Не связывайтесь с поддержкой C++ программистов

Юрий Минаев

О себе

Юрий Минаев

C++ разработчик в компании PVS-Studio

minaev@viva64.com

О чем поговорим

– Доктор, я буду жить?
– А смысл?

Анекдот

- О техподдержке профессиональных разработчиков
- О забавных случаях, которые нам попадались
- Разбавим серьезную атмосферу

О техподдержке

О техподдержке

- Программисты обычно знают, чего хотят...
- ...и часто сами понимают, в чем проблема
- Всегда приятно пообщаться с умным человеком

9 кругов тестирования

Мы тестируем:

- Много
- Часто
- Разнообразно

9 кругов тестирования

1. Тесты документации
2. UI-тесты
3. Юнит-тесты
4. Тесты диагностик
5. Прогон на базе реальных проектов
6. Динамический анализ
7. Статический анализ
8. Тесты мониторинга компиляции
9. Docker-тесты

Перейдем к историям

Не верьте стандартным функциям

```
bool set_value_convert(char_t *&dest, ....,
                       int value)
{
      char buf[128];
      sprintf(buf, "%d", value);

      return set_value_buffer(...., buf);
}
```

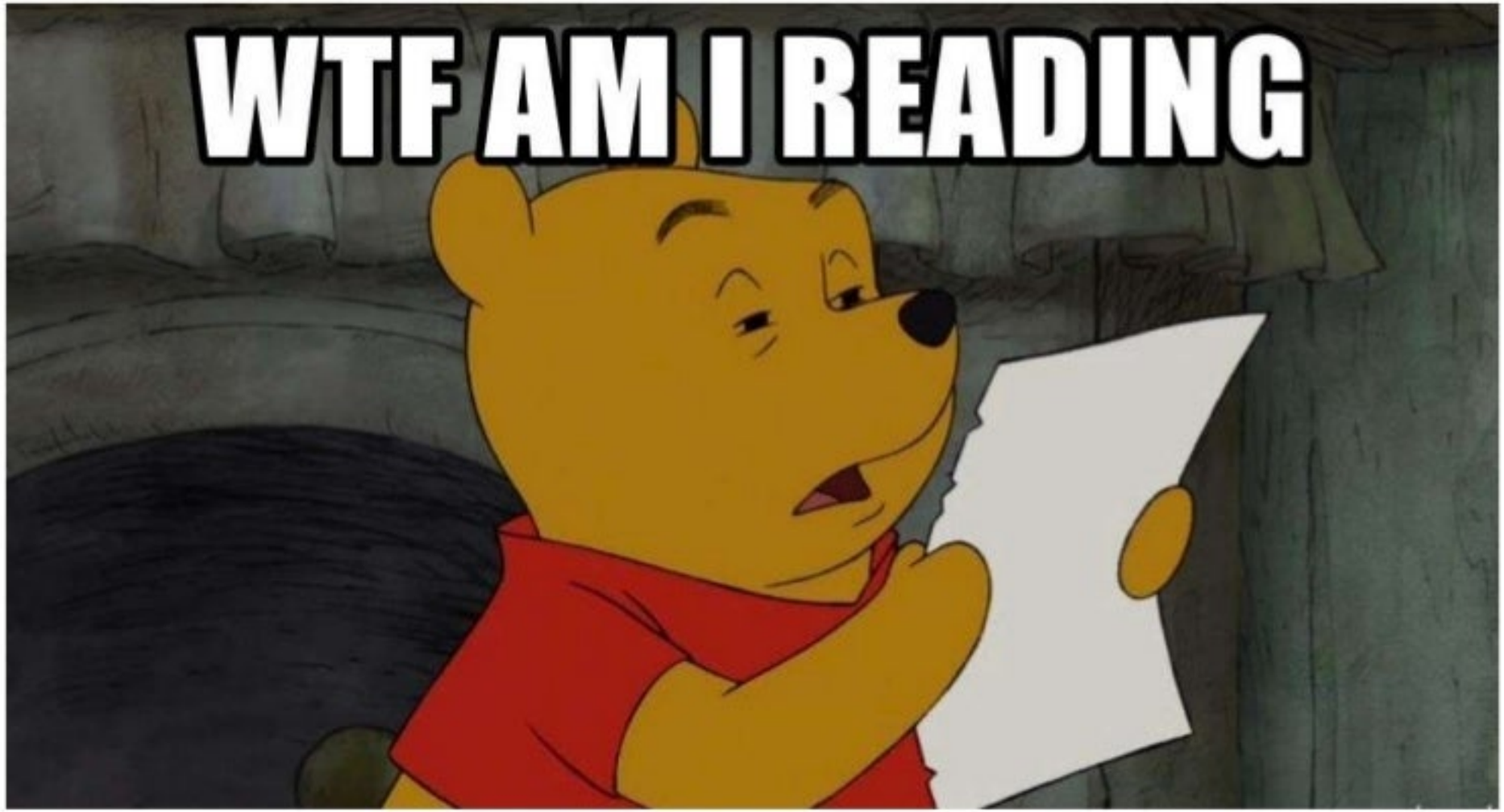
V614 Uninitialized buffer 'buf' used.

Не верьте стандартным функциям

```
bool set_value_convert(char_t *&dest, ....,
                       int value)
{
    char buf[128];
    sprintf(buf, "%d", value);

    return set_value_buffer(...., buf);
}
```

V614 Uninitialized buffer 'buf' used.

Не верьте стандартным функциям



Не верьте стандартным функциям

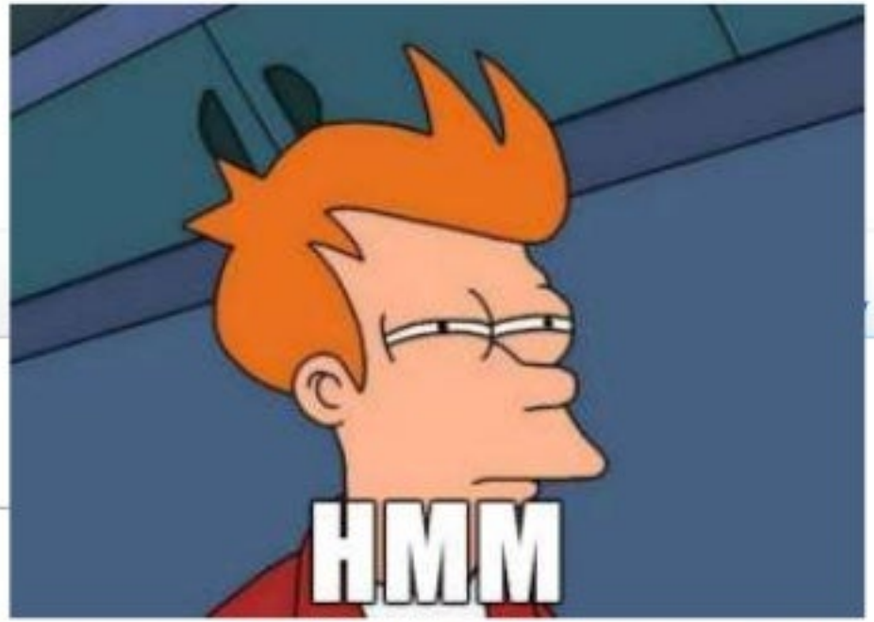
```
#define schar char
#define suchar unsigned schar
#define sprintf std::printf
#define satof atof
#define satoi atoi
```

Еще о стандартных функциях

```
static int EatWhitespace(FILE* InFile)
{
  int c;
  for (c = getc(InFile); isspace(c) && ('\n' != c);
                         c = getc(InFile));

  return (c);
}
```

V560 A part of conditional expression is always true: ('\n' != c).

Еще о стандартных функциях



cppreference.com

Page Discussion

C / Strings library / Null-terminated byte strings

isspace

Defined in header <ctype.h>

```
int isspace( int ch );
```

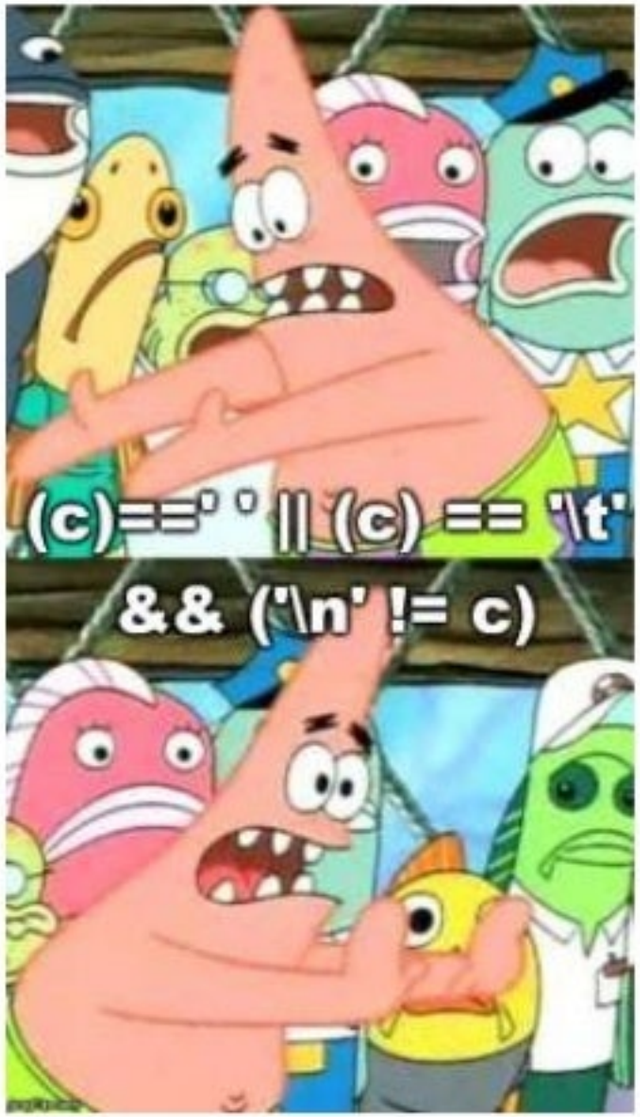
Checks if the given character is a whitespace character, i.e. either space (0x20), form feed (0x0c), line feed (0x0a), carriage return (0x0d), horizontal tab (0x09) or vertical tab (0x0b).

The behavior is undefined if the value of ch is not representable as `unsigned char` and is not equal to `EOF`.

Еще о стандартных функциях

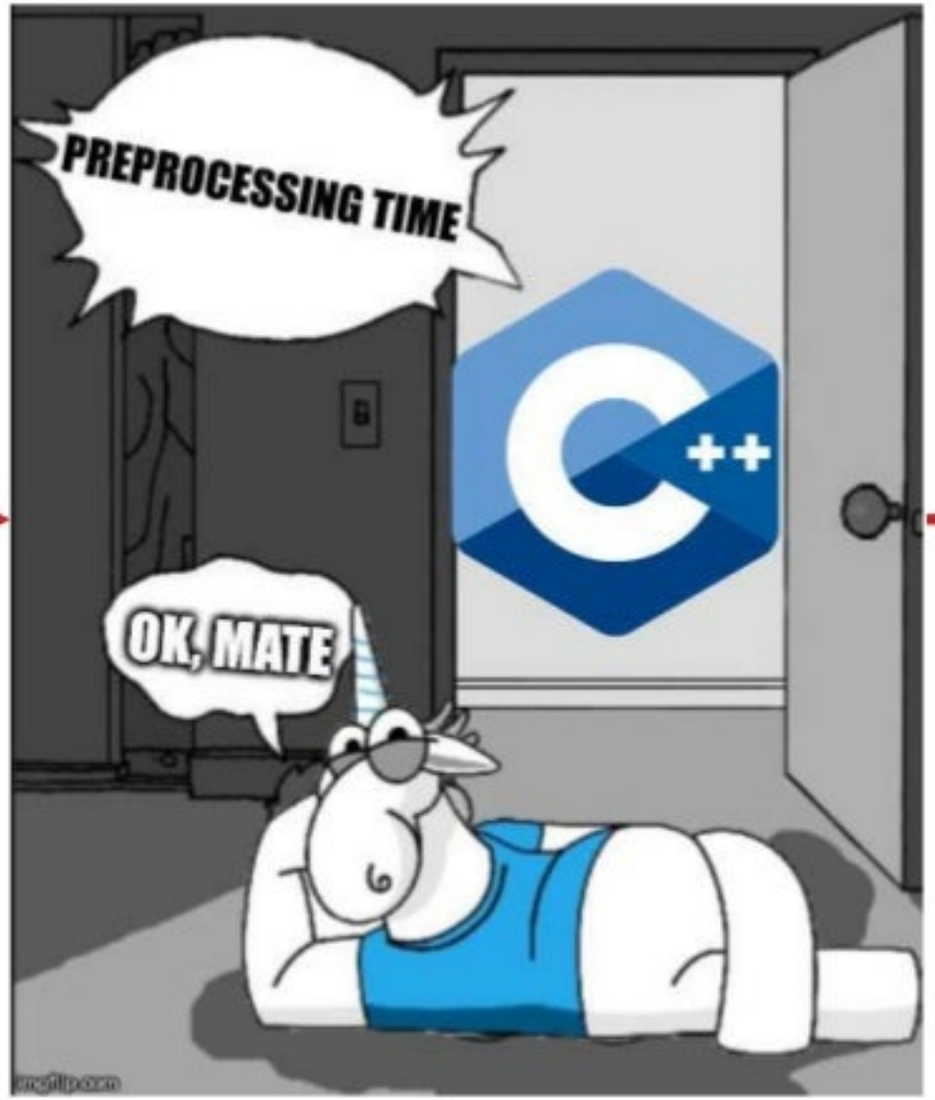
```
#define isspace(c) ((c)==' ' || (c) == '\t')

static int EatWhitespace(FILE* InFile)
{
  int c;
  for (c = getc(InFile);
       ((c)==' ' || (c) == '\t') && ('\n' != c);
       c = getc(InFile));

  return (c);
}
```

Еще о стандартных функциях




```
for (c = getc(InFile);
     ((c)==' ' || (c) == '\t')
         && ('\n' != c);
     c = getc(InFile));
```

Препроцессорные сказки



Препроцессорные сказки

```
void f()
{
  #line 42 "somefile.abc"
  int a = 10 / 0;
}
```

	Code	Message	File	Line
☆	V609	Divide by zero. Denominator '0' == 0.	somefile.abc	42

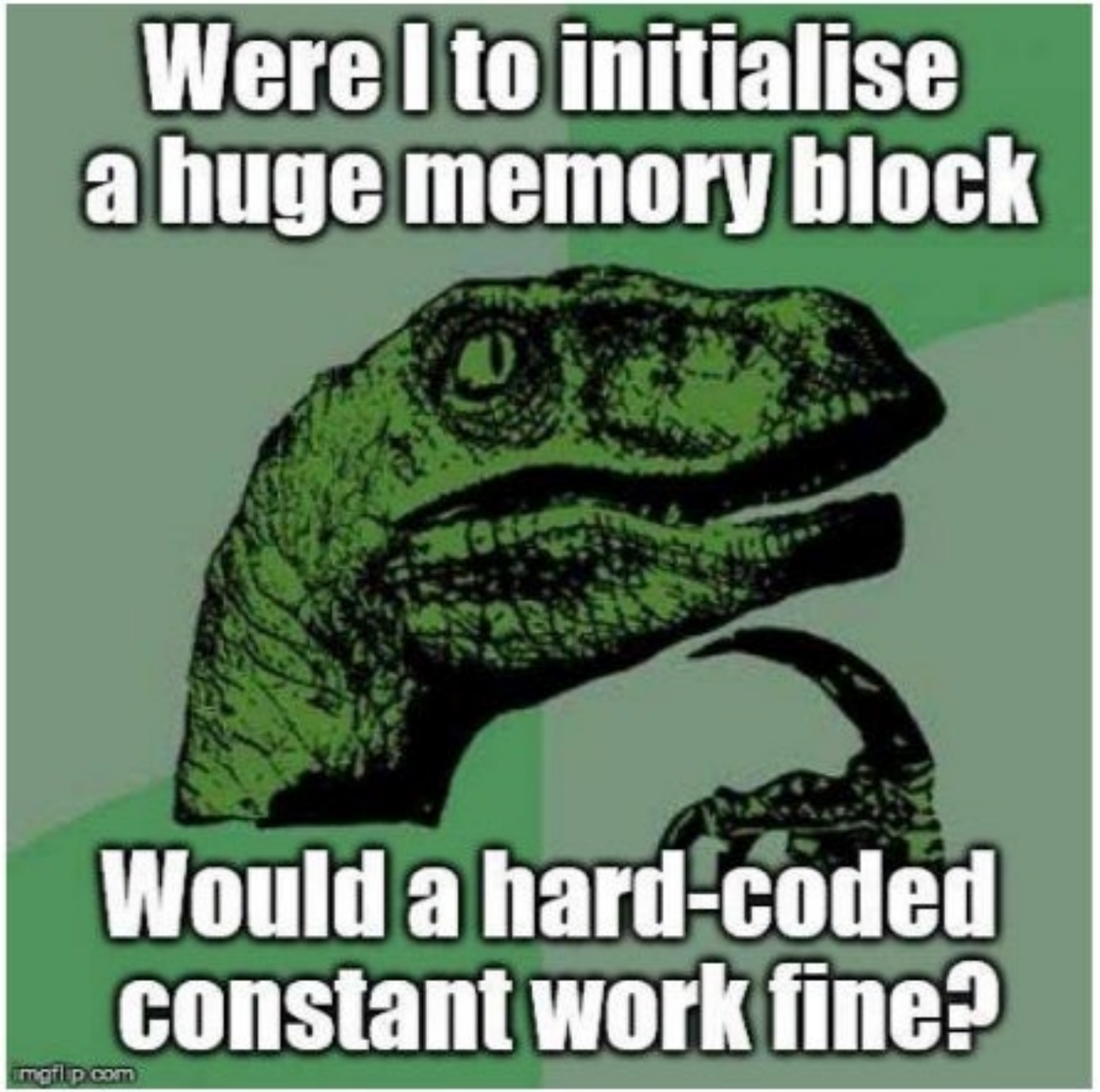
Препроцессорные сказки


```
#line 1 "D:\\Demo\\Source.cpp"
void f()
{
  #line 42 "D:\\Demo\\somefile.abc"
  int a = 10 / 0;
}
```

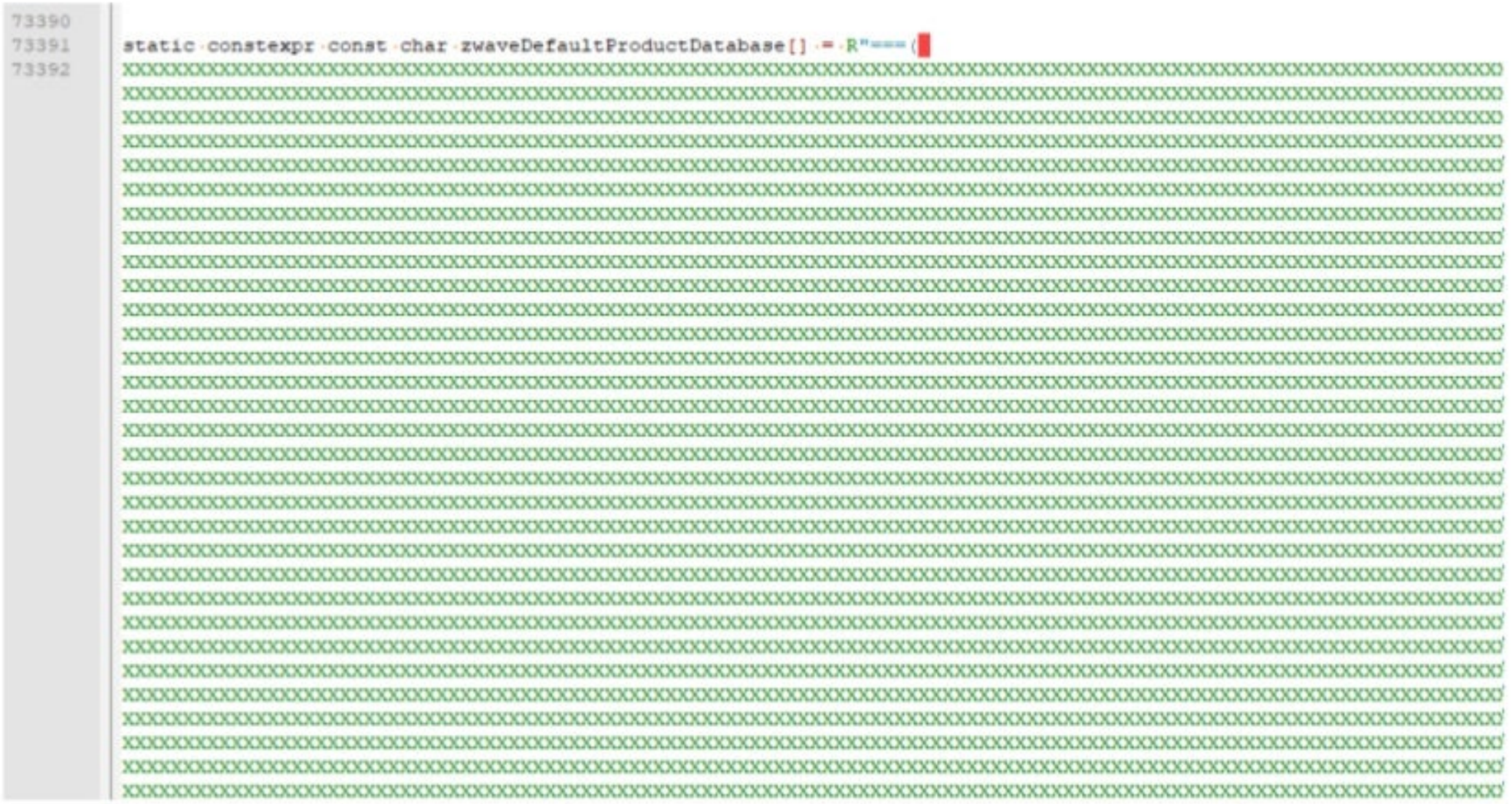
Препроцессорные сказки

	Code	Message	File	Line
☆	V609	Divide by zero. Denominator '0' == 0.	somefile.abc	42
☆	V002	Some diagnostic messages may contain incorrect line number in this file.	Source.cpp	1
☆	V011	Presence of #line directives may cause some diagnostic messages to have incorrect file name and line number.	Source.cpp	3

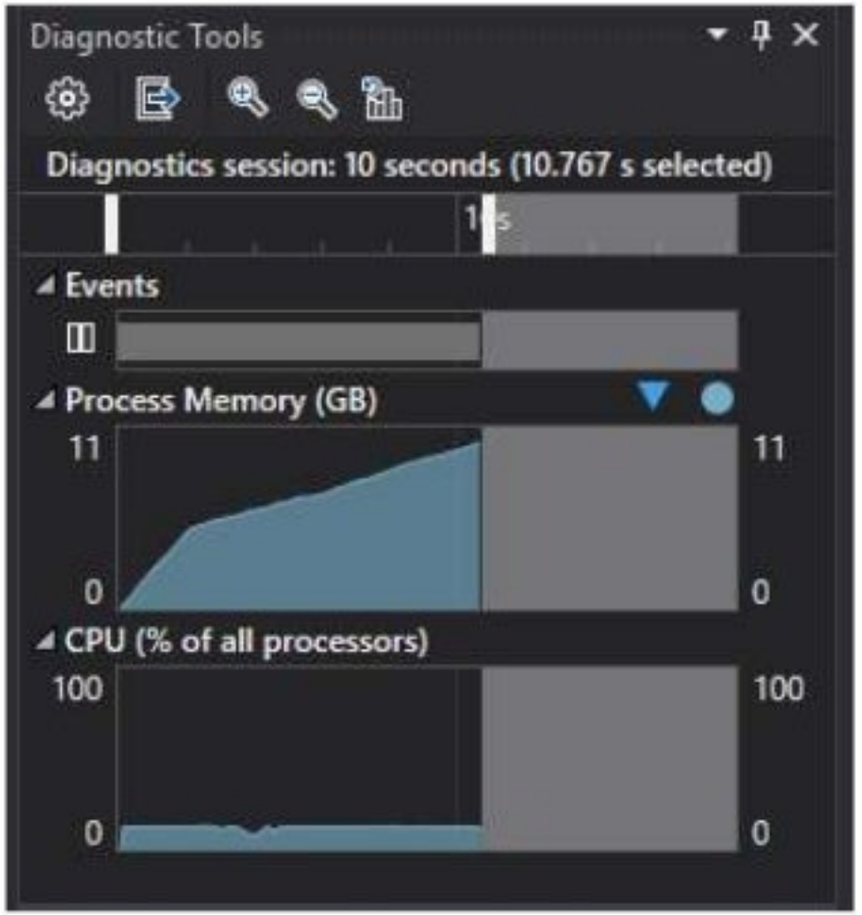
26 мегабайт хватит всем



26 мегабайт хватит всем



26 мегабайт хватит всем



О логике

```
if (ch >= 0x0FF00)
{
        if (!((ch >= 0x0FF10) && (ch <= 0x0FF19)) ||
              ((ch >= 0x0FF21) && (ch <= 0x0FF3A)) ||
              ((ch >= 0x0FF41) && (ch <= 0x0FF5A)))
        {
              //....
        }
}
```

V560 A part of conditional expression is always false: (ch >= 0x0FF21)
V560 A part of conditional expression is always true: (ch <= 0x0FF3A)
V560 A part of conditional expression is always false: (ch >= 0x0FF41)
V560 A part of conditional expression is always true: (ch <= 0x0FF5A)

О логике

О логике

```
if (ch >= 0x0FF00)
{
        if (!((ch >= 0x0FF10) && (ch <= 0x0FF19)) ||
              ((ch >= 0x0FF21) && (ch <= 0x0FF3A)) ||
              ((ch >= 0x0FF41) && (ch <= 0x0FF5A)))
        {
              //....
        }
}
```

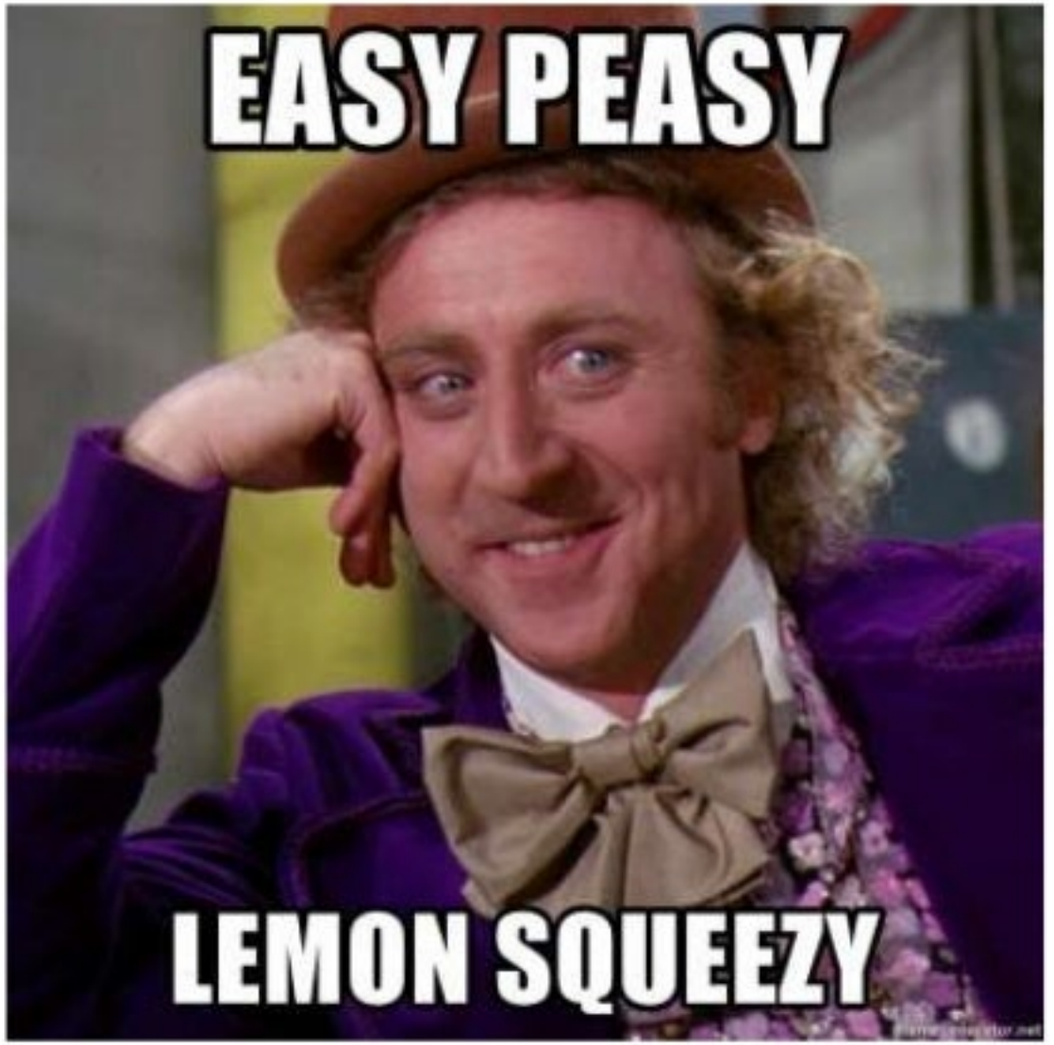
V560 A part of conditional expression is always false: (ch >= 0x0FF21)
V560 A part of conditional expression is always true: (ch <= 0x0FF3A)
V560 A part of conditional expression is always false: (ch >= 0x0FF41)
V560 A part of conditional expression is always true: (ch <= 0x0FF5A)

О логике

```
!(
 ((ch >= 0x0FF10) && (ch <= 0x0FF19)) ||
 ((ch >= 0x0FF21) && (ch <= 0x0FF3A)) ||
 ((ch >= 0x0FF41) && (ch <= 0x0FF5A))
)
```

Инициализация – это просто

```
class Vector
{
public:
   int x, y, z;
   Vector()
   {
      x = 0;
      y = 0;
   }
};
```



Инициализация – это просто

```
class Vector
{
public:
    int x, y, z;
    Vector() : x(0), y(0) {}
};
```

```
class Vector
{
public:
    int x = 0, y = 0, z;
    Vector() {}
};
```

Инициализация – это просто

```
class Vector
{
public:
    int x = 0, y = 0, z;
    Vector() = default;
};
```

```
class Vector
{
public:
    int x, y, z;
    Vector() { Init(); }
    void Init() { x = 0; y = 0; }
};
```

Инициализация – это просто

```
class Vector
{
public:
    int x, y, z;
    Vector() { memset(&x, 0, sizeof(x)); }
};
```

```
class Vector
{
public:
    int x, y, z;
    Vector(int x, int y, int z) :
        x(x), y(y) {}
    Vector() { new (this) Vector(0, 0, 0); }
};
```

Когда роботы устают

Преимущества статического анализа перед code review:

- *Малозатратно*
- *Анализатор не устает (в отличие от человека)*
- *Анализатор знает про ошибочные паттерны, о которых не догадываются программисты*

Когда роботы устают

Преимущества статического анализа перед code review:

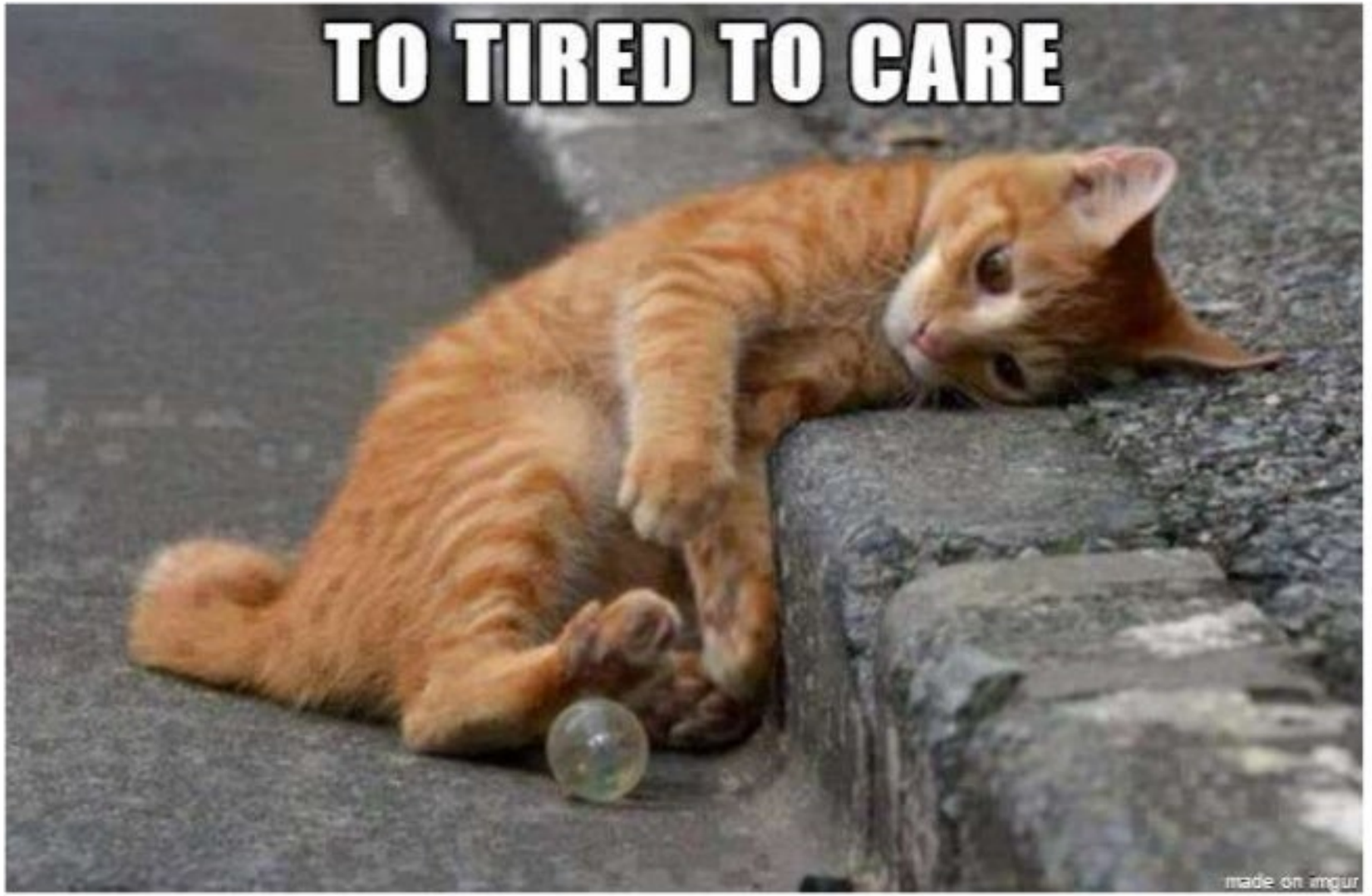
- *Малозатратно*
- *Анализатор не устает (в отличие от человека)*
- *Анализатор знает про ошибочные паттерны, о которых не догадываются программисты*

Когда роботы устают

```
1       #include "libpov.h"
2       int main(int cgc_argc, char *cgc_argv[]) {
3          cgc_negotiate_type1(0x0, 0x0, 0);
4          do {
5             unsigned char *read_00000;
6             unsigned int read_00000_len;
7             unsigned int read_00000_ptr = 0;
8             //***** length read
9             read_00000_len = 2;
10            read_00000 = (unsigned char*)cgc_malloc(read_00000_len);
11            int read_00000_res = cgc_length_read(0, read_00000, read_00000_len);
12            if (read_00000_res) {} //silence unused variable warning
13            //***** read match data
14            static unsigned char match_00000_00000[] =
15               "\x00\x00";
16            read_00000_ptr += cgc_data_match(read_00000 + read_00000_ptr, read_00000_l
                                                                    ...
              cgc_free(write_32776);
819452     } while (0);
819453     do {
819454        unsigned char *read_32776;
819455        unsigned int read_32776_len;
819456        unsigned int read_32776_ptr = 0;
819457        //***** length read
819458        read_32776_len = 2;
819459        read_32776 = (unsigned char*)cgc_malloc(read_32776_len);
819460        int read_32776_res = cgc_length_read(0, read_32776, read_32776_len);
819461        if (read_32776_res) {} //silence unused variable warning
819462        cgc_free(read_32776);
819463        if (read_32776_ptr) {}  //silence unused variable warning if any
819464     } while (0);
819465  }
819466
```

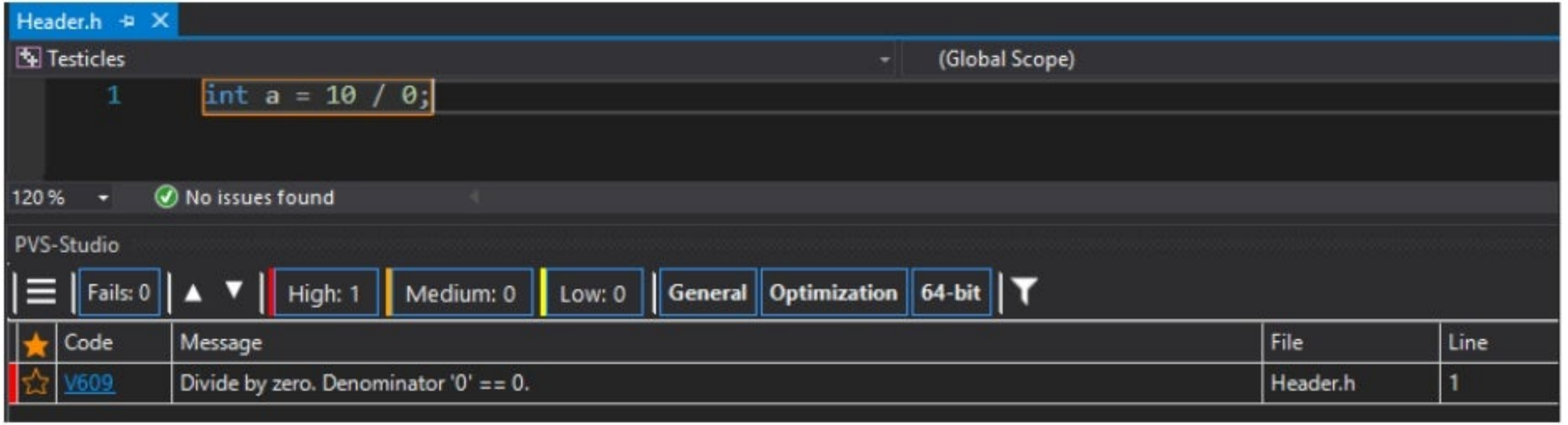
819466

Когда роботы устают



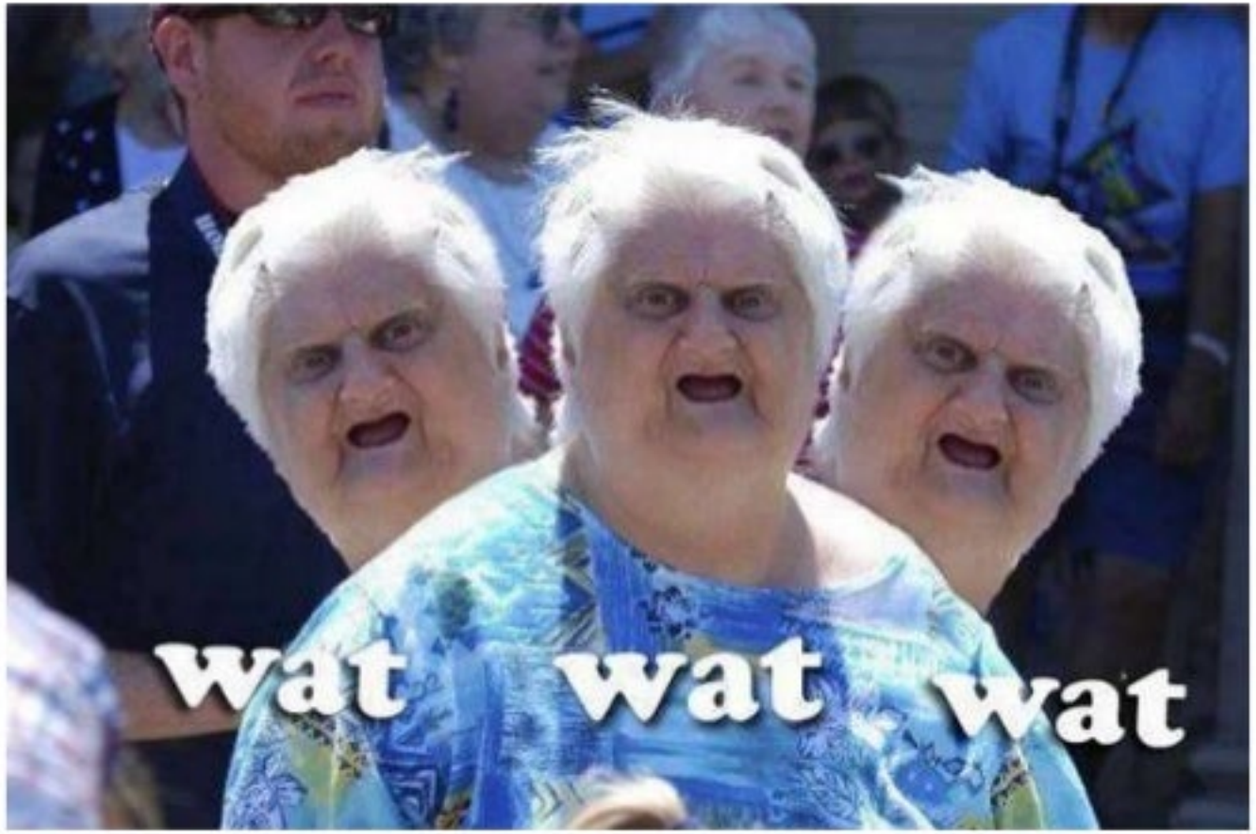
Объезд

Нормальные герои всегда идут в обход ©



Объезд

```
int f()
{
   #include "Header.h"
   return a;
}
```



Объезд

```
static wchar_t* getSectionName(int flag)
{
  switch (flag)
  {
  #include "finger_groups.h"
  }
  return NULL;
}
```

WinAPI is fun

Спросите об этом разработчиков Wine

WinAPI is fun

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

int main()
{
  PACL pDACL = NULL;
  PSECURITY_DESCRIPTOR pSD = NULL;
  ::GetNamedSecurityInfo(_T("ObjectName"), SE_FILE_OBJECT,
          DACL_SECURITY_INFORMATION, NULL, NULL, &pDACL, NULL, &pSD);
  auto test = pDACL == NULL;
  return 0;
}
```

V547 Expression 'pDACL == 0' is always true.

WinAPI is fun

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

int main()
{
  PACL pDACL = NULL;
  PSECURITY_DESCRIPTOR pSD = NULL;
  ::GetNamedSecurityInfo(_T("ObjectName"), SE_FILE_OBJECT,
         DACL_SECURITY_INFORMATION, NULL, NULL, &pDACL, NULL, &pSD);
  auto test = pDACL == NULL;
  return 0;
}
```

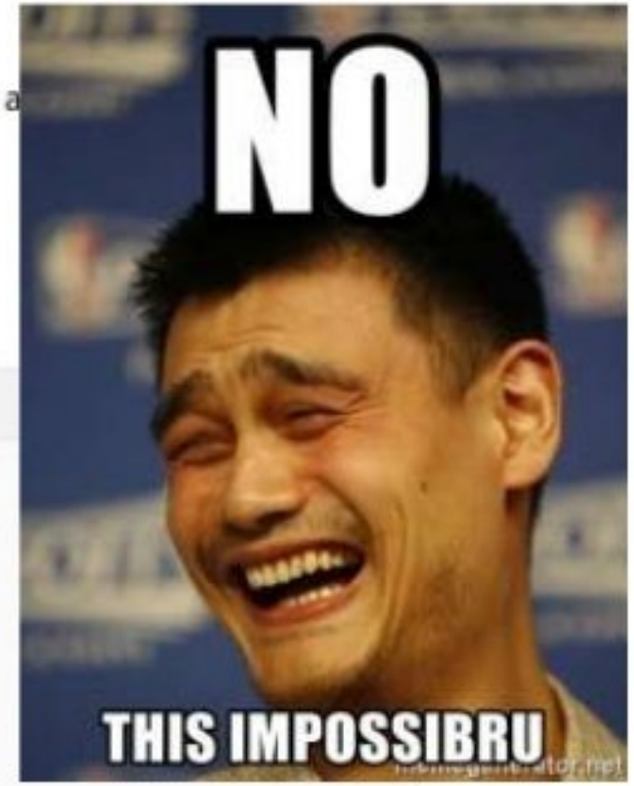
V547 Expression 'pDACL == 0' is always true.

WinAPI is fun

GetNamedSecurityInfoW function

12/05/2018 • 3 minutes to read

The **GetNamedSecurityInfo** function retrieves a copy of the security descriptor for a
name.



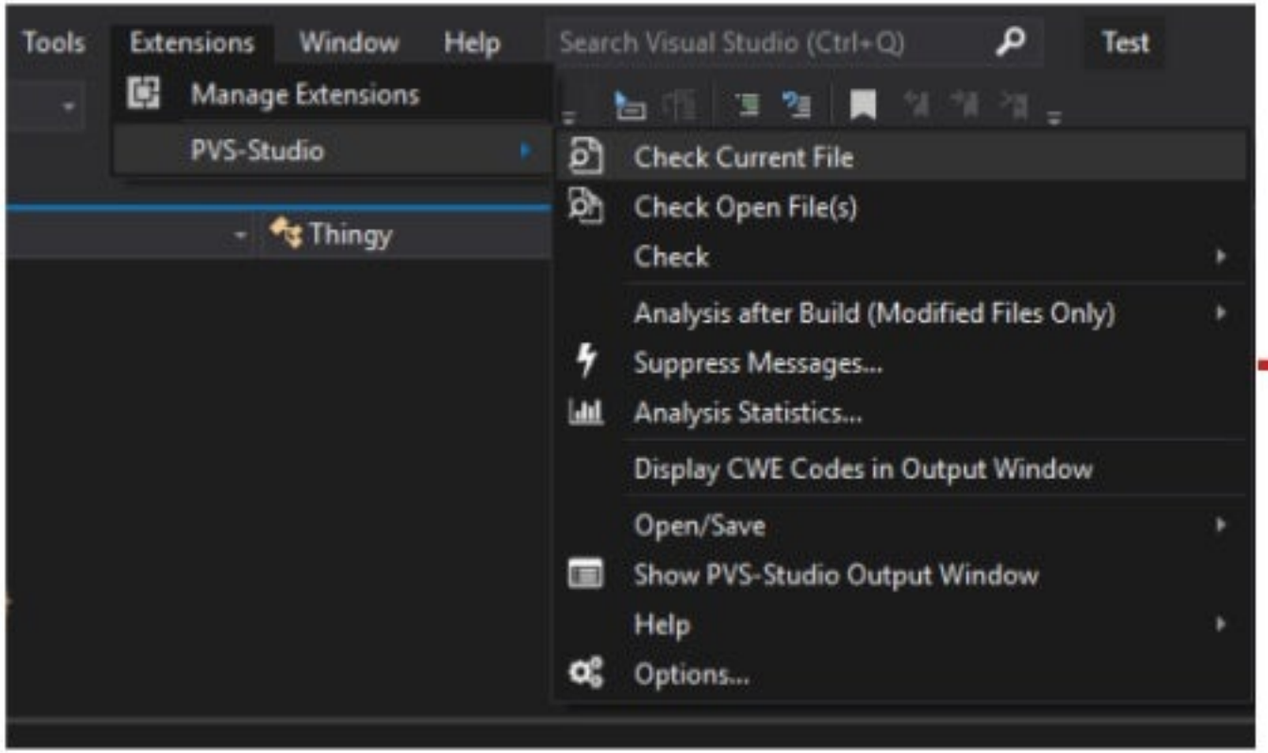
Syntax

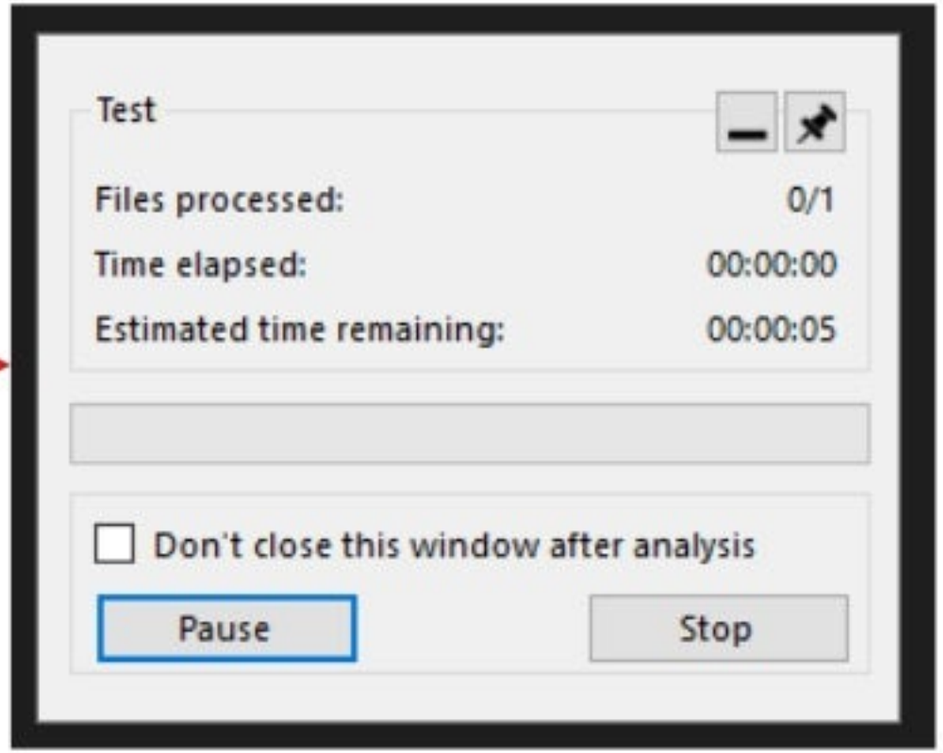
```
DWORD GetNamedSecurityInfoW(
  LPCWSTR                 pObjectName,
  SE_OBJECT_TYPE          ObjectType,
  SECURITY_INFORMATION    SecurityInfo,
  PSID                    *ppsidOwner,
  PSID                    *ppsidGroup,
  PACL                    *ppDacl,
  PACL                    *ppSacl,
  PSECURITY_DESCRIPTOR    *ppSecurityDescriptor
);
```

WinAPI is fun

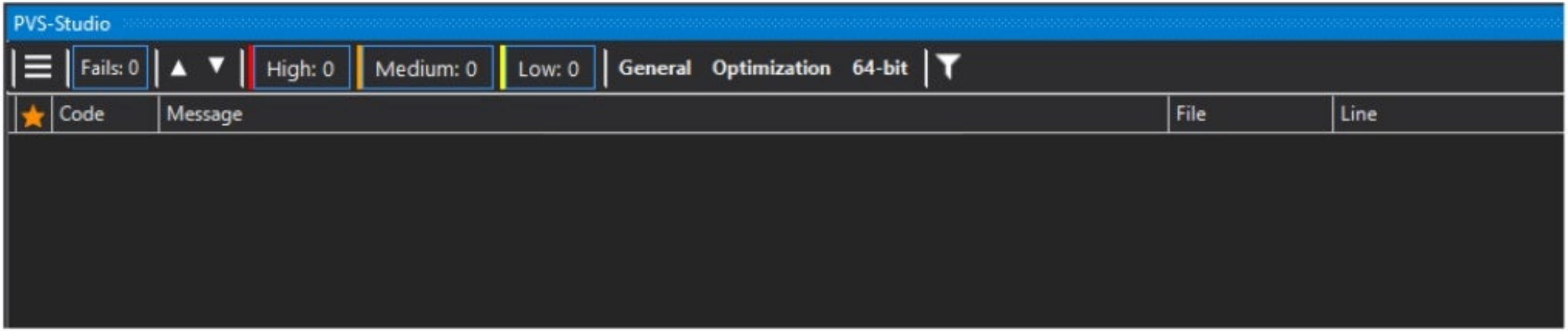
```
__declspec(dllimport)
DWORD
__stdcall
GetNamedSecurityInfoW(
        LPCWSTR                 pObjectName,
        SE_OBJECT_TYPE          ObjectType,
        SECURITY_INFORMATION    SecurityInfo,
        const PSID* ppsidOwner,
        const PSID* ppsidGroup,
        const PACL* ppDacl,
        const PACL* ppSacl,
        PSECURITY_DESCRIPTOR* ppSecurityDescriptor
);
```

Самый страшный баг

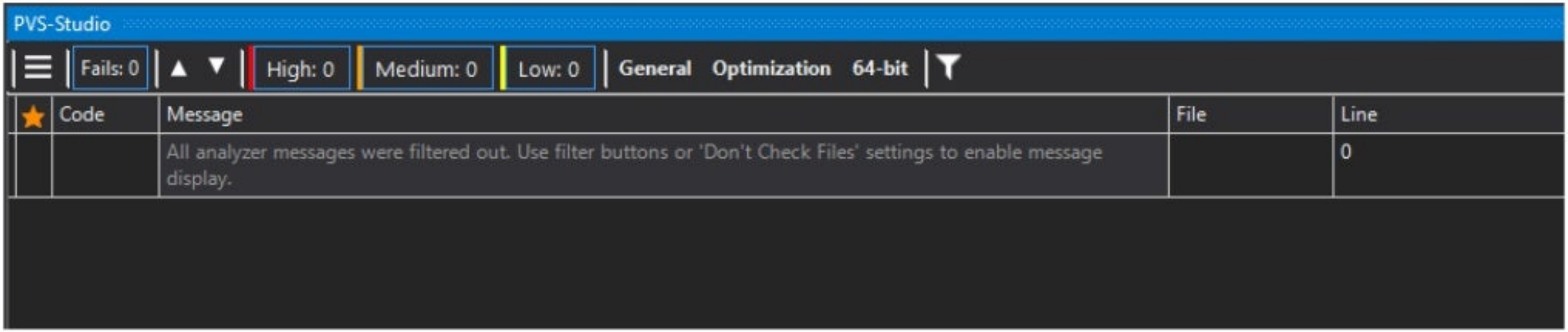




Самый страшный баг

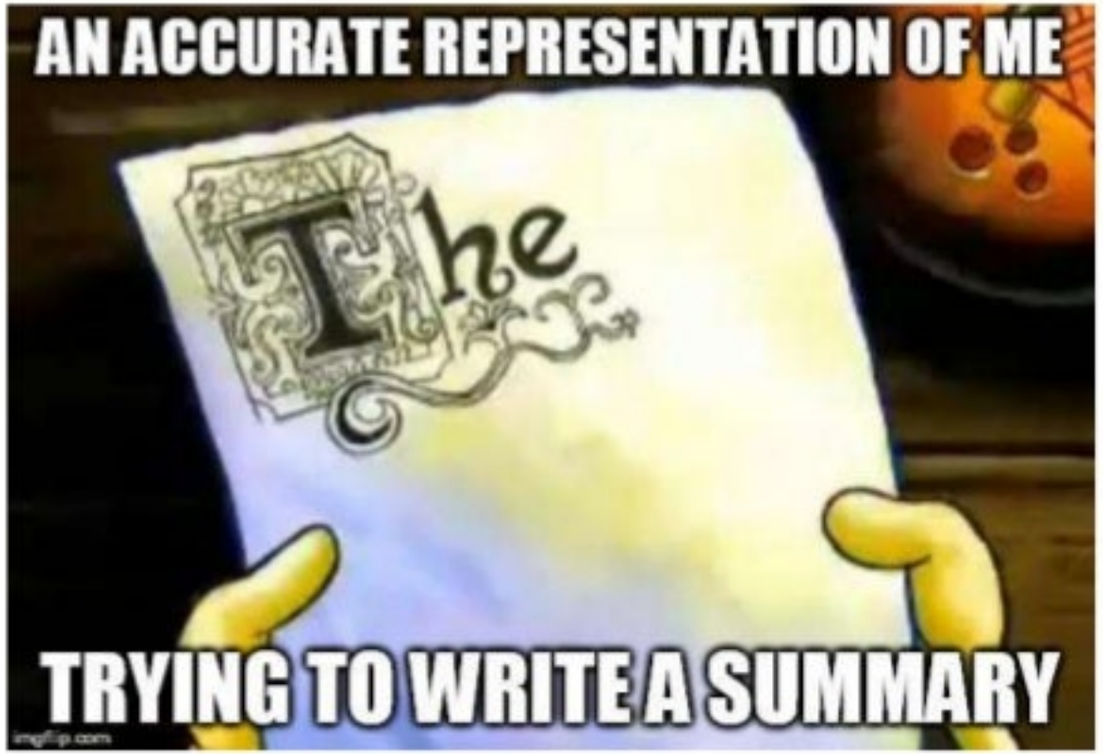


Самый страшный баг



Сухой остаток

- Тесты – это замечательно
- Но они не могут соревноваться с человеческой изобретательностью
- Мы любим наших пользователей и стараемся им помочь
- Но все же, не связывайтесь с поддержкой C++ программистов



Q&A

Юрий Минаев – minaev@viva64.com