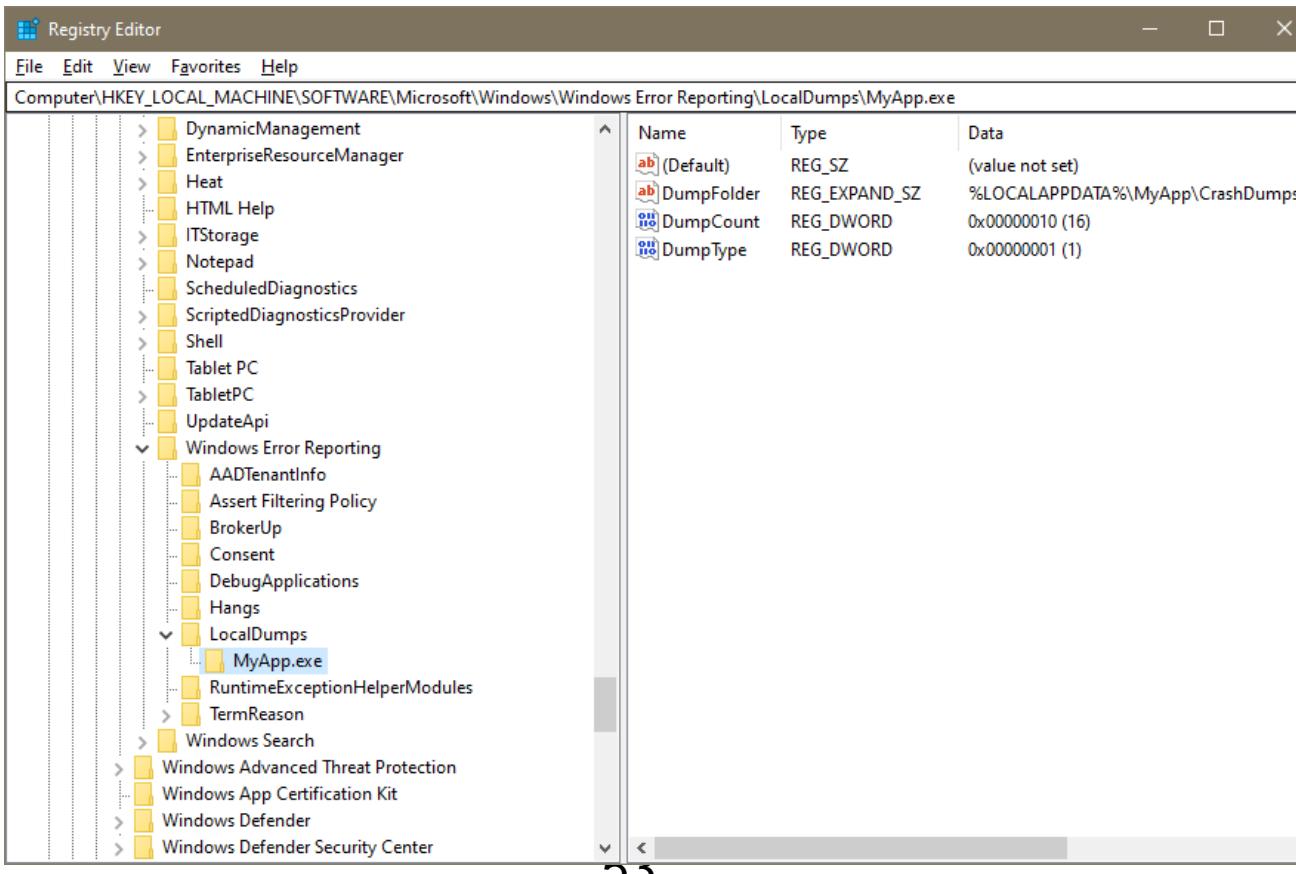
```
#include <dbghelp.h>
```

```
BOOL WINAPI MiniDumpWriteDump( HANDLE hProcess,
    DWORD ProcessId,
    HANDLE hFile,
    MINIDUMP_TYPE DumpType,
    PMINIDUMP_EXCEPTION_INFORMATION ExceptionParam,
    PMINIDUMP_USER_STREAM_INFORMATION UserStreamParam,
    PMINIDUMP_CALLBACK_INFORMATION CallbackParam );
```

```
#include <windows.h>
```

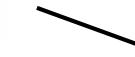
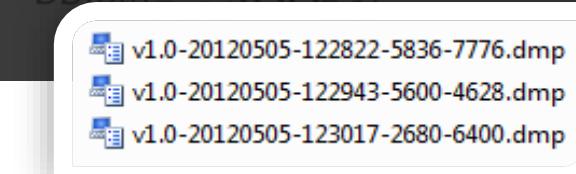
```
LONG CustomTopLevelFilter(_EXCEPTION_POINTERS *pExceptionInfo )
{
    return GenerateDump(pExceptionInfo);
}
void SetExceptionHook()
{
    ::SetUnhandledExceptionFilter(TopLevelFilter );
}
```

# Crash dump: Создание (вариант 2)



# Crash dump: Анализ

WinDbg



Visual Studio

Dump C:\Users\romangol\AppData\Local\Temp\AppName\v1.0-20120505-123017-2680-6400.dmp - WinDbg:6.12.0002.633 AMD64

File Edit View Debug Window Help

Microsoft (R) Windows Debugger Version 6.12.0002.633 AMD64  
Copyright (c) Microsoft Corporation. All rights reserved.

Loading Dump File [C:\Users\romangol\AppData\Local\Temp\AppName\v1.0-20120505-123017-2680-6400.dmp]  
User Mini Dump File: Only registers, stack and portions of memory are available

Symbol search path is: \*\*\* Invalid \*\*\*  
\*\*\*\*\*  
\* Symbol loading may be unreliable without a symbol search path. \*  
\* Use .symsrc to have the debugger choose a symbol path. \*  
\* After setting your symbol path, use .reload to refresh symbol locations. \*  
\*\*\*\*\*

Executable search path is:  
Windows 7 Version 7601 (Service Pack 1) MP (4 procs) Free x86 compatible  
Product: WinNt, suite: SingleUserTS  
Machine Name:  
Debug session time: Sat May 5 12:30:58.000 2012 (UTC + 4:00)  
System Uptime: not available  
Process Uptime: 0 days 0:00:53.000

This dump file has an exception of interest stored in it.  
The stored exception information can be accessed via .ecxr.  
(a78 1900) Integer divide-by-zero code c0000094 (first/second chance not available)  
eax=00380d78 ecx=00000000 edx=00000000 esi=00380d38 edi=0017ebd8  
eip=77e40c22 esp=0017e898 ebp=0017e8a8 iopl=0 nv up ei pl zr na pe nc  
cs=0023 ss=002b ds=002b es=0053 gs=002b fs=0000246  
\*\*\* ERROR: Symbol file could not be found. Defaulted to export symbols for ntdll.dll -  
ntdll! ZwGetContextThread+0x12:  
77e40c22 83c404 add esp,4

Ln 0, Col 0 | Sys 0:C:\User | Proc 000:a78 | Thrd 000:1900 | ASM | OVR | CAPS | NUM

tetris.exe.7240.dmp - Microsoft Visual Studio

File Edit View VAssistX Project Debug Team Tools Test Analyze Window Help

Stop IC Stop SP Issue GUID Search Settings Launch-After-Build

Solution Explorer Class View Property Manager Team Explorer

tetris.exe.7240.dmp

Minidump File Summary  
10/11/2019 12:08:38 PM

Dump Summary

Dump File Last Write Time Process Name Process Architecture Exception Code Exception Information Heap Information Error Information

tetris.exe.7240.dmp : D:\dumps\tetris\CrashDumps\tetris.exe.7240.dmp  
10/11/2019 12:08:38 PM  
tetris.exe : C:\Users\alexydr\Downloads\tetris-master\Release\tetris.exe  
x86  
0xC0000094  
The thread tried to divide an integer value by an integer divisor of zero.  
Not Present

Actions

Debug with Native Only Set symbol paths Copy all to clipboard

System Information

OS Version 10.0.18362  
CLR Version(s)

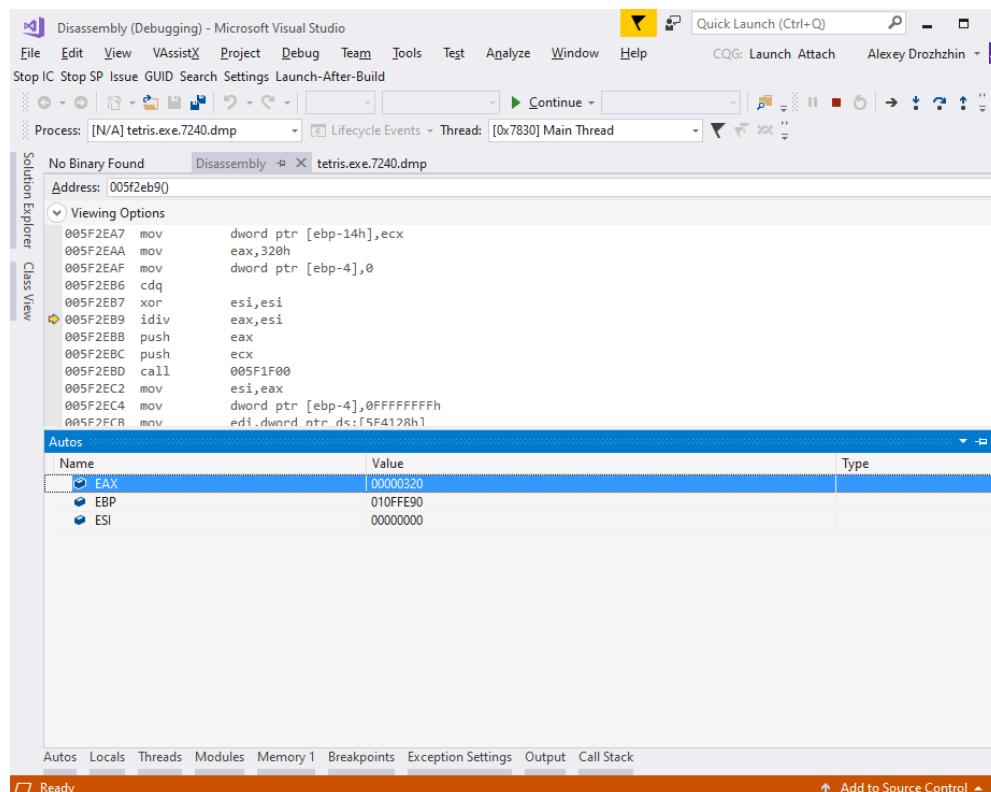
Modules

Module Name	Module Version	Module Path
tetris.exe	0.0.0	C:\Users\alexydr\Downloads\tetris-master\Release\tetris.exe
ntdll.dll	10.0.18362.1	C:\Windows\System32\ntdll.dll
kernel32.dll	10.0.18362.86	C:\Windows\System32\kernel32.dll
KERNELBASE.dll	10.0.18362.239	C:\Windows\System32\KERNELBASE.dll
apphelp.dll	10.0.18362.1	C:\Windows\System32\apphelp.dll
user32.dll	10.0.18362.1	C:\Windows\System32\user32.dll
win32u.dll	10.0.18362.239	C:\Windows\System32\win32u.dll
winmm.dll	10.0.18362.1	C:\Windows\System32\winmm.dll

Output Find Symbol Results Error List

Ready Add to Source Control

# Crash dump: Анализ: без PDB



Stop IC Stop SP Issue GUID Search Settings Launch-After-Build



No Binary Found      Disassembly    tetris.exe.7240.dmp

Address: 005F2EB90

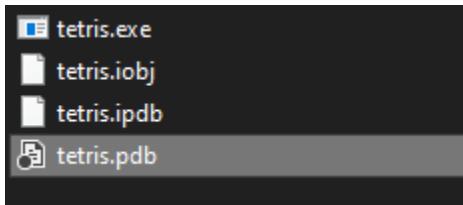
Viewing Options

005F2EA7	mov	dword ptr [ebp-14h],ecx
005F2EAA	mov	eax,320h
005F2EAF	mov	dword ptr [ebp-4],0
005F2EB6	cdq	
005F2EB7	xor	esi,esi
005F2EB9	idiv	eax,esi
005F2EBB	push	eax
005F2EBC	push	ecx
005F2EBD	call	005F1F00
005F2EC2	mov	esi,eax
005F2EC4	mov	dword ptr [ebp-4],0FFFFFFFh
005F2FCB	mov	edi,dword ptr ds:[5F4128h]

## Autos

Name	Value	Type
EAX	00000320	
EBP	010FFE90	
ESI	00000000	

# Crash dump: Анализ: с PDB



main.cpp (Debugging) - Microsoft Visual Studio

File Edit View VAssistX Project Debug Team Tools Test Analyze Window Help

Stop IC Stop SP Issue GUID Search Settings Launch-After-Build

Process: [N/A] tetris.exe.12456.dmp Lifecycle Events Thread: [0xA3C] Main Thread

Solution Explorer Class View

main.cpp Disassembly tetris.exe.12456.dmp

main

```
1  #include "game\TetrisGame.h"
2
3
4  /*
5   * The entry point. I like to make it look clear and small, so pretty much everything is covered inside other classes
6   */
7  void main()
8  {
9      int b = 0;
10     int i = 800;
11     // All I have to do is make a TetrisGame and then run() it!
12     TetrisGame *game = new TetrisGame(800, i/b); X
13     i = 400;
14     game->Run();
15
16     // Don't forget to clean up though
17     delete game;
18 }
```

Exception Unhandled

Unhandled exception at 0x00DA88BC (tetris.exe) in tetris.exe.12456.dmp: 0xC0000094: Integer division by zero.

Copy Details Exception Settings

100 %

Autos Locals Threads Modules Memory 1 Breakpoints Exception Settings Output Call Stack

Ready Ln 12 Col 1 Ch 1 INS

Add to Source Control

Stop IC Stop SP Issue GUID Search Settings Launch-After-Build



main.cpp

Disassembly

tetris.exe.12456.dmp

```
main
void main()
```

```
1 #include "game\TetrisGame.h"
2
3
4 /* 
5  * The entry point. I like to make it look clear and small, so pretty much everything is covered inside other classes
6 */
7 void main()
8 {
9     int b = 0;
10    int i = 800;
11    // All I have to do is make a TetrisGame and then Run() it!
12    TetrisGame *game = new TetrisGame(800, i/b); X
13    i = 400;
14    game->Run();
15
16    // Don't forget to clean up tho
17    delete game;
18 }
```

## Exception Unhandled

Unhandled exception at 0x00DA88BC (tetris.exe) in tetris.exe.12456.dmp: 0xC0000094: Integer division by zero.

[Copy Details](#)[Exception Settings](#)

100 %

Autos Locals Threads Modules Memory 1 Breakpoints Exception Settings Output Call Stack

Ready

Ln 12

Col 1

Ch 1

INS

Add to Source Control

# Crash dump: Краткий обзор материала

- Назначение
- Создание
- Анализ
  - Без PDB
  - С PDB

# Q&A

# Очень интересная литература



- Tarik Soulami, Inside Windows Debugging: A Practical Guide to Debugging and Tracing Strategies in Windows
- Д.Роббинс. Поиск и устранение ошибок в программах под Windows
- С.Макконнелл. Совершенный код
- Д.Востоков, Memory Dump Analysis Anthology



# CQG

1 800-525-7082  
[www.cqg.com](http://www.cqg.com)

### ***Disclaimer***

*Trading and investment carry a high level of risk, and CQG, Inc. does not make any recommendations for buying or selling any financial instruments. We offer educational information on ways to use our sophisticated CQG trading tools, but it is up to our customers and other readers to make their own trading and investment decisions or to consult with a registered investment advisor.*

© 2019 CQG, Inc. All rights reserved.

CQG®, DOMTrader®, SnapTrader®, TFlow®, TFOBV®, TFOBVO®, TFVOL®, and Data Factory™ are trademarks of CQG, Inc.