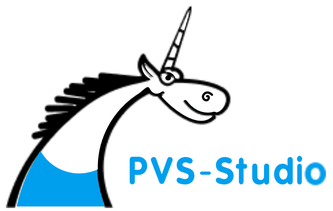

PVS-Studio

Типовые паттерны опечаток в коде и как с ними бороться



Андрей Карпов

CBDO (директор по развитию бизнеса)

Андрей Карпов

CBDO (директор по развитию бизнеса)

- Один из основателей PVS-Studio
- 17 лет в сфере качества и анализа кода
- Хабр: @Andrey2008

Долгое время являлся CTO компании и занимался разработкой C++ ядра анализатора. Основная деятельность на данный момент — управление командами, обучение сотрудников и продвижение методологии статического анализа кода.

@PROGRAMMING_TALES

PVS-Studio

Статический анализатор кода PVS-Studio

- Поиска ошибок и потенциальных уязвимостей в коде программ на языках : C, C++, C#, Java
- Краткое знакомство:

https://vkvideo.ru/video-11805870_456239344

Проверка открытых проектов

- Для популяризации методологии статического анализа кода команда PVS-Studio регулярно публикуем статьи о проверке открытых проектов
- Нашли и сообщили про 16 000+ багов в открытых проектах
- Неудивительно, что я насмотрелся…
- https://pvs-studio.ru/ru/blog/inspections/

Это актуально опытным программистам?

Да

- Заблуждение про опечатки:
  их допускают только студенты, неопытные программисты и т.п.

- Точно могу сказать, их допускают все!

- Даже разработчики выверенных проектов, таких как операционные системы и компиляторы

LLVM. Не там поставлена закрывающая скобка


```
bool isStoreUsed(const FrameIndexEntry &StoreFIE,
                 ExprIterator Candidates,
                 bool IncludeLocalAccesses = true) const;
....
if (SRU.isStoreUsed(*FIE,
                    Prev ? SRU.expr_begin(*Prev) :
                           SRU.expr_begin(BB)),
                    /*IncludeLocalAccesses=*/false) {
```

PVS-Studio: V521 Such expressions using the ',' operator are dangerous. Make sure the expression is correct. ShrinkWrapping.cpp 80

Опечатки: отсутствие throw

Нет throw (Elasticsearch, Java)


```
@Override
public void
setAutoCommit(boolean autoCommit) throws SQLException {
  checkOpen();
  if (autoCommit == false) {
    new SQLFeatureNotSupportedException(
          "Non auto-commit is not supported");
  }
}
```

PVS-Studio: V6006 The object was created but it is not being used. The 'throw' keyword could be missing. JdbcConnection.java 93

Нет throw (ROOT, C++)


```
template <typename Value_t, typename Container_t>
inline RTensor<Value_t, Container_t> RTensor<Value_t, Container_t>::Transpose()
{
  if (fLayout == MemoryLayout::RowMajor) {
    fLayout = MemoryLayout::ColumnMajor;
  } else if (fLayout == MemoryLayout::ColumnMajor) {
    fLayout = MemoryLayout::RowMajor;
  } else {
    std::runtime_error("Memory layout is not known.");
  }
  ....
```

PVS-Studio: V596 The object was created but it is not being used. The 'throw' keyword could be missing: throw runtime_error(FOO); RTensor.hxx 363

Нет throw (Azure PowerShell, C#)


```
private void StartRPITestFailover()
{
  ....
  else
  {
    new ArgumentException(
      Resources
        .UnsupportedDirectionForTFO); // Throw Unsupported Direction
                                      // Exception
  }
```

PVS-Studio: V3006 The object was created but it is not being used. The 'throw' keyword could be missing: throw new ArgumentException(FOO). StartAzureRmRecoveryServicesAsrTestFailoverJob.cs 259

Профилактика

- ~~Unit-тесты~~ — слабый совет с точки зрения практики. Сложно и малопродуктивно покрыть все пути обработки ошибок
- Статический анализ кода
- Теперь знаете про такой антипаттерн и будете аккуратнее при обзоре кода

Опечатки: числа

Люди путают константы

- Например, используют восьмеричные числа.
  В C и C++:
  721
  0721 — восьмеричное значение, равное 465

- Редкое, но случается

Ошибочная константа (Chromium, C++)


```
// Coefficients used to convert from RGB to monochrome.
const uint32 kRedCoefficient = 2125;
const uint32 kGreenCoefficient = 7154;
const uint32 kBlueCoefficient = 0721;
const uint32 kColorCoefficientDenominator = 10000;
```

PVS-Studio: V536 Be advised that the utilized constant value is represented by an octal form. Oct: 0721, Dec: 465. pwg_encoder.cc 24

А вот чего действительно много — это опечаток, связанных с 0, 1 и 2

- Ноль, один, два, Фредди заберёт тебя
  https://pvs-studio.ru/ru/blog/posts/cpp/0713/

Проект XNU kernel, язык C


```
uint32_t gss_krb5_3des_unwrap_mbuf(....) {
  ....
  for (cflag = 1; cflag >= 0; cflag--) {
    *minor = gss_krb5_3des_token_get(....);
    if (*minor == 0)
      break;
    wrap.Seal_Alg[0] = 0xff;
    wrap.Seal_Alg[0] = 0xff;
  }
```

PVS-Studio: V519 The 'wrap.Seal_Alg[0]' variable is assigned values twice successively. Perhaps this is a mistake. Check lines: 2070, 2071. gss_krb5_mech.c 2071

Проект LibreOffice, язык C++


```
Sequence< OUString > FirebirdDriver::
getSupportedServiceNames_Static() throw
                                    (RuntimeException)
{
  Sequence< OUString > aSNS(2);
  aSNS[0] = "com.sun.star.sdbc.Driver";
  aSNS[0] = "com.sun.star.sdbcx.Driver";
  return aSNS;
}
```

PVS-Studio: V519 The 'aSNS[0]' variable is assigned values twice successively. Perhaps this is a mistake. Check lines: 137, 138. driver.cxx 138

Проект Blender, язык C


```
static void initSnapSpatial(TransInfo* t,
  float r_snap[3], float* r_snap_precision)
{
  r_snap[0] = r_snap[1] = 1.0f;
  r_snap[1] = 0.0f;
  *r_snap_precision = 0.1f;
```

PVS-Studio: V519. The 'r_snap[1]' variable is assigned values twice successively. Perhaps this is a mistake. transform.c. Check lines: 1727, 1728.

Проект OpenCOLLADA, язык C++


```
struct short2
{
  short values[2];
  short2(short s1, short s2)
  {
    values[0] = s1;
    values[2] = s2;
  }
  ....
```

PVS-Studio: V557 Array overrun is possible. The '2' index is pointing beyond array bound. mayadmtypes.h 48

Профилактика: инициализация без использование индексов

- Меньше кода — меньше способов ошибиться

```
struct short2
{
  short values[2];
  short2(short s1, short s2) : values { s1, s2 }
  { }
  ....
};
```

Использовать анализаторы кода

- Этот совет ещё не раз будет звучать
- Иногда даже неявно
- Ведь все рассматриваемые здесь ошибки найдены с помощью статического анализатора кода PVS-Studio

Разновидность: ошибки в именах, содержащих 0, 1 и 2

- a0, a1
- l1, l2
- …

Проект Hive, язык Java


```
@Override
public List<ServiceInstance> getAllInstancesOrdered() {
  List<ServiceInstance> list = new LinkedList<>();
  list.addAll(instances.values());
  Collections.sort(list, new Comparator<ServiceInstance>() {
    @Override
      public int compare(ServiceInstance o1, ServiceInstance o2) {
      return o2.getWorkerIdentity().compareTo(o2.getWorkerIdentity());
    }
  });
  return list;
}
```

PVS-Studio: V6009 Function 'compareTo' receives an odd argument. An object 'o2.getWorkerIdentity()' is used as an argument to its own method.
LlapFixedRegistryImpl.java(244)

Проект LLVM, язык C++


```
Instruction *InstCombiner::visitXor(BinaryOperator &I) {
  ....
  if (Op0I && Op1I && Op0I->isShift() &&
      Op0I->getOpcode() == Op1I->getOpcode() &&
      Op0I->getOperand(1) == Op1I->getOperand(1) &&
      (Op1I->hasOneUse() || Op1I->hasOneUse())) {
```

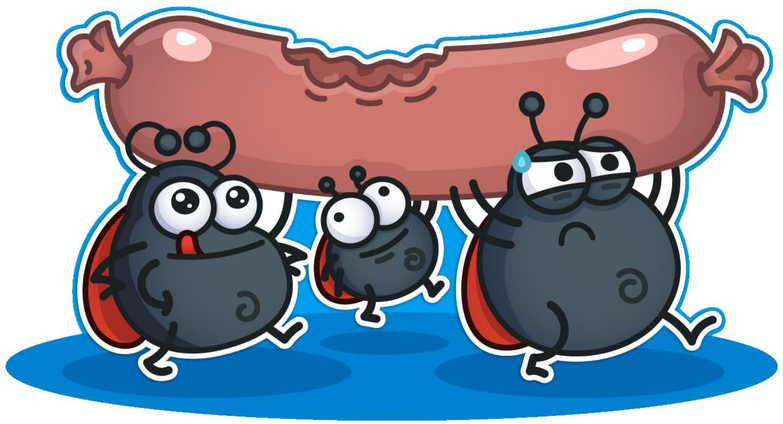
PVS-Studio: V501 There are identical sub-expressions to the left and to the right of the '||' operator: Op1I->hasOneUse () || Op1I->hasOneUse () LLVMInstCombine instcombineandorxor.cpp 2246

Как бороться?

- Иногда может помочь более аккуратное оформление кода

```
public static bool Equals(ref UInt256 a, ref UInt256 b)
{
  return a.s0 == b.s0 && a.s1 == b.s1 && a.s2 == b.s2 && a.s2 == b.s2;
}
```

- И тут мы подходим к теме колбасы ☺



Опечатки: колбаса

Nethermind, язык C#


```
public static bool Equals(ref UInt256 a, ref UInt256 b)
{
  return a.s0 == b.s0 && a.s1 == b.s1 && a.s2 == b.s2 && a.s2 == b.s2;
}
```

PVS-Studio: V3001 There are identical sub-expressions 'a.s2 == b.s2' to the left and to the right of the '&&' operator. Nethermind.Dirichlet.Numerics UInt256.cs 1154

TDengine, язык C


```
static bool rpcNoDelayMsg(tmsg_t msgType) {
  if (msgType == TDMT_VND_FETCH_TTL_EXPIRED_TBS || msgType == TDMT_VND_S3MIGRATE || msgType == TDMT_VND_S3MIGRATE ||
      msgType == TDMT_VND_QUERY_COMPACT_PROGRESS || msgType == TDMT_VND_DROP_TTL_TABLE) {
    return true;
  }
  return false;
}
```

Всматриваться не хочется? Да?

TDengine, язык C


```
pcNoDelayMsg(tmsg_t msgType) {
 == TDMT_VND_FETCH_TTL_EXPIRED_TBS || msgType == TDMT_VND_S3MIGRATE || msgType == TDMT_VND_S3MIGRATE ||
 == TDMT_VND_QUERY_COMPACT_PROGRESS || msgType == TDMT_VND_DROP_TTL_TABLE) {
ue;

e;
```

PVS-Studio: V501 There are identical sub-expressions 'msgType == TDMT_VND_S3MIGRATE' to the left and to the right of the '||' operator. dmTransport.c 398

Как бороться?

- Статический анализ
- Не писать длинные строки
- Красиво оформлять код

- **Форматирование кода таблицей**

Форматирование кода таблицей


```
public static bool Equals(ref UInt256 a, ref UInt256 b)
{
  return a.s0 == b.s0 &&
         a.s1 == b.s1 &&
         a.s2 == b.s2 &&
         a.s2 == b.s2;
}
```

Ошибка лучше заметна

Ещё лучше: логические операторы слева


```
public static bool Equals(ref UInt256 a, ref UInt256 b)
{
  return     a.s0 == b.s0
          && a.s1 == b.s1
          && a.s2 == b.s2
          && a.s2 == b.s2;
}
```

Почему лучше, станет понятно
из следующего примера

Ошибка заметнее, но много пробелов


```
static bool rpcNoDelayMsg(tmsg_t msgType) {
  if (msgType == TDMT_VND_FETCH_TTL_EXPIRED_TBS  ||
      msgType == TDMT_VND_S3MIGRATE              ||
      msgType == TDMT_VND_S3MIGRATE              ||
      msgType == TDMT_VND_QUERY_COMPACT_PROGRESS ||
      msgType == TDMT_VND_DROP_TTL_TABLE) {
    return true;
  }
  return false;
}
```

Всё переформатировать? ;(


```
static bool rpcNoDelayMsg(tmsg_t msgType) {
  if (msgType == TDMT_VND_FETCH_TTL_EXPIRED_TBS  ||
      msgType == TDMT_VND_S3MIGRATE              ||
      msgType == FIX_FIX_FOOOOOOOOOOOOOOOOOOOOOO ||
      msgType == TDMT_VND_QUERY_COMPACT_PROGRESS ||
      msgType == TDMT_VND_DROP_TTL_TABLE) {
    return true;
  }
  return false;
}
```

Так лучше


```
static bool rpcNoDelayMsg(tmsg_t msgType) {
  if (   msgType == TDMT_VND_FETCH_TTL_EXPIRED_TBS
      || msgType == TDMT_VND_S3MIGRATE
      || msgType == TDMT_VND_S3MIGRATE
      || msgType == TDMT_VND_QUERY_COMPACT_PROGRESS
      || msgType == TDMT_VND_DROP_TTL_TABLE) {
    return true;
  }
  return false;
}
```

Код можно сделать ещё изящнее и короче


```
static bool rpcNoDelayMsg(tmsg_t msgType) {
  return     msgType == TDMT_VND_FETCH_TTL_EXPIRED_TBS
          || msgType == TDMT_VND_S3MIGRATE
          || msgType == TDMT_VND_QUERY_COMPACT_PROGRESS
          || msgType == TDMT_VND_DROP_TTL_TABLE);
}
```

Однажды нам в поддержку прислали «ложное срабатывание»


```
522
523 if (ch >= 0x0FF00)
524 {
525     if (!((ch >= 0x0FF10) && (ch <= 0x0FF19)) || ((ch >= 0x0FF21) && (ch <= 0x0FF3A)) || ((ch >= 0x0FF41) && (ch <= 0x0FF5A)))
526     {
527         if (j == 0)
528             continue;
529         ch = chx;
530     }
531 }
```

☆	V560	A part of conditional expression is always false: (ch >= 0x0FF21).	decodew.cpp	525
☆	V560	A part of conditional expression is always true: (ch <= 0x0FF3A).	decodew.cpp	525
☆	V560	A part of conditional expression is always false: (ch >= 0x0FF41).	decodew.cpp	525
☆	V560	A part of conditional expression is always true: (ch <= 0x0FF5A).	decodew.cpp	525

Давайте проведём рефакторинг кода


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

Логический оператор *НЕ* (!) применяется только к первому подвыражению.

Не хватает ещё одних скобочек :-/

Ещё больше скобок — это путь не туда. Лучше так:


```
const bool isLetterOrDigit =    (ch >= 0x0FF10 && ch <= 0x0FF19)  // 0..9
                             || (ch >= 0x0FF21 && ch <= 0x0FF3A)  // A..Z
                             || (ch >= 0x0FF41 && ch <= 0x0FF5A); // a..z
if (!isLetterOrDigit)
```

Обратите внимание, что я убрал часть скобок. Как мы только что видели, большое количество скобок вовсе не помогло избежать ошибки.

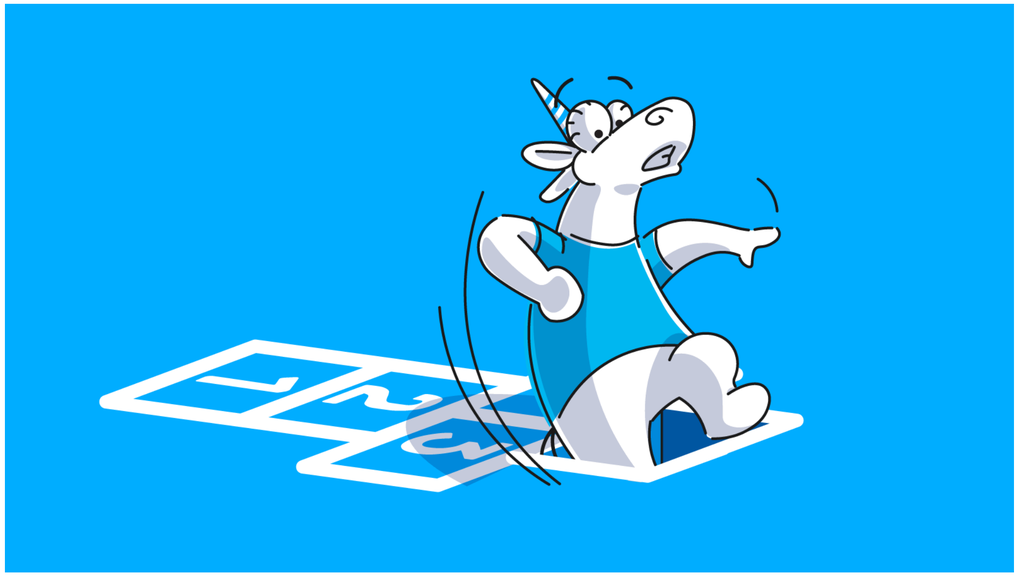
Форматирование кода таблицей

- Подробнее про эту тему
  https://pvs-studio.ru/ru/blog/terms/7003/

Эффект последней строки

Описал этот эффект ещё в 2014 году

- Отсылка к альпинистам
- Люди чаще всего допускают ошибку в последней строке однотипного кода – публикация



Проект Godot Engine, язык C++


```
String SoftBody::get_configuration_warning() const {
  ....
  Transform t = get_transform();
  if ((ABS(t.basis.get_axis(0).length() - 1.0) > 0.05 ||
       ABS(t.basis.get_axis(1).length() - 1.0) > 0.05 ||
       ABS(t.basis.get_axis(0).length() - 1.0) > 0.05)) {
    if (!warning.empty())
```

PVS-Studio: V501 CWE-570 There are identical sub-expressions to the left and to the right of the '||' operator. soft_body.cpp 399

Проект Elasticsearch, язык Java


```
for (int i = 0; i < values.length; i++) {
  if (values[i] == null) continue;
  if (values[i] instanceof String) continue;
  if (values[i] instanceof Text) continue;
  if (values[i] instanceof Long) continue;
  if (values[i] instanceof Integer) continue;
  if (values[i] instanceof Short) continue;
  if (values[i] instanceof Byte) continue;
  if (values[i] instanceof Double) continue;
  if (values[i] instanceof Float) continue;
  if (values[i] instanceof Boolean) continue;
  if (values[i] instanceof Boolean) continue;
  throw new IllegalArgumentException(....);
}
```

PVS-Studio: V6039 There are two 'if' statements with identical conditional expressions. The first 'if' statement contains method return. This means that the second 'if' statement is senseless. SearchAfterBuilder.java(94), SearchAfterBuilder.java(93)

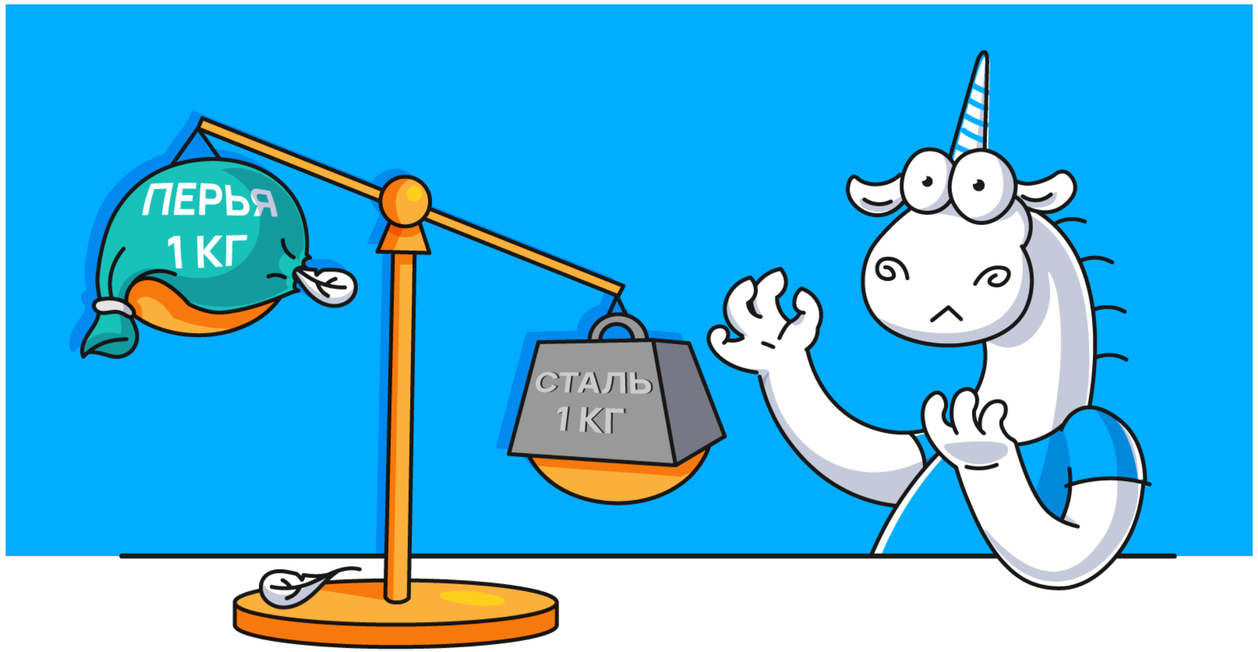
Как бороться?

- Знать про этот антипаттерн и внимательнее проверять последние строки во время обзоров кода
- По возможности превращать однородные блоки кода в вызовы функции
- Использовать PVS-Studio, который хорошо выявляет опечатки разнообразнейших видов

Опечатки в функциях сравнения

Зло живёт в функциях сравнения (2017)

- Функции сравнения просты и кажется, что нет смысла их внимательно проверять
- Функции сравнения скучно проверять



https://pvs-studio.ru/ru/blog/posts/cpp/0509/

Проект IronPython, язык C#


```
public static int Compare(SourceLocation left,
                          SourceLocation right) {
  if (left < right) return -1;
  if (right > left) return 1;
  return 0;
}
```

PVS-Studio: V3021 There are two 'if' statements with identical conditional expressions. The first 'if' statement contains method return. This means that the second 'if' statement is senseless. SourceLocation.cs 156

Проект Serious Engine, язык C++


```
inline BOOL IsEqual(CTexParams tp) {
  return tp_iFilter      == tp.tp_iFilter &&
         tp_iAnisotropy  == tp_iAnisotropy &&
         tp_eWrapU       == tp.tp_eWrapU &&
         tp_eWrapV       == tp.tp_eWrapV;
};
```

PVS-Studio: V501 There are identical sub-expressions to the left and to the right of the '==' operator: tp_iAnisotropy == tp_iAnisotropy gfx_wrapper.h 180

Проект CryEngine, язык C++


```
bool
operator==(const SComputePipelineStateDescription& other) const
{
  return 0 == memcmp(this, &other, sizeof(this));
}
```

PVS-Studio: V579 The memcmp function receives the pointer and its size as arguments. It is possibly a mistake. Inspect the third argument. graphicspipelinestateset.h 58

Как бороться?

- Знать про этот антипаттерн и не лениться на обзорах кода проверять функции сравнения
- Статический анализ кода

- А точно ли нужен оператор сравнения? Бывает, что иногда он написан зря, так как его может сгенерировать компилятор

Неправильное использование unique_ptr


```
std::unique_ptr<OptionDefinition> m_options_definition_up;

Status SetOptionsFromArray(StructuredData::Dictionary& options) {
  Status error;
  m_num_options = options.GetSize();
  m_options_definition_up.reset(new OptionDefinition[m_num_options]);
  ....
}
```

PVS-Studio: V554 Incorrect use of unique_ptr. The memory allocated with 'new []' will be cleaned using 'delete'. CommandObjectCommands.cpp 1384

Опечатка: пропустили []


```
// Правильно:
std::unique_ptr<OptionDefinition []> m_options_definition_up;
```

Это именно опечатка, а не ошибка по незнанию. Думаю, авторы проекта LLVM знают, как правильно использовать *unique_ptr*.

Однако и они не защищены от опечаток, так как баг найден нами в LLVM 19.

Опечатки в юнит-тестах

Один пример (XMage, Java)


```
@Test
public void test_Simple_LongGame() {
    ....
    addCard(Zone.LIBRARY, playerA, "Mountain", 10);
    addCard(Zone.LIBRARY, playerA, "Forest", 10);
    addCard(Zone.LIBRARY, playerA, "Lightning Bolt", 20);
    addCard(Zone.LIBRARY, playerA, "Balduvian Bears", 10);
    //
    addCard(Zone.LIBRARY, playerB, "Mountain", 10);
    addCard(Zone.LIBRARY, playerA, "Forest", 10);
    addCard(Zone.LIBRARY, playerB, "Lightning Bolt", 20);
    addCard(Zone.LIBRARY, playerB, "Balduvian Bears", 10);
    ....
```

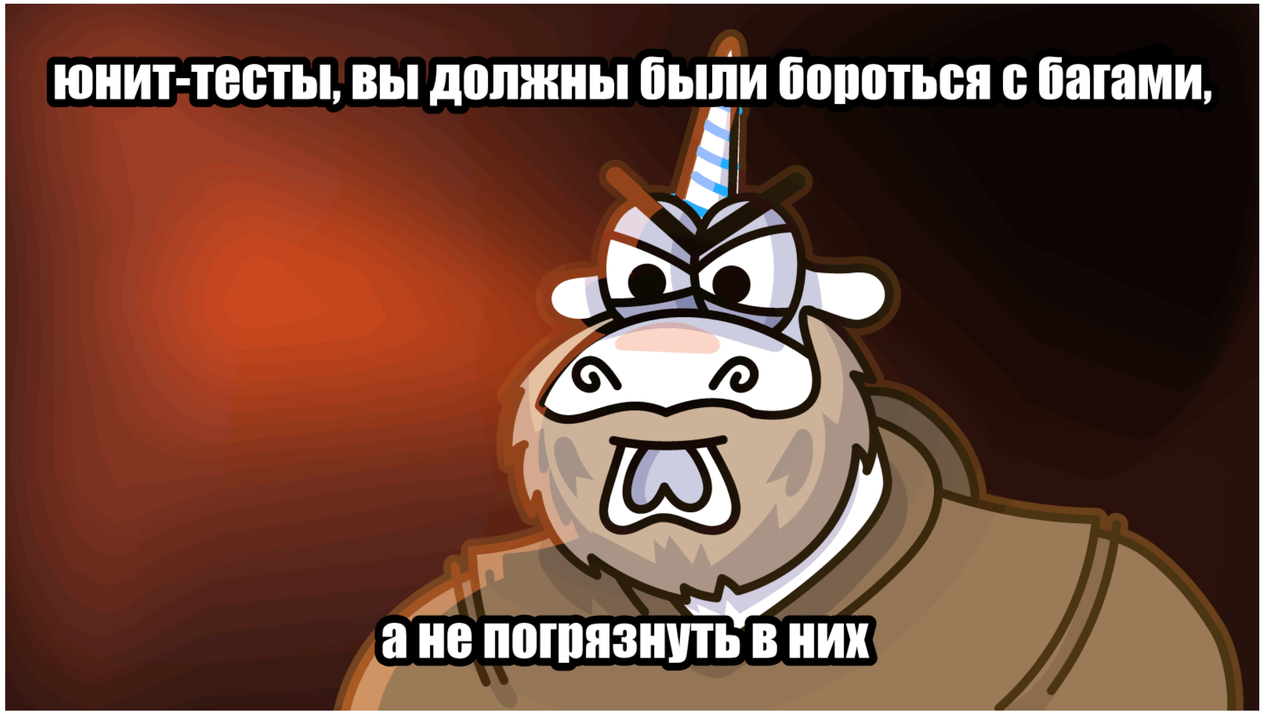
PVS-Studio: V6072 Two similar code fragments were found. Perhaps, this is a typo and 'playerB' variable should be used instead of 'playerA'. SubTypeChangingEffectsTest.java(162), ...

Профилактика опечаток в юнит-тестах

- Опечатки в юнит-тестах такие же, как и в других местах
- Поэтому и рекомендации такие же, как мы уже рассмотрели
- Почему они выделены в отдельный случай?
- Потому что ошибок в тестах много!
- Тесты не тестируют ☺
- Тесты менее внимательно смотрят на обзорах кода

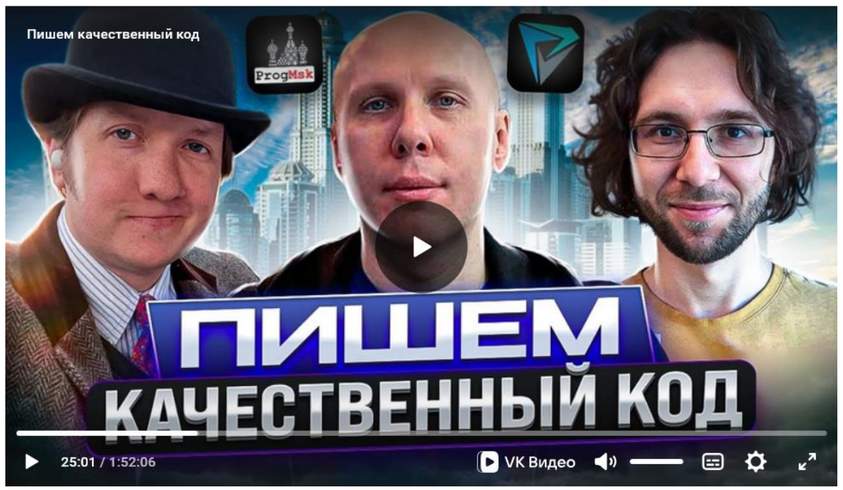
Опечатки в юнит-тестах

- Про ошибки в юнит-тестах: https://pvs-studio.ru/ru/blog/posts/cpp/1179/
- Большинство сводится к опечаткам



Профилактика опечаток в юнит-тестах

- Предупреждён — значит вооружён
- Прочитайте приведённую ранее статью
- Посмотрите доклад «Как статический анализ дополняет TDD» https://pvs-studio.ru/ru/blog/video/10963/
- **Станете внимательнее относиться к коду в тестах. Profit!**



Жемчужина в коллекции

PVS-Studio

Самая красивая ошибка 2024 года (проект DPDK)

- Публикация: https://pvs-studio.ru/ru/blog/posts/cpp/1176/
- Есть метод, который превращает именованные константы из `enum dbg_status` в строки
- Взглянем сначала на `enum` и массив строк

```
enum dbg_status {
  DBG_STATUS_OK,
  DBG_STATUS_APP_VERSION_NOT_SET,
  DBG_STATUS_UNSUPPORTED_APP_VERSION,
  DBG_STATUS_DBG_BLOCK_NOT_RESET,
  DBG_STATUS_INVALID_ARGS,
  DBG_STATUS_OUTPUT_ALREADY_SET,
  DBG_STATUS_INVALID_PCI_BUF_SIZE,
  DBG_STATUS_PCI_BUF_ALLOC_FAILED,
  DBG_STATUS_PCI_BUF_NOT_ALLOCATED,
  DBG_STATUS_INVALID_FILTER_TRIGGER_DWORDS,
  DBG_STATUS_NO_MATCHING_FRAMING_MODE,
  DBG_STATUS_VFC_READ_ERROR,
  DBG_STATUS_STORM_ALREADY_ENABLED,
  DBG_STATUS_STORM_NOT_ENABLED,
  DBG_STATUS_BLOCK_ALREADY_ENABLED,
  DBG_STATUS_BLOCK_NOT_ENABLED,
  DBG_STATUS_NO_INPUT_ENABLED,
  DBG_STATUS_NO_FILTER_TRIGGER_256B,
  DBG_STATUS_FILTER_ALREADY_ENABLED,
  DBG_STATUS_TRIGGER_ALREADY_ENABLED,
  DBG_STATUS_TRIGGER_NOT_ENABLED,
  DBG_STATUS_CANT_ADD_CONSTRAINT,
  DBG_STATUS_TOO_MANY_TRIGGER_STATES,
  DBG_STATUS_TOO_MANY_CONSTRAINTS,
  DBG_STATUS_RECORDING_NOT_STARTED,
  DBG_STATUS_DATA_DIDNT_TRIGGER,
  DBG_STATUS_NO_DATA_RECORDED,
  DBG_STATUS_DUMP_BUF_TOO_SMALL,
  DBG_STATUS_DUMP_NOT_CHUNK_ALIGNED,
  DBG_STATUS_UNKNOWN_CHIP,
  DBG_STATUS_VIRT_MEM_ALLOC_FAILED,
  DBG_STATUS_BLOCK_IN_RESET,
  DBG_STATUS_INVALID_TRACE_SIGNATURE,
  DBG_STATUS_INVALID_NVRAM_BUNDLE,
  DBG_STATUS_NVRAM_GET_IMAGE_FAILED,
  DBG_STATUS_NON_ALIGNED_NVRAM_IMAGE,
  DBG_STATUS_NVRAM_READ_FAILED,
  DBG_STATUS_IDLE_CHK_PARSE_FAILED,
  DBG_STATUS_MCP_TRACE_BAD_DATA,
  DBG_STATUS_MCP_TRACE_NO_META,
  DBG_STATUS_MCP_COULD_NOT_HALT,
  DBG_STATUS_MCP_COULD_NOT_RESUME,
  DBG_STATUS_RESERVED0,
  DBG_STATUS_SEMI_FIFO_NOT_EMPTY,
  DBG_STATUS_IGU_FIFO_BAD_DATA,
  DBG_STATUS_MCP_COULD_NOT_MASK_PRTY,
  DBG_STATUS_FW_ASSERTS_PARSE_FAILED,
  DBG_STATUS_REG_FIFO_BAD_DATA,
  DBG_STATUS_PROTECTION_OVERRIDE_BAD_DATA,
  DBG_STATUS_DBG_ARRAY_NOT_SET,
  DBG_STATUS_RESERVED1,
  DBG_STATUS_NON_MATCHING_LINES,
  DBG_STATUS_INSUFFICIENT_HW_IDS,
  DBG_STATUS_DBG_BUS_IN_USE,
  DBG_STATUS_INVALID_STORM_DBG_MODE,
  DBG_STATUS_OTHER_ENGINE_BB_ONLY,
  DBG_STATUS_FILTER_SINGLE_HW_ID,
  DBG_STATUS_TRIGGER_SINGLE_HW_ID,
  DBG_STATUS_MISSING_TRIGGER_STATE_STORM,
  MAX_DBG_STATUS
};
```

Длинное перечисление именованных констант. Даже всматриваться при обзоре кода не хочется.

```
static const char* const s_status_str[] = {
  /* DBG_STATUS_OK */
  "Operation completed successfully",

  /* DBG_STATUS_APP_VERSION_NOT_SET */
  "Debug application version wasn't set",

  /* DBG_STATUS_UNSUPPORTED_APP_VERSION */
  "Unsupported debug application version",

  /* DBG_STATUS_DBG_BLOCK_NOT_RESET */
  "The debug block wasn't reset since the last recording",

  /* DBG_STATUS_INVALID_ARGS */
  "Invalid arguments",

  /* DBG_STATUS_OUTPUT_ALREADY_SET */
  "The debug output was already set",

  /* DBG_STATUS_INVALID_PCI_BUF_SIZE */
  "Invalid PCI buffer size",

  /* DBG_STATUS_PCI_BUF_ALLOC_FAILED */
  "PCI buffer allocation failed",

  /* DBG_STATUS_PCI_BUF_NOT_ALLOCATED */
  "A PCI buffer wasn't allocated",

  /* DBG_STATUS_INVALID_FILTER_TRIGGER_DWORDS */
  "The filter/trigger constraint dword offsets are not "
  "enabled for recording",


  /* DBG_STATUS_VFC_READ_ERROR */
  "Error reading from VFC",

  /* DBG_STATUS_STORM_ALREADY_ENABLED */
  "The Storm was already enabled",

  /* DBG_STATUS_STORM_NOT_ENABLED */
  "The specified Storm wasn't enabled",

  /* DBG_STATUS_BLOCK_ALREADY_ENABLED */
  "The block was already enabled",

  /* DBG_STATUS_BLOCK_NOT_ENABLED */
  "The specified block wasn't enabled",

  /* DBG_STATUS_NO_INPUT_ENABLED */
  "No input was enabled for recording",

  /* DBG_STATUS_NO_FILTER_TRIGGER_256B */
  "Filters and triggers are not allowed in E4 256-bit
mode",

  /* DBG_STATUS_FILTER_ALREADY_ENABLED */
  "The filter was already enabled",

  /* DBG_STATUS_TRIGGER_ALREADY_ENABLED */
  "The trigger was already enabled",

  /* DBG_STATUS_TRIGGER_NOT_ENABLED */
  "The trigger wasn't enabled",

  /* DBG_STATUS_CANT_ADD_CONSTRAINT */
  "A constraint can be added only after a filter was "
  "enabled or a trigger state was added",

  /* DBG_STATUS_TOO_MANY_TRIGGER_STATES */
  "Cannot add more than 3 trigger states",

  /* DBG_STATUS_TOO_MANY_CONSTRAINTS */
  "Cannot add more than 4 constraints per filter or trigger
state",

  /* DBG_STATUS_RECORDING_NOT_STARTED */
  "The recording wasn't started",

  /* DBG_STATUS_DATA_DID_NOT_TRIGGER */
  "A trigger was configured, but it didn't trigger",

  /* DBG_STATUS_NO_DATA_RECORDED */
  "No data was recorded",

  /* DBG_STATUS_DUMP_BUF_TOO_SMALL */
  "Dump buffer is too small",

  /* DBG_STATUS_DUMP_NOT_CHUNK_ALIGNED */
  "Dumped data is not aligned to chunks",

  /* DBG_STATUS_UNKNOWN_CHIP */
  "Unknown chip",

  /* DBG_STATUS_VIRT_MEM_ALLOC_FAILED */
  "Failed allocating virtual memory",

  /* DBG_STATUS_BLOCK_IN_RESET */
  "The input block is in reset",

  /* DBG_STATUS_INVALID_TRACE_SIGNATURE */
  "Invalid MCP trace signature found in NVRAM",

  /* DBG_STATUS_INVALID_NVRAM_BUNDLE */
  "Invalid bundle ID found in NVRAM",

  /* DBG_STATUS_NVRAM_GET_IMAGE_FAILED */
  "Failed getting NVRAM image",

  /* DBG_STATUS_NON_ALIGNED_NVRAM_IMAGE */
  "NVRAM image is not dword-aligned",

  /* DBG_STATUS_NVRAM_READ_FAILED */
  "Failed reading from NVRAM",

  /* DBG_STATUS_IDLE_CHK_PARSE_FAILED */
  "Idle check parsing failed",

  /* DBG_STATUS_MCP_TRACE_BAD_DATA */
  "MCP Trace data is corrupt",

  /* DBG_STATUS_MCP_TRACE_NO_META */
  "Dump doesn't contain meta data - it must be provided in
image file",

  /* DBG_STATUS_MCP_COULD_NOT_HALT */
  "Failed to halt MCP",

  /* DBG_STATUS_MCP_COULD_NOT_RESUME */
  "Failed to resume MCP after halt",

  /* DBG_STATUS_RESERVED0 */
  "",

  /* DBG_STATUS_SEMI_FIFO_NOT_EMPTY */
  "Failed to empty SEMI sync FIFO",

  /* DBG_STATUS_IGU_FIFO_BAD_DATA */
  "IGU FIFO data is corrupt",

  /* DBG_STATUS_MCP_COULD_NOT_MASK_PRTY */
  "MCP failed to mask parities",

  /* DBG_STATUS_FW_ASSERTS_PARSE_FAILED */
  "FW Asserts parsing failed",

  /* DBG_STATUS_REG_FIFO_BAD_DATA */
  "GRC FIFO data is corrupt",

  /* DBG_STATUS_PROTECTION_OVERRIDE_BAD_DATA */
  "Protection Override data is corrupt",

  /* DBG_STATUS_DBG_ARRAY_NOT_SET */
  "Debug arrays were not set "
  "(when using binary files, dbg_set_bin_ptr must be
called)",

  /* DBG_STATUS_RESERVED1 */
  "",

  /* DBG_STATUS_NON_MATCHING_LINES */
  "Non-matching debug lines - in E4, all lines must be of "
  "the same type (either 128b or 256b)",

  /* DBG_STATUS_INSUFFICIENT_HW_IDS */
  "Insufficient HW IDs. Try to record less Storms/blocks",

  /* DBG_STATUS_DBG_BUS_IN_USE */
  "The debug bus is in use",

  /* DBG_STATUS_INVALID_STORM_DBG_MODE */
  "The storm debug mode is not supported in the current
chip",

  /* DBG_STATUS_OTHER_ENGINE_BB_ONLY */
  "Other engine is supported only in BB",

  /* DBG_STATUS_FILTER_SINGLE_HW_ID */
  "The configured filter mode requires a single Storm/block
input",

  /* DBG_STATUS_TRIGGER_SINGLE_HW_ID */
  "The configured filter mode requires that all the
constraints of a "
  "single trigger state will be defined on a single
Storm/block input",

  /* DBG_STATUS_MISSING_TRIGGER_STATE_STORM */
  "When triggering on Storm data, the Storm to trigger on
must be specified"
};
```

Массив строк. Скучно? Вот именно поэтому, скорее всего, и существует опечатка.

Где ошибка?


```
const char*
qed_dbg_get_status_str(enum dbg_status status)
{
  return (status < MAX_DBG_STATUS) ?
    s_status_str[status] : "Invalid debug status";
}
```

PVS-Studio: V557 Array overrun is possible. The value of 'status' index could reach 58. qede_debug.c 7149

Вначале я подумал, что это другой классический паттерн ошибки


```
enum Efoo {
  A, B, C, COUNT
};
char Cfoo[] = { 'A', 'B', 'C' };
char Convert(unsigned id)
{
  return (id > COUNT) ? 0 : Cfoo[id];
}
```

Последний элемент перечисления используется как количество элементов в нём.

Но нет, это не наш случай


```
const char*
qed_dbg_get_status_str(enum dbg_status status)
{
  return (status < MAX_DBG_STATUS) ?
    s_status_str[status] : "Invalid debug status";
}
```

MAX_DBG_STATUS – это действительно количество элементов в перечислении, но проверка правильная.

```
DBG_STATUS_PCI_BUF_NOT_ALLOCATED,
DBG_STATUS_INVALID_FILTER_TRIGGER_DWORDS,
DBG_STATUS_NO_MATCHING_FRAMING_MODE,
DBG_STATUS_VFC_READ_ERROR,
....
/* DBG_STATUS_INVALID_FILTER_TRIGGER_DWORDS */
 "The filter/trigger constraint dword offsets are not "
 "enabled for recording",


/* DBG_STATUS_VFC_READ_ERROR */
"Error reading from VFC",
```

Кстати, интересен факт наличия двух пустых строк. Это явно связано с опечаткой.

Пропущена строка для константы DBG_STATUS_NO_MATCHING_FRAMING_MODE

Особенности ошибки

- Возможен выход за границу массива
- Функция возвращает неправильные строки для всех констант, начиная с DBG_STATUS_NO_MATCHING_FRAMING_MODE.

- Несоответствие сложно заметить на обзоре кода

Профилактика

- Использовать PVS-Studio
- Использовать `static_assert`
- Есть ещё такой подход, но он не лучше, так как не гарантирует, что для всех индексов будут прописаны строки:

```
static const char* const s_status_str[] = {
  [DBG_STATUS_OK]                         = "Operation completed successfully",
  [DBG_STATUS_APP_VERSION_NOT_SET]        = "Debug application version wasn't set",
  [DBG_STATUS_UNSUPPORTED_APP_VERSION]    = "Unsupported debug application version",
  ....
};
```

Используйте _Static_assert в C


```
#define countof(array) ( sizeof(array) / sizeof((array)[0]) )

_Static_assert(countof(s_status_str) == MAX_DBG_STATUS,
  "The number of enumerators and string literals are not equal.");
```

Такой countof — плохая практика, но в C красивее не получается.

Используйте static_assert в C++


```
template <typename T, size_t N>
constexpr size_t countof(T(&)[N])
{
  return N;
}
```


```
static_assert(countof(s_status_str) == dbg_status::MAX_DBG_STATUS,
  "The number of enumerators and string literals are not equal.");
```

Защищённый countof. Подробности в статье:

https://pvs-studio.ru/ru/blog/posts/cpp/1112/

Общая рекомендация

Пишите простой и красивый код: его будет легче понять, а ошибки в нём будут заметнее

Интересное для чтения

Монументальная книга: Совершенный код

- Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. - М. : Издательско-торговый дом "Русская редакция"; СПб.: Питер, 2005. - 896 стр.: ил.

Вредные советы для C++ программистов

- Электронный вариант:
  https://pvs-studio.ru/ru/book-giveaway/

Путеводитель C++ программиста по неопределённому поведению

- https://pvs-studio.ru/ru/blog/posts/cpp/1211/

Остались вопросы

Сделай свой проект
чистым и безопасным
вместе с PVS-Studio

PVS-Studio


PVS-Studio