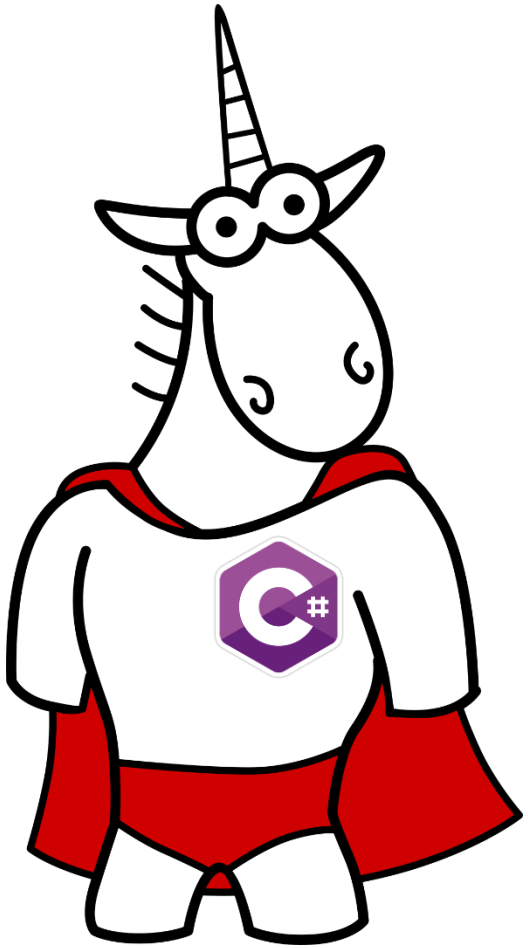
Мифы о безопасности C# кода



Сергей Васильев
PVS-Studio
vasiliev@viva64.com

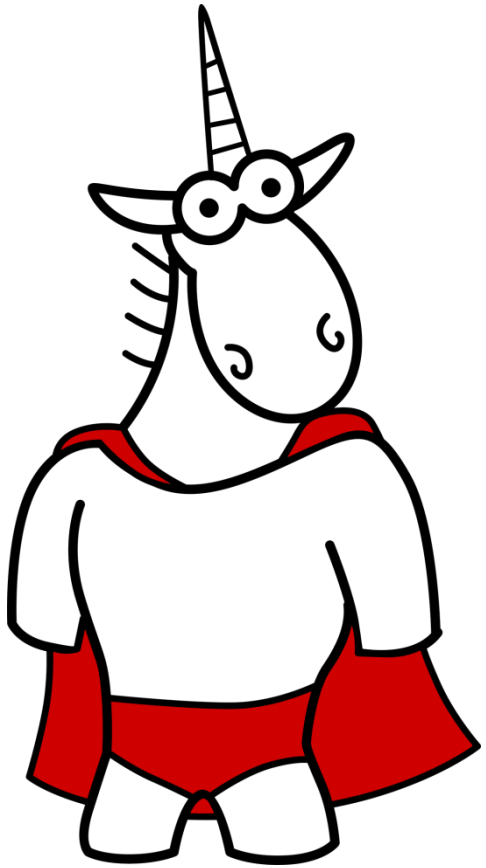
PVS-Studio

О докладчике

- Васильев Сергей
- Ведущий разработчик PVS-Studio.
- Автор статей о проверке open-source проектов.
- Хабрахабр: @foto_shooter
- E-mail: vasiliev@viva64.com

Откуда информация?

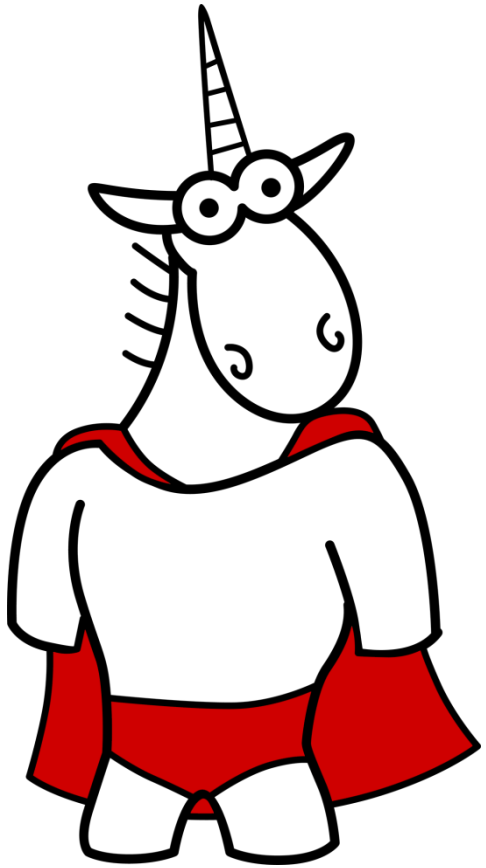
- Какой лучший способ узнать,
  как ошибаются программисты?



Откуда информация?

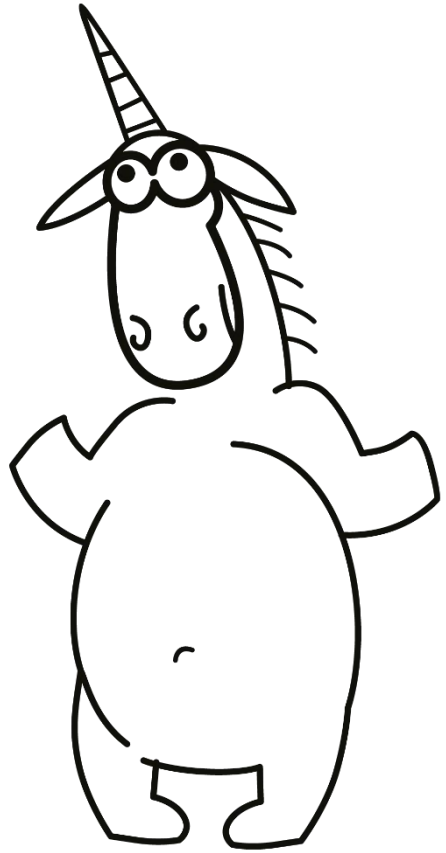
- Какой лучший способ узнать, как ошибаются программисты?

- Посмотреть ошибки на примере реальных проектов!
- Нашли больше 10 000 ошибок.
- Проверили больше 270 проектов.



О чём поговорим?

- C# безопаснее, чем C++, но намного ли?
- Какие ошибки перекочевали в C# из C++, а какие только появились.
- И профессионалы ошибаются: какие ошибки удалось найти в open-source проектах.



Почему C# лучше, чем C++

- Нет ‘сырым’ указателям!
- Исключения вместо неопределённого поведения.
- Некоторые операции, доступные в C++, недоступны в C#.
- FCL – бери и используй!

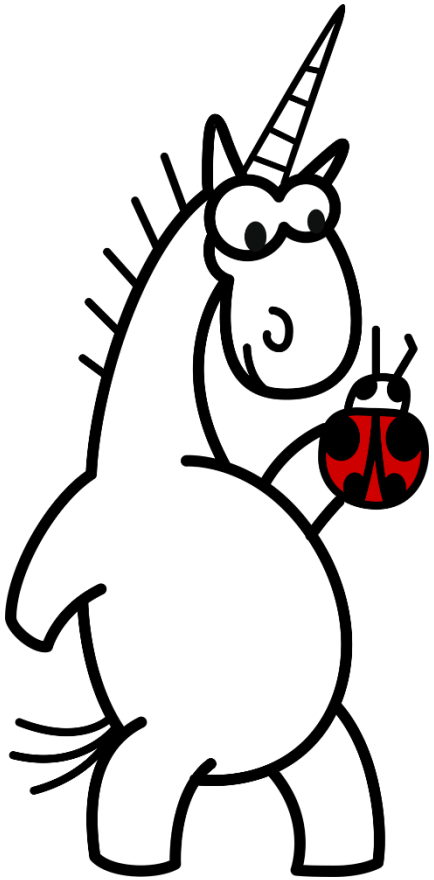
Программируем на C# без ошибок?

Проекты, в которых нашлись ошибки:

- Roslyn;
- CoreFX;
- Mono;
- MSBuild;
- PowerShell;
- Code Contracts;
- Unity 3D Components;
- Xamarin.Forms;
- Space Engineers;
- SharpDevelop;
- NUnit Framework;
- Sony ATF;
- Umbraco;
- Xenko Game Engine;

Разыменование нулевых ссылок

- В C++ разыменовывали нулевые указатели.
- В C# разыменовываем нулевые ссылки.
- Стало лучше, но проблемы остались.



Разыменование нулевых ссылок

PowerShell

```
private void CopyFileFromRemoteSession(....)
{
  ....
  ArrayList remoteFileStreams =
    GetRemoteSourceAlternateStreams(ps, sourceFileFullName);
  if ((remoteFileStreams.Count > 0) &&
      (remoteFileStreams != null))
  ....
}
```

Разыменование нулевых ссылок

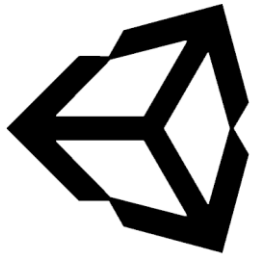
PowerShell

```
private void CopyFileFromRemoteSession(....)
{
  ....
  ArrayList remoteFileStreams =
    GetRemoteSourceAlternateStreams(ps, sourceFileFullName);
  if ((remoteFileStreams.Count > 0) &&
      (remoteFileStreams != null))
  ....
}
```

Разыменование нулевых ссылок

Unity 3D Components

```
public static CrawledMemorySnapshot Unpack(....)
{
  ....
  var result = new CrawledMemorySnapshot
  {
    ....
    staticFields = packedSnapshot.typeDescriptions
      .Where(t => t.staticFieldBytes != null &
                  t.staticFieldBytes.Length > 0)
      ....
  };
  ....
}
```

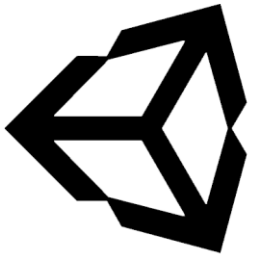


PVS-Studio

Разыменование нулевых ссылок

Unity 3D Components

```
public static CrawledMemorySnapshot Unpack(....)
{
  ....
  var result = new CrawledMemorySnapshot
  {
    ....
    staticFields = packedSnapshot.typeDescriptions
      .Where(t => t.staticFieldBytes != null &
                  t.staticFieldBytes.Length > 0)
      ....
  };
  ....
}
```



PVS-Studio

Разыменование нулевых ссылок

FlashDevelop

```
private void AddFolderItems(MergableMenu menu, string path)
{
  ....
  DirectoryNode node = projectTree.SelectedNode as DirectoryNode;
  if (node.InsideClasspath == node)
    menu.Add(RemoveSourcePath, 2, true);
  else if (node != null && ....) {
    menu.Add(AddSourcePath, 2, false);
  }
  ....
}
```

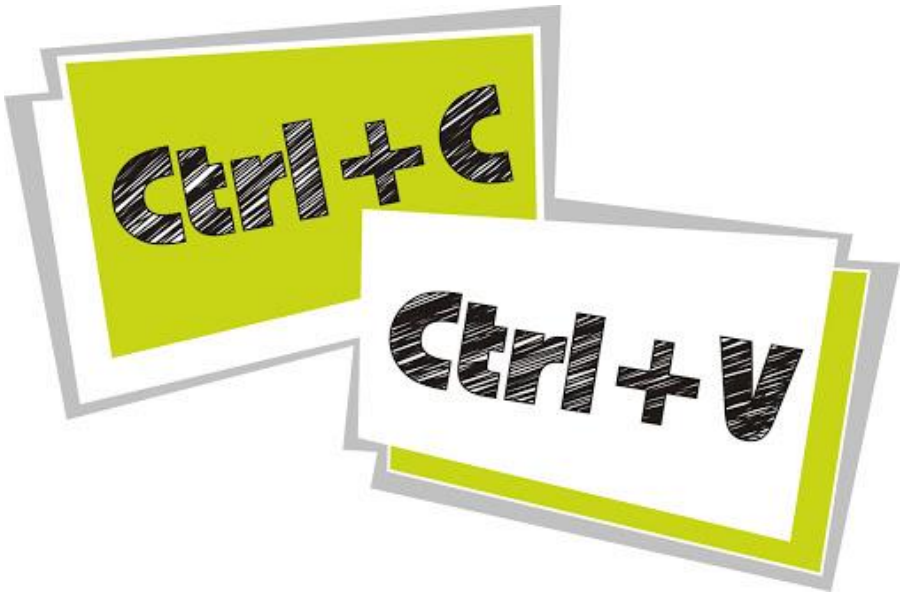
Разыменование нулевых ссылок

FlashDevelop

```
private void AddFolderItems(MergableMenu menu, string path)
{
  ....
  DirectoryNode node = projectTree.SelectedNode as DirectoryNode;
  if (node.InsideClasspath == node)
    menu.Add(RemoveSourcePath, 2, true);
  else if (node != null && ....) {
    menu.Add(AddSourcePath, 2, false);
  }
  ....
}
```

Copy-paste

- Был и в C++, остался и в C#.
- Copy-paste – зло, но зло необходимое.
- Не просто дублирующийся код, а припрятанные в листве грабли.



Опечатки

- Были и в C++, остались и в C#.
- Компилятор предупреждает далеко не всегда.
- Проблемы со схожими именами переменных и не только.

Опечатки и copy-paste

Xamarin.Forms

```
public double Left   { get; set; }
public double Top    { get; set; }
public double Right  { get; set; }
public double Bottom { get; set; }

internal bool IsDefault
{
  get { return Left  == 0 && Top  == 0 &&
               Right == 0 && Left == 0; }
}
```

Опечатки и copy-paste

Xamarin.Forms

```
public double Left   { get; set; }
public double Top    { get; set; }
public double Right  { get; set; }
public double Bottom { get; set; }

internal bool IsDefault
{
  get { return Left  == 0 && Top  == 0 &&
               Right == 0 && Left == 0; }
}
```

PVS-Studio

Опечатки и copy-paste

Roslyn

```
public void IndexerMemberRace() {
  ....
  for (int i = 0; i < 20; i++) {
    ....
    if (i % 2 == 0) {
      thread1.Start();
      thread2.Start();
    } else {
      thread1.Start();
      thread2.Start();
    }
    ....
  }
  ....
}
```

Опечатки и copy-paste

Roslyn

```
public void IndexerMemberRace() {
  ....
  for (int i = 0; i < 20; i++) {
    ....
    if (i % 2 == 0) {
      thread1.Start();
      thread2.Start();
    } else {
      thread1.Start();
      thread2.Start();
    }
    ....
  }
  ....
}
```

Опечатки и copy-paste

Sony ATF

```
public ProgressCompleteEventArgs(Exception progressError,
                                 object progressResult,
                                 bool cancelled)
{
  ProgressError = ProgressError;
  ProgressResult = progressResult;
  Cancelled = cancelled;
}
```

Опечатки и copy-paste

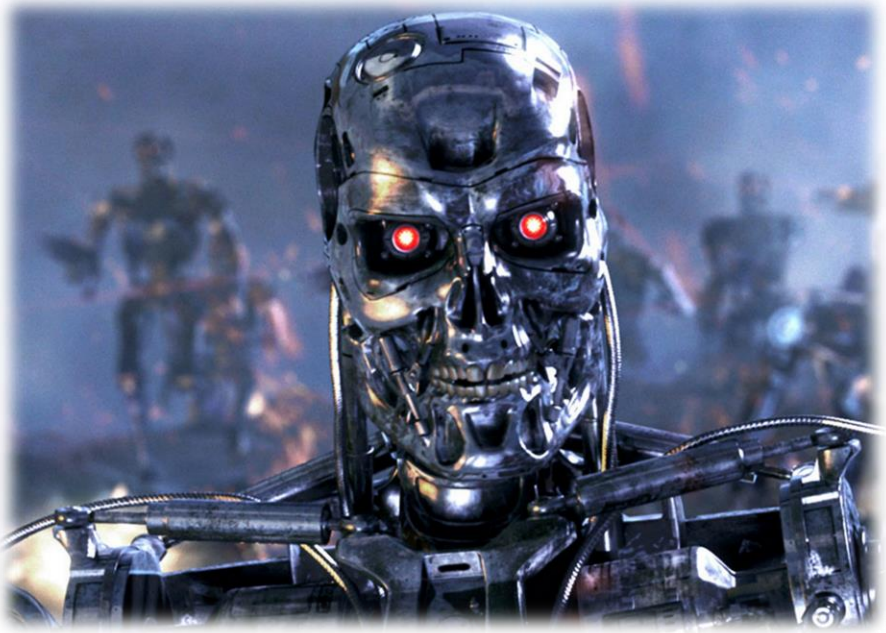
Sony ATF

```
public ProgressCompleteEventArgs(Exception progressError,
                                 object progressResult,
                                 bool cancelled)
{
  ProgressError = ProgressError;
  ProgressResult = progressResult;
  Cancelled = cancelled;
}
```

Следите за строками форматирования!

Неправильное использование printf в C может привести к:

- непредсказуемому поведению;
- выводу мусора;
- повреждению стека;
- выполнению произвольного кода;
- ….



Следите за строками форматирования!

- Неправильное использование
  String.Format может привести к падению.
- Стало лучше, но простор
  для ошибок остался большим.
- Да здравствуют
  интерполированные строки?

Строки форматирования

SharpDevelop

```
public override string ToString()
{
  return String.Format("Thread Name = {1} Suspended = {2}",
                       ID, Name, Suspended);
}
```

Строки форматирования

SharpDevelop

```
public override string ToString()
{
  return String.Format("Thread Name = {1} Suspended = {2}",
                       ID, Name, Suspended);
}
```

Строки форматирования

Mono

```
public override string ToString ()
{
  return string.Format ("ListViewSubItem {{0}}", text);
}
```

Строки форматирования

Mono

```
public override string ToString ()
{
  return string.Format ("ListViewSubItem {{0}}", text);
}
```

Как ошибиться, используя оператор ‘as’?

Неверное сравнение ссылки после приведения
с использованием оператора ‘as’.

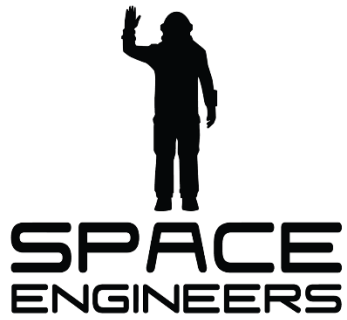
Паттерн:

```
var derObj = baseObj as Derived;
if (baseObj == null)
  ....
```

Как ошибиться, используя оператор ‘as’?

Space Engineers

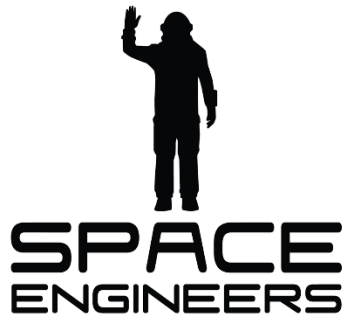
```
private void contextMenu_ItemClicked(....)
{
  ....
  var actionsItem = item as MyToolbarItemActions;
  if (item != null) {
    if (idx < 0 || idx >= actionsItem.PossibleActions(ShownToolbar.ToolbarType)
                                                .Count)
      RemoveToolbarItem(slot);
    ....
  }
  ....
}
```



Как ошибиться, используя оператор ‘as’?

Space Engineers

```
private void contextMenu_ItemClicked(....)
{
  ....
  var actionsItem = item as MyToolbarItemActions;
  if (item != null) {
    if (idx < 0 || idx >= actionsItem.PossibleActions(ShownToolbar.ToolbarType)
                                                .Count)
      RemoveToolbarItem(slot);
    ....
  }
  ....
}
```



Специфические ошибки

Тысячи их! (с)

Например:

- Ошибки использования атрибута [ThreadStatic].
- Ошибки сериализации объектов.
- Ошибки синхронизации потоков.
- Забыли про иммутабельность строк.
- Забыли, как работает LINQ.
- Неправильное использование null-conditional оператора.
- И т.д. и т.п.

Специфические ошибки

SharpDevelop

```
void Init()
{
  ....
  this.SequencePoints.OrderBy(item => item.Line)
                     .OrderBy(item => item.Column);
}
```

Специфические ошибки

SharpDevelop

```
void Init()
{
  ....
  this.SequencePoints.OrderBy(item => item.Line)
                     .OrderBy(item => item.Column);
}
```

Специфические ошибки

Mono

```
static class Profiler
{
  [ThreadStatic]
  private static Stopwatch timer = new Stopwatch();
  ....
}
```

PVS-Studio

Специфические ошибки

Mono

```
static class Profiler
{
  [ThreadStatic]
  private static Stopwatch timer = new Stopwatch();
  ....
}
```

Документация по правилу V3089:

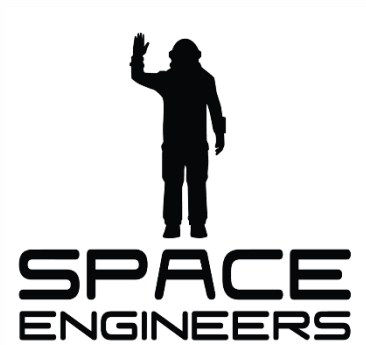
http://www.viva64.com/ru/w/V3089/

PVS-Studio

Специфические ошибки

Space Engineers

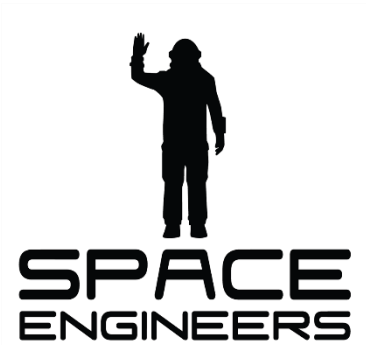
```
public class MyHierarchyComponentBase : MyEntityComponentBase
{
  ....
  [ThreadStatic]
  HashSet<object> m_queryResults = new HashSet<object>();
  ....
}
```



Специфические ошибки

Space Engineers

```
public class MyHierarchyComponentBase : MyEntityComponentBase
{
  ....
  [ThreadStatic]
  HashSet<object> m_queryResults = new HashSet<object>();
  ....
}
```



Специфические ошибки

PowerShell

```
private CatchClauseAst CatchBlockRule(.... ref List<TypeConstraintAst> errorAsts)
{
  ....
  if (errorAsts == null) {
    errorAsts = exceptionTypes;
  } else {
    errorAsts.Concat(exceptionTypes);
  }
  ....
}
```

Специфические ошибки

PowerShell

```
private CatchClauseAst CatchBlockRule(.... ref List<TypeConstraintAst> errorAsts)
{
  ....
  if (errorAsts == null) {
    errorAsts = exceptionTypes;
  } else {
    errorAsts.Concat(exceptionTypes);
  }
  ....
}
```

PVS-Studio

Специфические ошибки

Microsoft Bot Builder SDK

```
public async Task<ClaimsIdentity>
  GetIdentityAsync(string authorizationHeader)
{
  ....
  string[] parts = authorizationHeader?.Split(' ');
  if (parts.Length == 2)
    return await GetIdentityAsync(parts[0], parts[1])
                  .ConfigureAwait(false);
  ....
}
```

Специфические ошибки

Microsoft Bot Builder SDK

```
public async Task<ClaimsIdentity>
  GetIdentityAsync(string authorizationHeader)
{
  ....
  string[] parts = authorizationHeader?.Split(' ');
  if (parts.Length == 2)
    return await GetIdentityAsync(parts[0], parts[1])
                  .ConfigureAwait(false);
  ....
}
```

Специфические ошибки

SharpDevelop

```
public string Text { get; set; }
....
protected override void OnKeyUp(KeyEventArgs e)
{
  ....
  editor.Text.Insert(editor.CaretIndex, Environment.NewLine);
  ....
}
```

Специфические ошибки

SharpDevelop

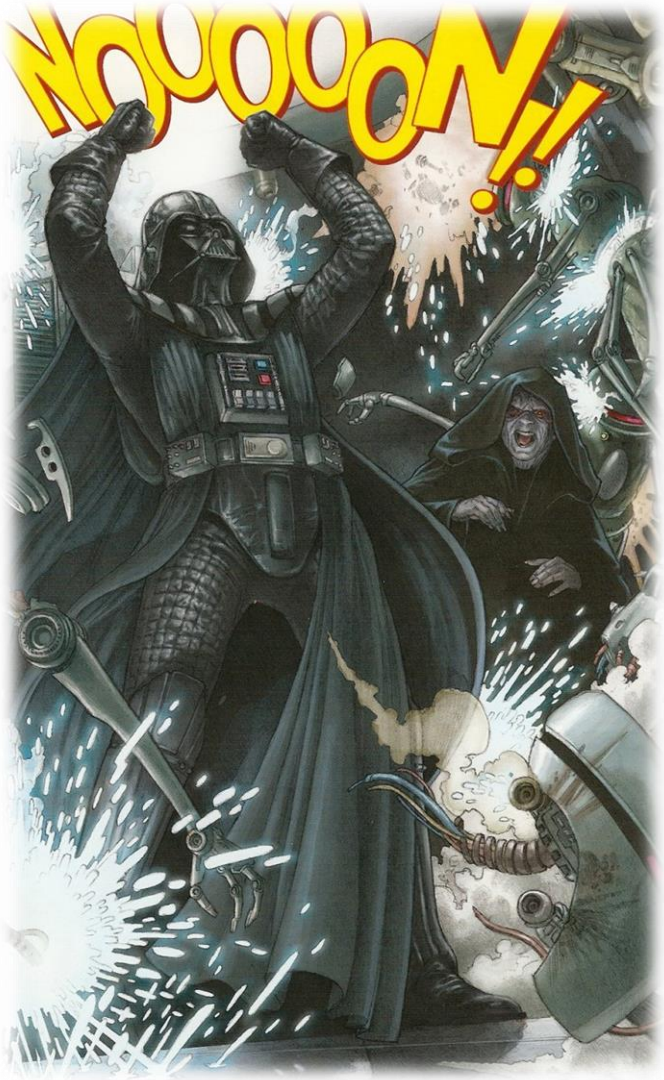
```
public string Text { get; set; }
....
protected override void OnKeyUp(KeyEventArgs e)
{
  ....
  editor.Text.Insert(editor.CaretIndex, Environment.NewLine);
  ....
}
```

Как избежать ошибок?

PVS-Studio

Как избежать ошибок?

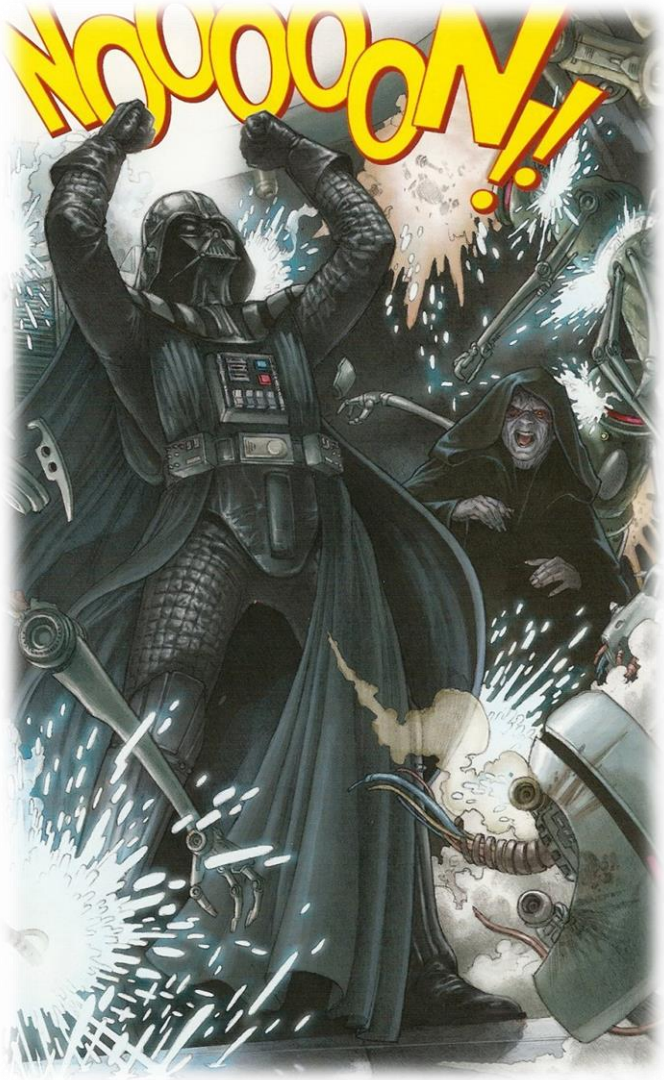
Никак!



PVS-Studio

Как избежать ошибок?

Никак!



Не расстраивайтесь.
Ошибки будут всегда,
но можно попытаться
минимизировать
их количество.

Как избежать ошибок?

- По возможности, используйте интерполированные строки.
- Не переусложняйте.
- Следите за форматированием кода.
- Используйте специализированные средства поиска ошибок.
- Не скупитесь на фигурные скобки.

Использование интерполированных строк

- Ошибки:
  - Несоответствие количества аргументов;
  - Неверная нумерация элементов формата.
- Предотвращаем ошибки неверного количества аргументов.
- Пример:

```
return $"ID = {ID}, Name = {Name}";
```

Не переусложняйте

Xenko Game Engine

```
public static ContainmentType BoxContainsSphere(ref BoundingBox box,
                                                ref BoundingSphere sphere)
{
  ....
  if ((((box.Minimum.X + sphere.Radius <= sphere.Center.X)  &&
        (sphere.Center.X <= box.Maximum.X - sphere.Radius)) &&
       ((box.Maximum.X - box.Minimum.X > sphere.Radius)     &&
       (box.Minimum.Y + sphere.Radius <= sphere.Center.Y))) &&
      (((sphere.Center.Y <= box.Maximum.Y - sphere.Radius)  &&
        (box.Maximum.Y - box.Minimum.Y > sphere.Radius))    &&
      (((box.Minimum.Z + sphere.Radius <= sphere.Center.Z)  &&
      (sphere.Center.Z <= box.Maximum.Z - sphere.Radius))   &&
        (box.Maximum.X - box.Minimum.X > sphere.Radius))))
  ....
}
```

Не переусложняйте

Xenko Game Engine

```
public static ContainmentType BoxContainsSphere(ref BoundingBox box,
                                                ref BoundingSphere sphere)
{
  ....
  if ((((box.Minimum.X + sphere.Radius <= sphere.Center.X)  &&
        (sphere.Center.X <= box.Maximum.X - sphere.Radius)) &&
       ((box.Maximum.X - box.Minimum.X > sphere.Radius)     &&
       (box.Minimum.Y + sphere.Radius <= sphere.Center.Y))) &&
      (((sphere.Center.Y <= box.Maximum.Y - sphere.Radius)  &&
        (box.Maximum.Y - box.Minimum.Y > sphere.Radius))    &&
      (((box.Minimum.Z + sphere.Radius <= sphere.Center.Z)  &&
      (sphere.Center.Z <= box.Maximum.Z - sphere.Radius))   &&
        (box.Maximum.X - box.Minimum.X > sphere.Radius))))
  ....
}
```

Не переусложняйте

Xenko Game Engine

```
public static ContainmentType BoxContainsSphere(ref BoundingBox box,
                                                 ref BoundingSphere sphere)
{
  ....
  var xCompRes = ....;
  var yCompRes = ....;
  var zCompRes = ....;
  if (xCompRes && yCompRes && zCompRes)
  ....
}
```

Следите за форматированием

Mono

```
static bool AreEqual (VisualStyleElement value1,
                      VisualStyleElement value2)
{
  return
    value1.ClassName == value1.ClassName &&
    value1.Part == value2.Part &&
    value1.State == value2.State;
}
```

Следите за форматированием

Mono

```
static bool AreEqual (VisualStyleElement value1,
                      VisualStyleElement value2)
{
  return
    value1.ClassName == value1.ClassName &&
    value1.Part == value2.Part &&
    value1.State == value2.State;
}
```

Следите за форматированием

Mono

```
static bool AreEqual (VisualStyleElement value1,
                      VisualStyleElement value2)
{
  return
       value1.ClassName == value1.ClassName
    && value1.Part      == value2.Part
    && value1.State     == value2.State;
}
```

Аккуратнее при copy-paste

Accord.Net

```
public Blob[] GetObjects(UnmanagedImage image,
                         bool extractInOriginalSize)
{
  ....
  if ((image.PixelFormat != PixelFormat.Format24bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format8bppIndexed) &&
      (image.PixelFormat != PixelFormat.Format32bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format32bppArgb)   &&
      (image.PixelFormat != PixelFormat.Format32bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format32bppPArgb)
      )
  ....
}
```

Аккуратнее при copy-paste

Accord.Net

```
public Blob[] GetObjects(UnmanagedImage image,
                         bool extractInOriginalSize)
{
  ....
  if ((image.PixelFormat != PixelFormat.Format24bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format8bppIndexed) &&
      (image.PixelFormat != PixelFormat.Format32bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format32bppArgb)   &&
      (image.PixelFormat != PixelFormat.Format32bppRgb)    &&
      (image.PixelFormat != PixelFormat.Format32bppPArgb)
      )
  ....
}
```

Не скупитесь на фигурные скобки

Sony ATF

```
public static QuatF Slerp(QuatF q1, QuatF q2, float t)
{
  double dot = q2.X * q1.X + q2.Y * q1.Y + q2.Z * q1.Z + q2.W * q1.W;

  if (dot < 0)
    q1.X = -q1.X; q1.Y = -q1.Y; q1.Z = -q1.Z; q1.W = -q1.W;
  ....
}
```

Не скупитесь на фигурные скобки

Sony ATF

```
public static QuatF Slerp(QuatF q1, QuatF q2, float t)
{
  double dot = q2.X * q1.X + q2.Y * q1.Y + q2.Z * q1.Z + q2.W * q1.W;

  if (dot < 0)
    q1.X = -q1.X; q1.Y = -q1.Y; q1.Z = -q1.Z; q1.W = -q1.W;
  ....
}
```

Не скупитесь на фигурные скобки

Sony ATF

```
public static QuatF Slerp(QuatF q1, QuatF q2, float t)
{
  double dot = q2.X * q1.X + q2.Y * q1.Y + q2.Z * q1.Z + q2.W * q1.W;

  if (dot < 0)
    { q1.X = -q1.X; q1.Y = -q1.Y; q1.Z = -q1.Z; q1.W = -q1.W; }
  ....
}
```

Внимательнее со схожими именами

MonoDevelop

```
public string WhiteSpaceText { get; set; }

public WhitespaceNode(string whiteSpaceText,
                      TextLocation startLocation)
{
  this.WhiteSpaceText = WhiteSpaceText;
  this.startLocation  = startLocation;
}
```

Внимательнее со схожими именами

MonoDevelop

```
public string WhiteSpaceText { get; set; }

public WhitespaceNode(string whiteSpaceText,
                      TextLocation startLocation)
{
  this.WhiteSpaceText = WhiteSpaceText;
  this.startLocation  = startLocation;
}
```

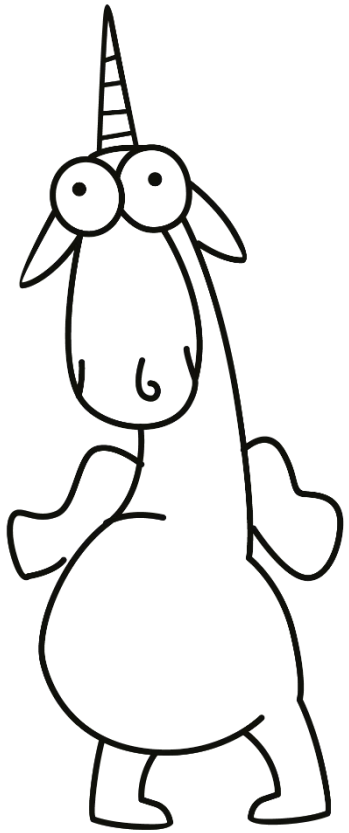
Статический анализ

- Исследуем исходный код программ без их выполнения.
- Ищем ошибки на ранних этапах написания кода.
- От разовых проверок толка мало – необходимо регулярное использование анализатора.
- Не стоит думать, что решает все проблемы разом.

Анализатор – друг или враг?

Видение программиста:

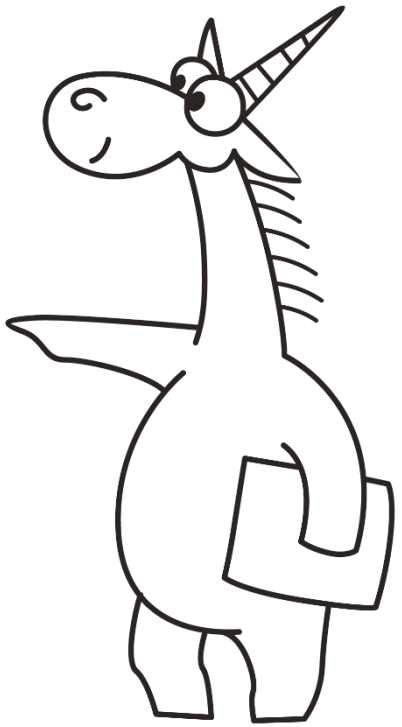
- Локальное использование:
  анализатор – друг.
- Использование на сборочном сервере:
  анализатор – враг.



PVS-Studio

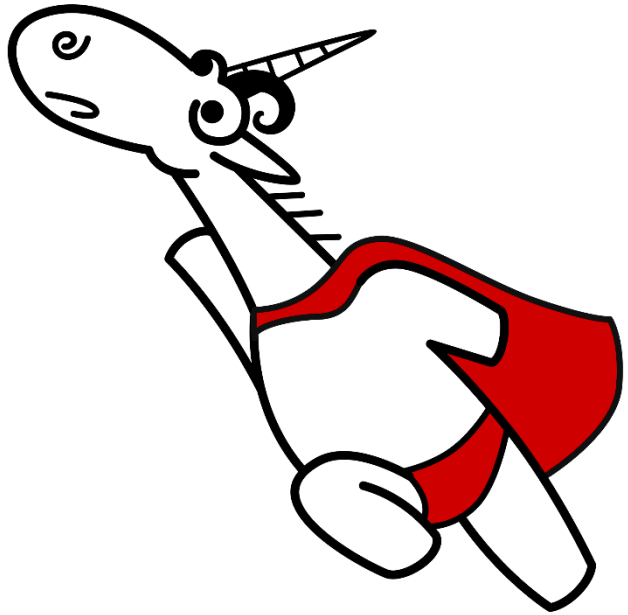
Анализатор – друг или враг?

Наиболее эффективный вариант – совмещение локального использования и проверок на сборочном сервере.



PVS-Studio

- Сайт: http://www.viva64.com/
- Статьи о проверке проектов;
- Технические статьи;
- База ошибок, найденных в open-source проектах;
- Можно скачать и попробовать анализатор на интересующем проекте.



PVS-Studio

Вопросы?

- E-mail: vasiliev@viva64.com
- Хабрахабр: @foto_shooter
- PVS-Studio: viva64.com

PVS-Studio