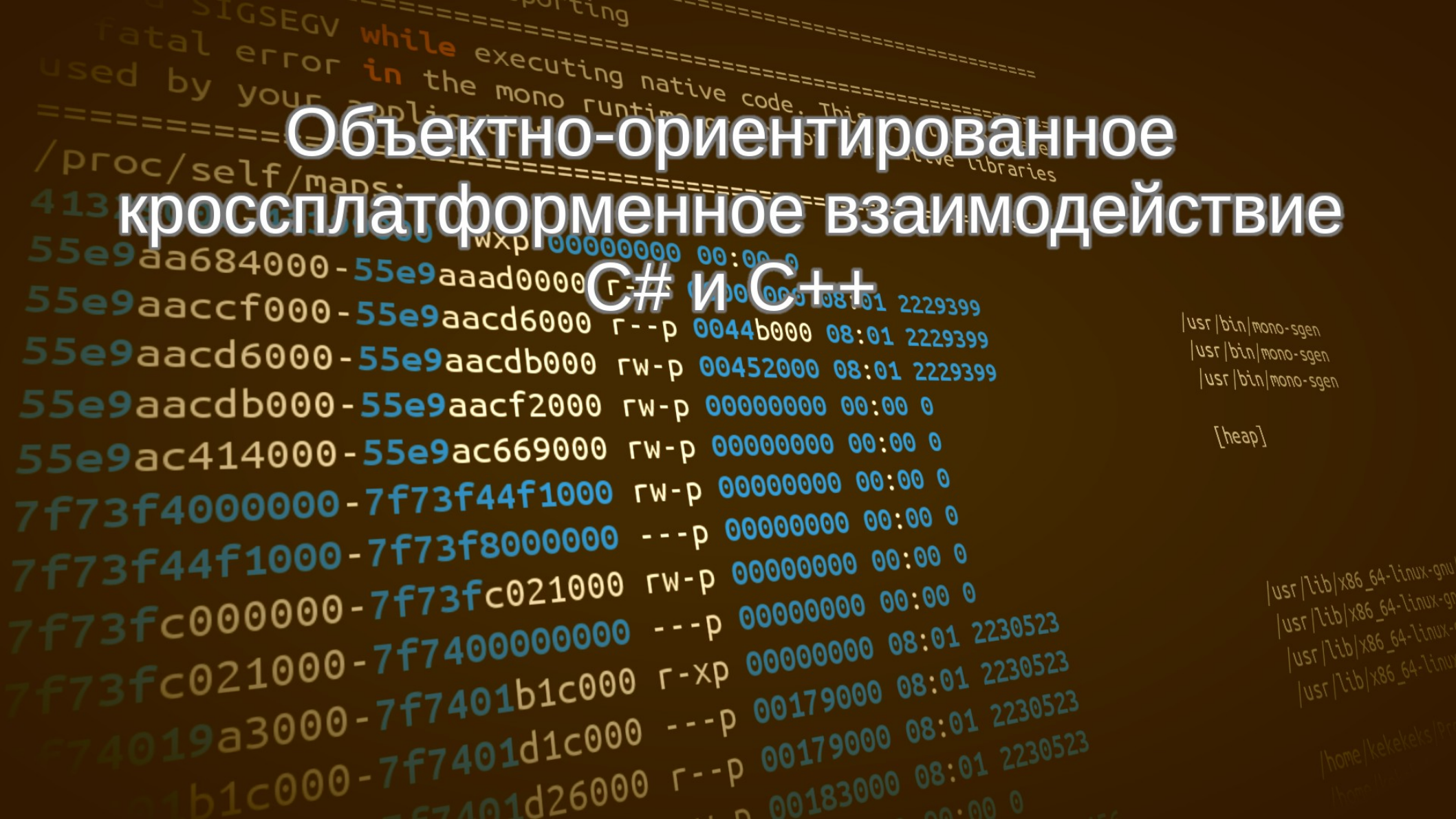
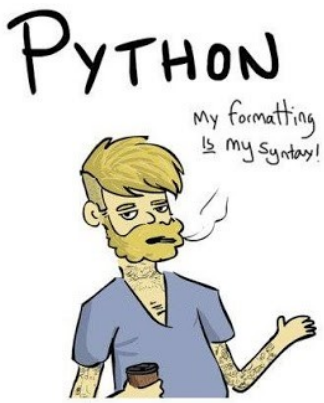
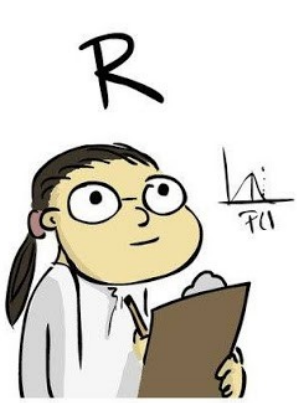
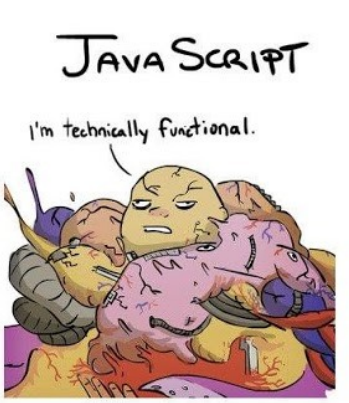

Объектно-ориентированное
кроссплатформенное взаимодействие
C# и C++


PYTHON
My formatting is my syntax!


R


JAVA


JAVASCRIPT
I'm technically functional.


PHP
Can I GET you guys anything?


HASKELL


PERL


C
C++


C#
I am nothing like my father.


RUBY ON RAILS

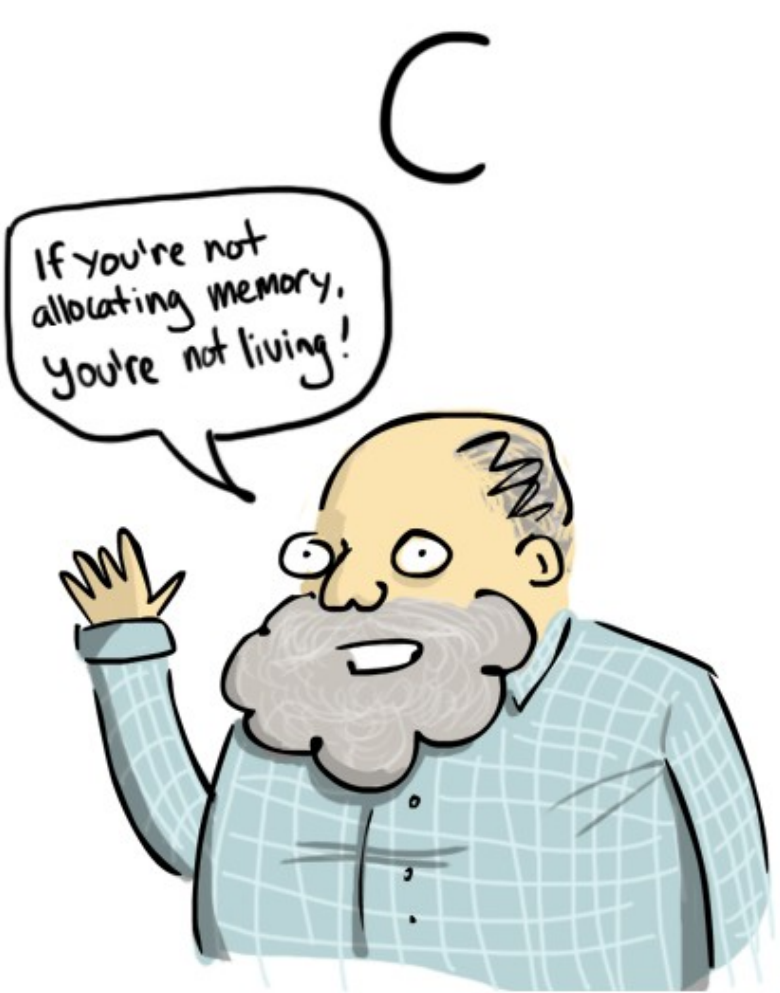

ASSEMBLY


ERLANG
DESTROY ONE OF MY PROCESSES & I WILL ONLY GROW STRONGER


GO
Google


BRAINFUCK
[
.
D.martino
leftoversalad.tumblr.com


Rust
I'll be designated driver tonight!
We know
Don't talk while eating you could choke and die haha


C
If you're not allocating memory, you're not living!


C++
If you're not allocating memory, you're not living!

Легкость интеропа с C

Легкость интеропа с C

- Накидал [DllImport] и в дамки

Легкость интеропа с C

- Накидал [DllImport] и в дамки
- Все типы данных C представимы в C#, некоторые ещё и конвертируются автоматически (string -> wchar*/char*)

Легкость интеропа с C

- Накидал [DllImport] и в дамки
- Все типы данных C представимы в C#, некоторые ещё и конвертируются автоматически (string -> wchar*/char*)
- Структуры данных — это просто структуры (понятные предсказуемые смещения всех полей)

Легкость интеропа с C

- Накидал [DllImport] и в дамки
- Все типы данных C представимы в C#, некоторые ещё и конвертируются автоматически (string -> wchar*/char*)
- Структуры данных — это просто структуры (понятные предсказуемые смещения всех полей)
- Можно спокойно вызывать любую C-функцию, а C передать любой метод, лямбду итп.

Проблемы интеропа с C

Проблемы интероп с C

- Управление памятью (два вопроса: «кто?» и «как?»)

Проблемы интероп с C

- Управление памятью (два вопроса: «кто?» и «как?»)

```
char* get_file_name(char* path);
```

Проблемы интероп с C

- Управление памятью (два вопроса: «кто?» и «как?»)

```
char* get_file_name(char* path);
```

- Управление временем жизни делегатов

```
public delegate void MyCallback(int arg);

[DllImport(Consts.LibraryPath)]
static extern void SetCCallback(MyCallback cb);

SetCCallback(i => Console.WriteLine("Called: " + i));
```


```
private static MyCallback _callback;

_callback = i => Console.WriteLine("Called:" + i);
SetCCallback(_callback);
```


```
void SetCallback(MyCallback cb, void* userData)
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```


```
public delegate void MyTargetedCallback(int arg);

private static MyCallback _staticCallback = MyCallbackHandler;
static void MyCallbackHandler(int arg, IntPtr userData)
{
    var cb = (MyTargetedCallback) GCHandle.FromIntPtr(userData).Target;
    cb(arg);
}

public static GCHandle SetCallback(MyTargetedCallback cb)
{
    var handle = GCHandle.Alloc(cb);
    SetCCallback(_staticCallback, GCHandle.ToIntPtr(handle));
    return handle;
}
```

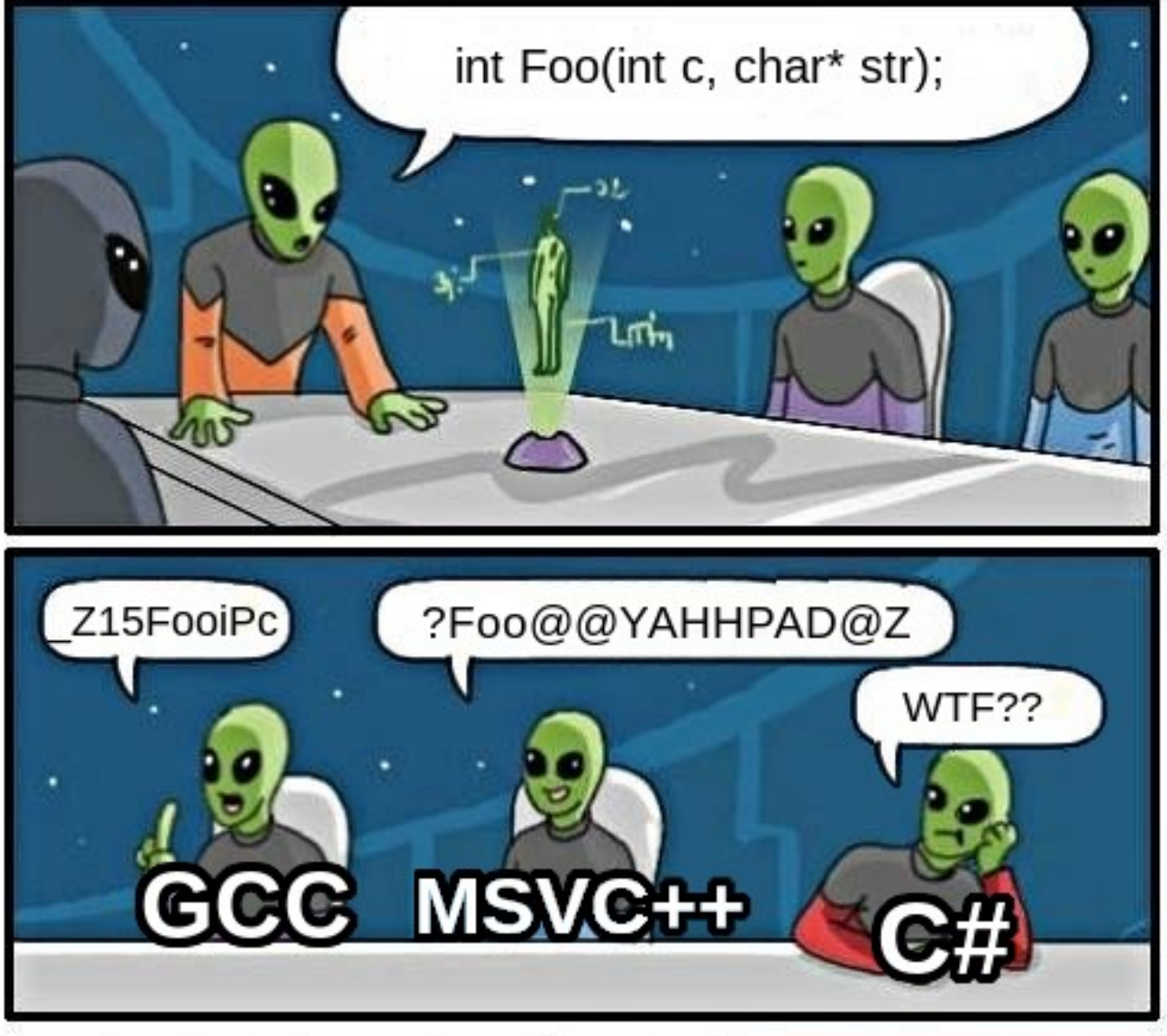

Поговорим про C++

Поговорим про C++

- Накидал [DllImport] и

Поговорим про C++

- Накидал [DllImport] и
- ?Test@@YAXV?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@V?$set@V?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@U?$less@V?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@@2@V?$allocator@V?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@std@@@2@@2@@2@@Z?Test2@MyClass@@QAEHAAHPBD@Z


int Foo(int c, char* str);
_Z15FooiPc
?Foo@@YAHHPAD@Z
WTF??
GCC
MSVC++
C#

Сложности интероперa с C++

Сложности интероперa с C++

- Мэнглинг имён (http://demangler.com/)

Сложности интеропа с C++

- Мэнглинг имён (http://demangler.com/)
- Структуры не структуры

Сложности интеропа с C++

- Мэнглинг имён (http://demangler.com/)
- Структуры не структуры
- Разные модели наследования

Сложности интероп с C++

- Мэнглинг имён (http://demangler.com/)
- Структуры не структуры
- Разные модели наследования
- Разные модели управления памятью (GC в C# vs. («полу»-)ручное в C++)

Исторически это решалось

Исторически это решалось

- На Windows — Managed C++, затем C++/CLI

Исторически это решалось

- На Windows — Managed C++, затем C++/CLI
- На *nix — glue-код + P/Invoke

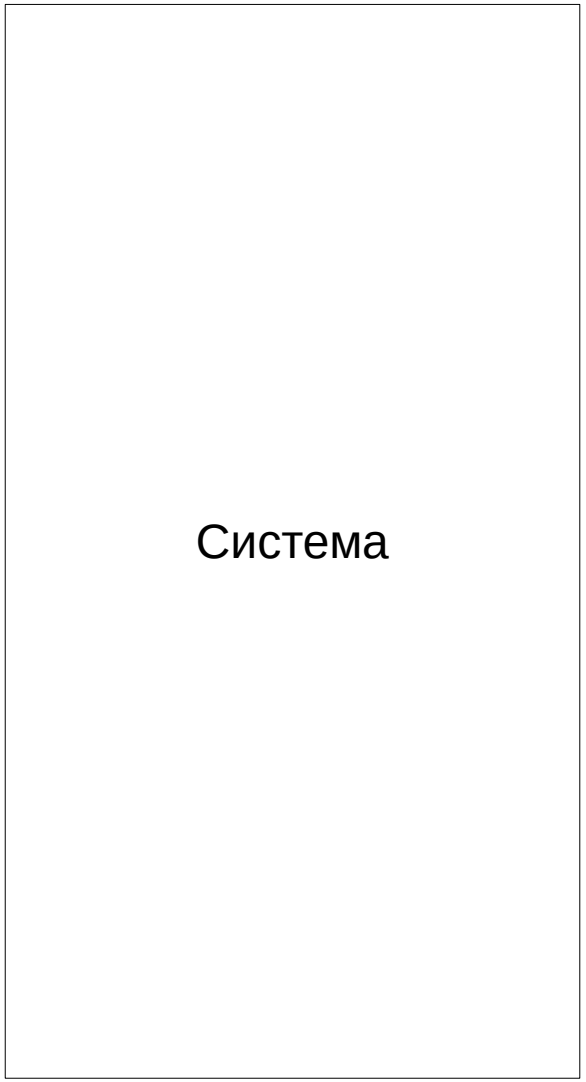
Исторически это решалось

- На Windows — Managed C++, затем C++/CLI
- На *nix — glue-код + P/Invoke

```
extern "C" int
call_MyClass_void_Test2_intptr_charptr
    (MyClass* cl, int* arg, const char* arg2)
{
    cl->Test2(arg, arg2);
}
```

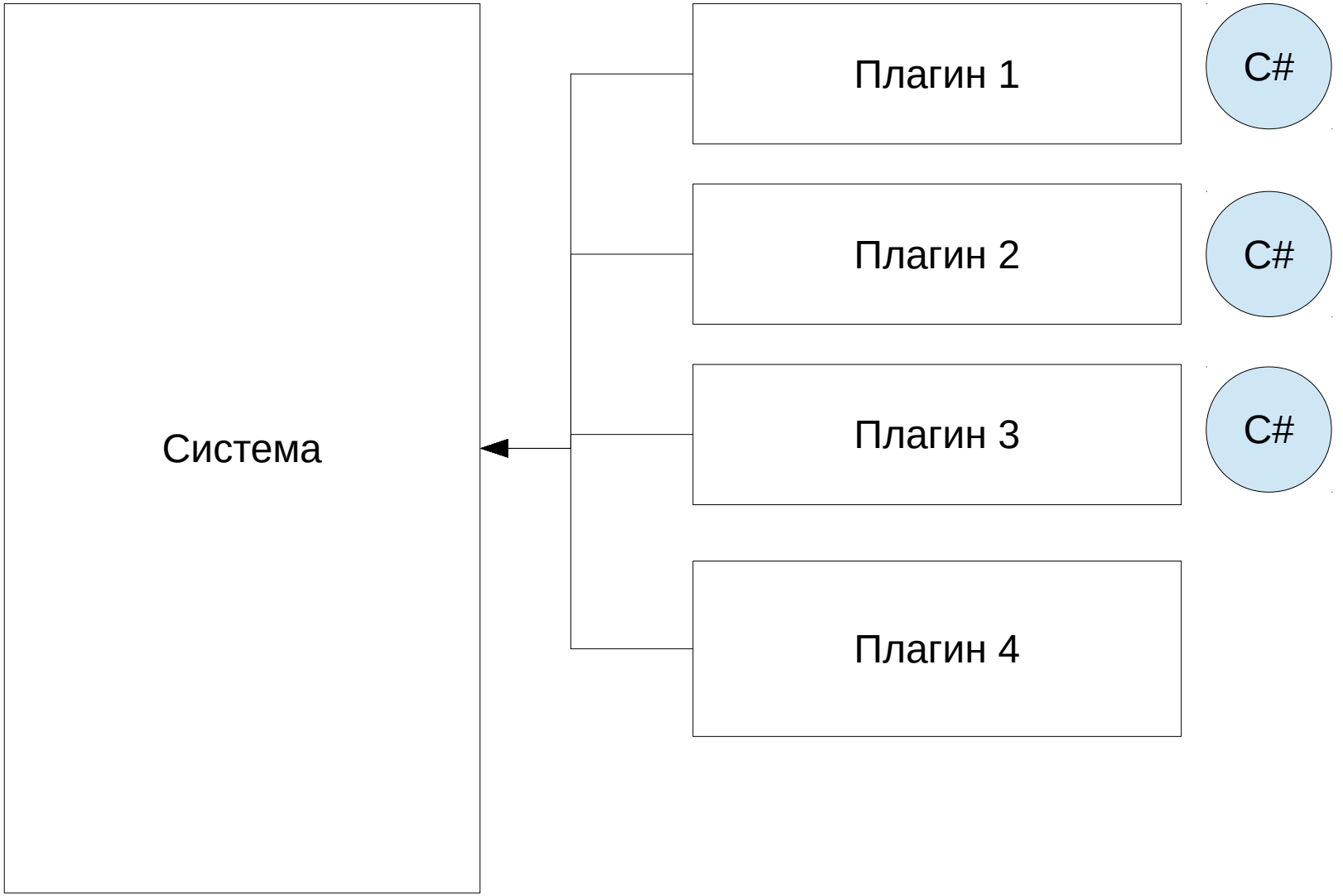

Проблемы, толком не решаемые glue-кодом

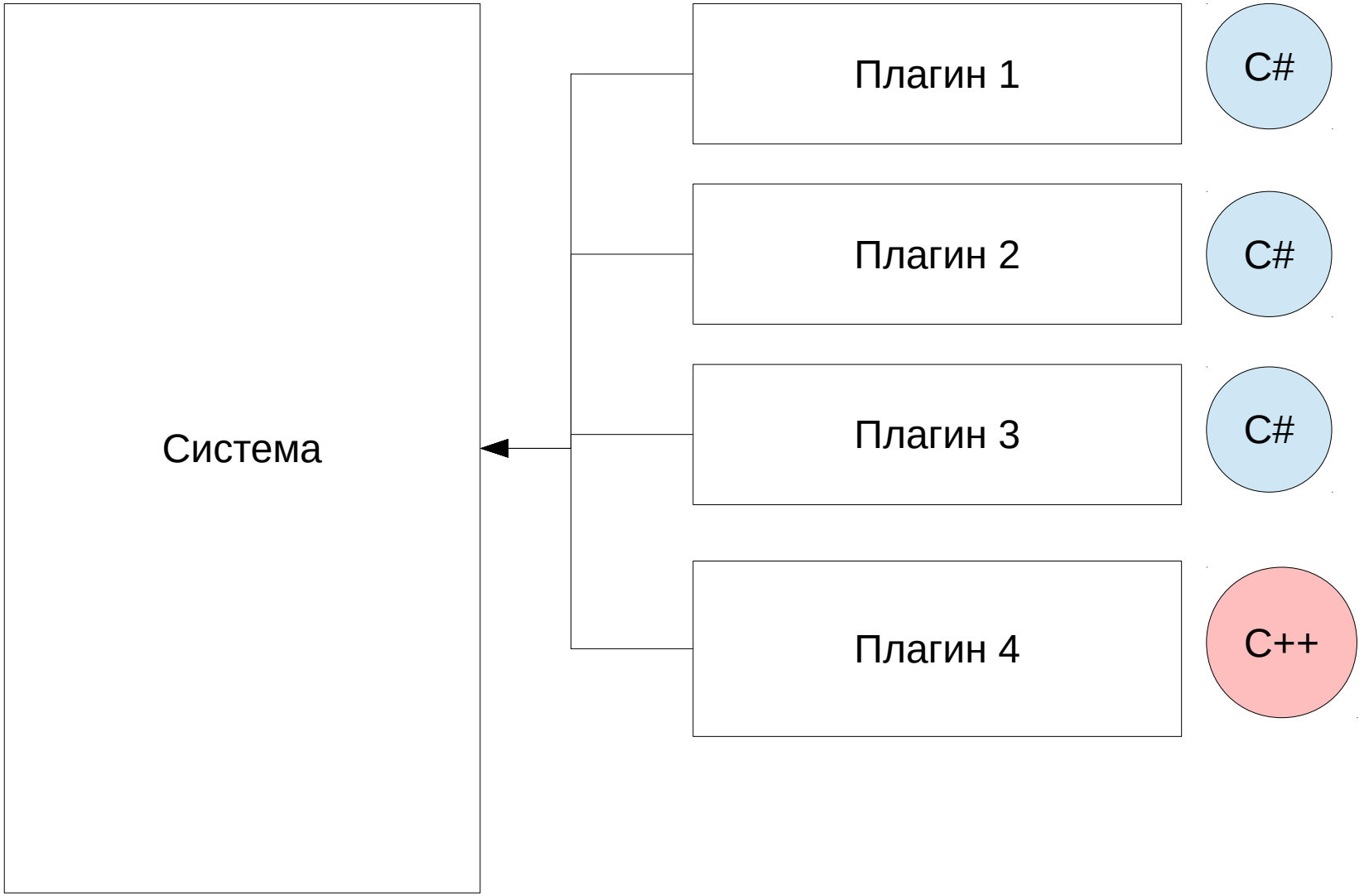
- в случае с экспортом C++-объекта в C# не всегда ясно, кто "владеет" объектом
- в случае с экспортом C#-объекта в C++ всё ещё веселее

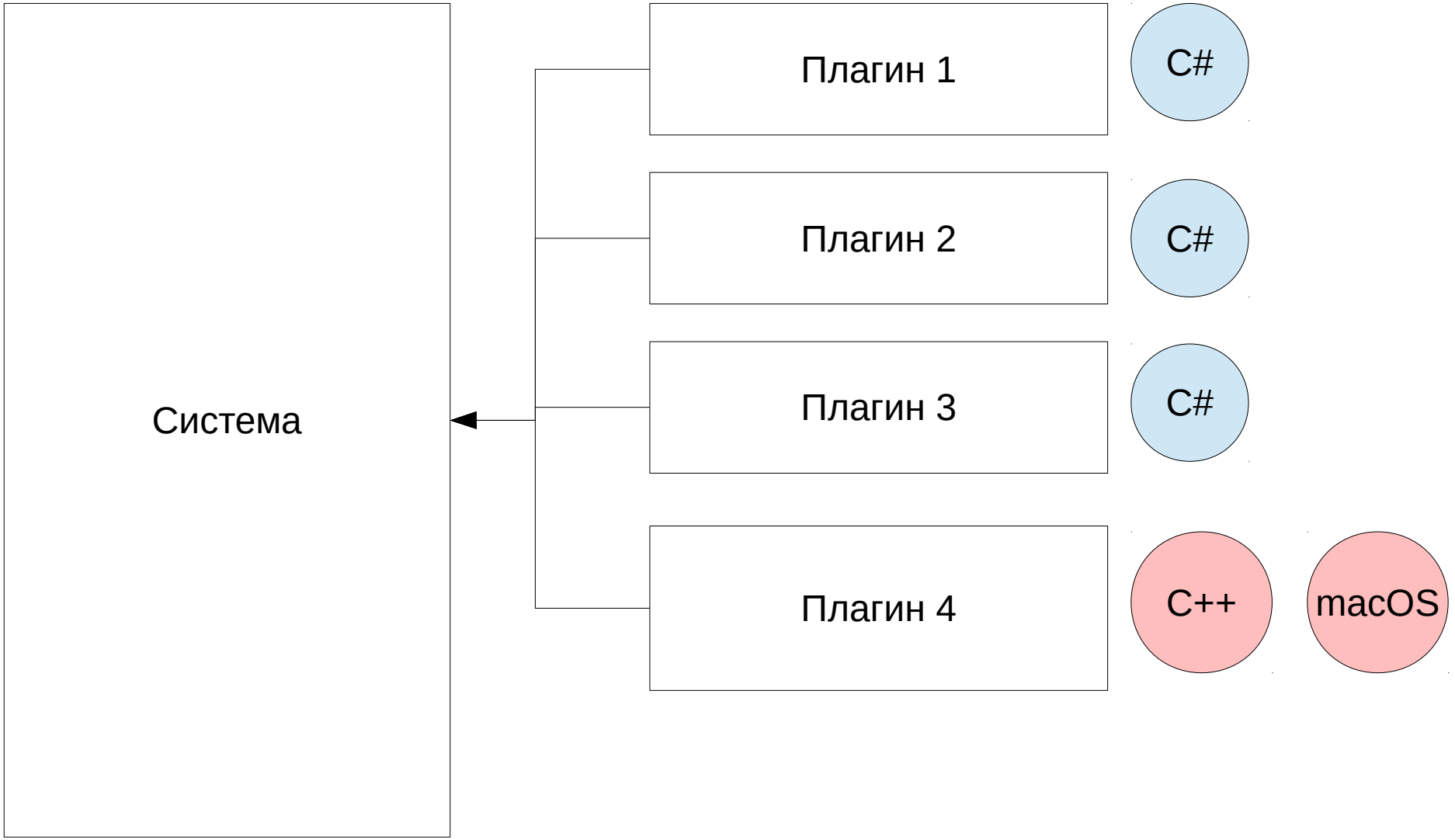

Система

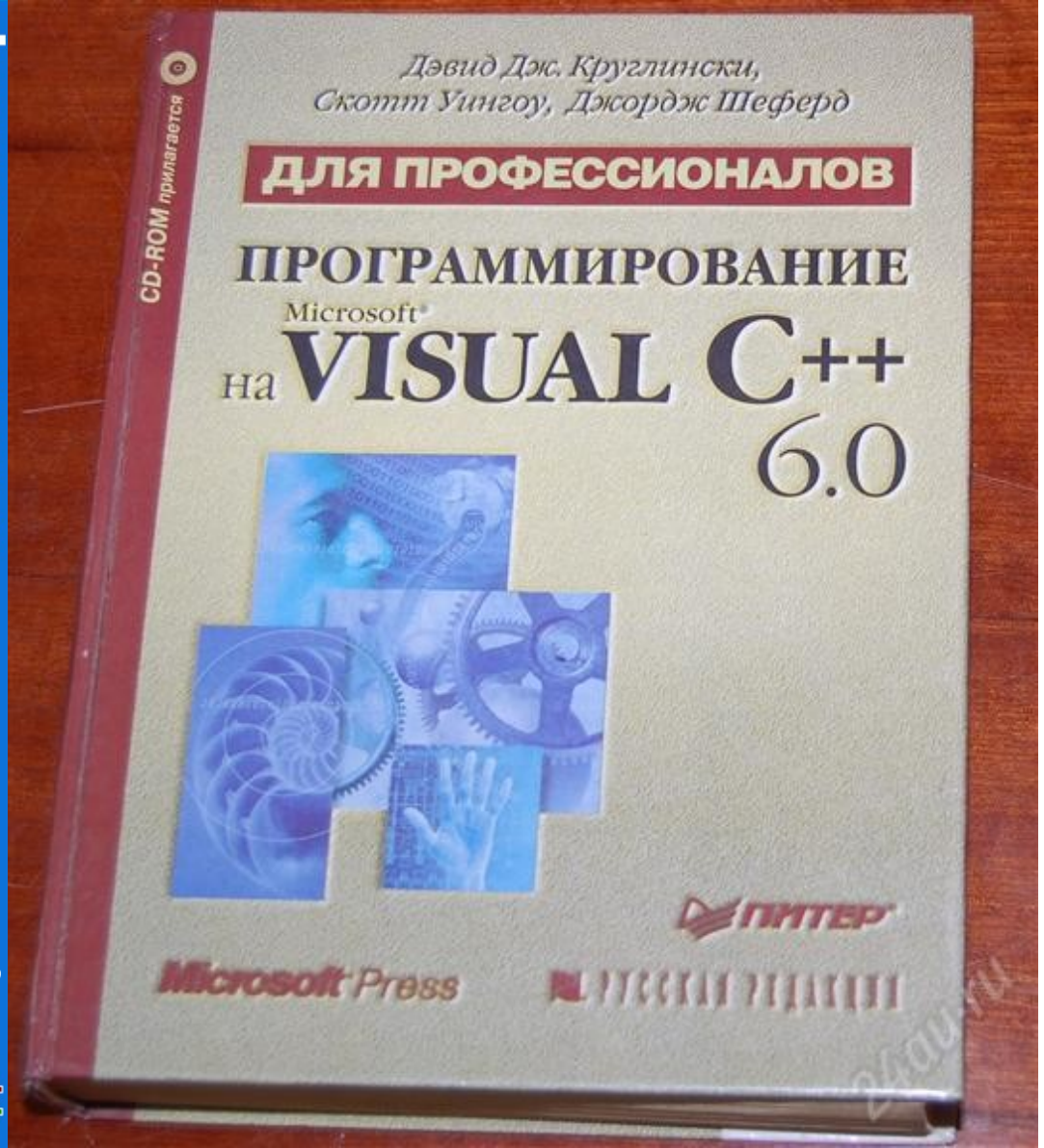

Система
Плагин 1
Плагин 2
Плагин 3

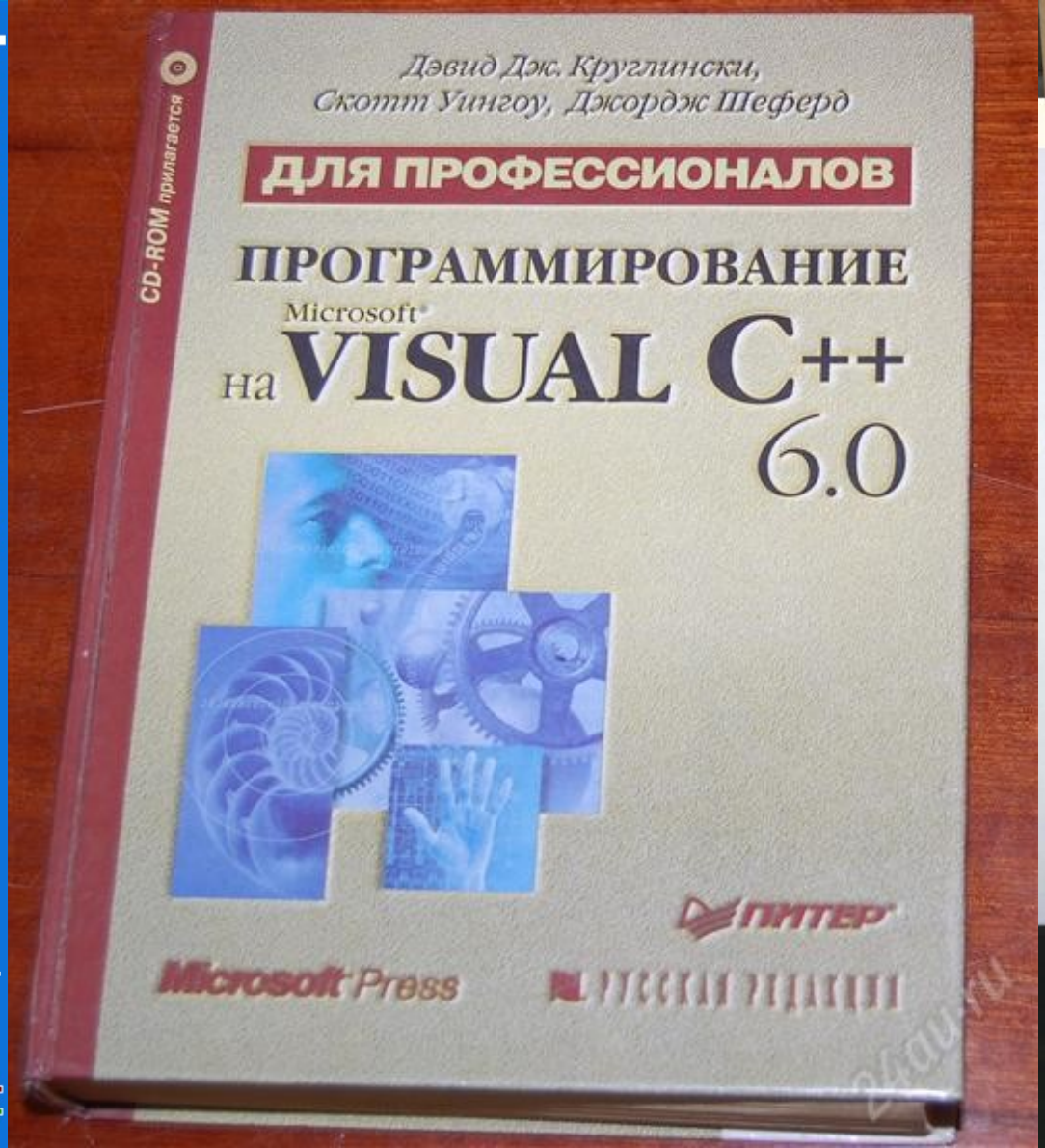

Система
Плагин 1
Плагин 2
Плагин 3
Плагин 4

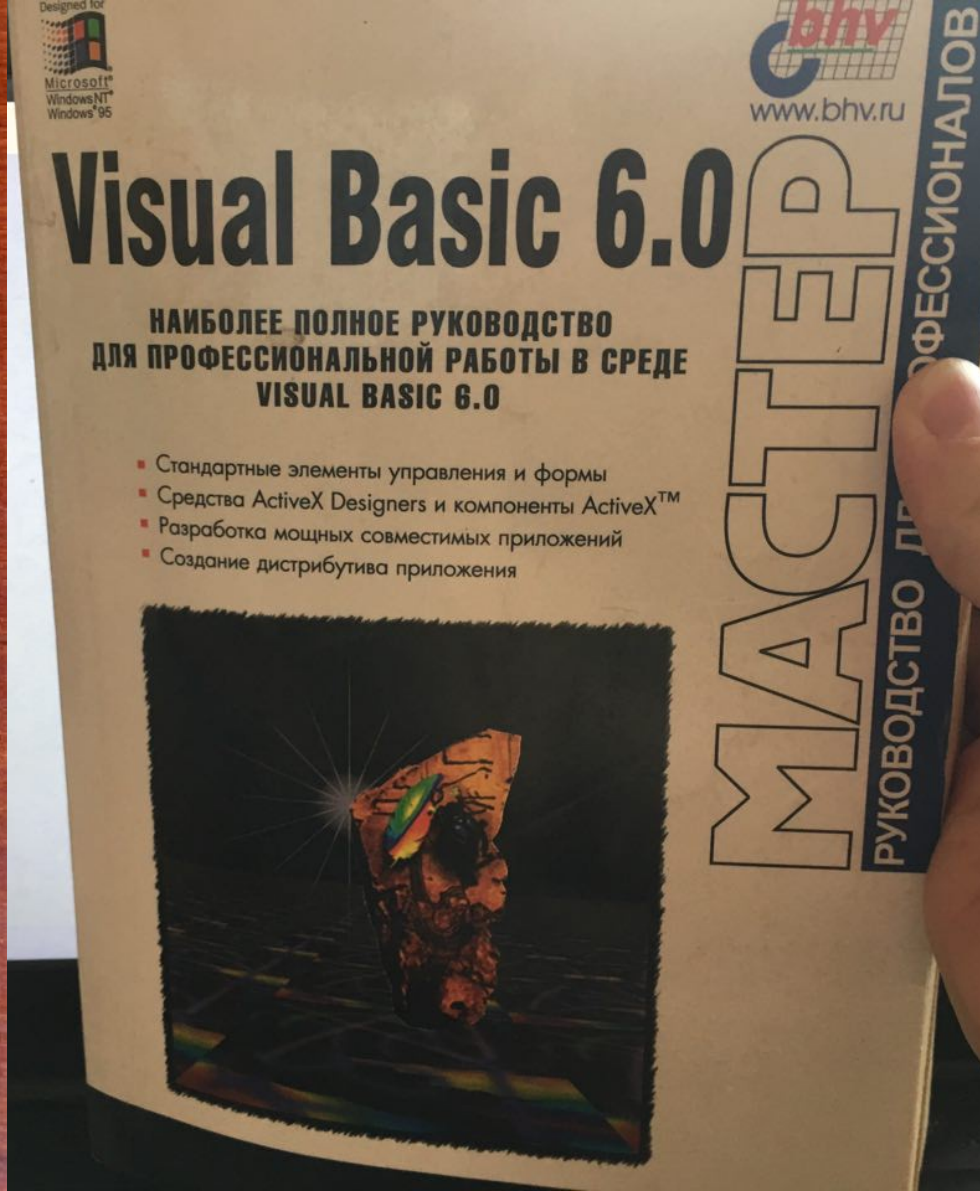

Система
Плагин 1
Плагин 2
Плагин 3
Плагин 4
C#
C#
C#


Система
Плагин 1
Плагин 2
Плагин 3
Плагин 4
C#
C#
C#
C++


Система
Плагин 1
Плагин 2
Плагин 3
Плагин 4
C#
C#
C#
C++
macOS


Дэвид Дж. Круглински,
Скотт Уингоу, Джордж Шеферд
ДЛЯ ПРОФЕССИОНАЛОВ
CD-ROM прилагается
ПРОГРАММИРОВАНИЕ
Microsoft
на VISUAL C++
6.0
ПИТЕР
Microsoft Press


Дэвид Дж. Круглински,
Скотт Уингоу, Джордж Шеферд
CD-ROM прилагается
ДЛЯ ПРОФЕССИОНАЛОВ
ПРОГРАММИРОВАНИЕ
Microsoft
на VISUAL C++
6.0
ПИТЕР
Microsoft Press


Designed for
Microsoft
Windows NT
Windows 95
bhv
www.bhv.ru
Visual Basic 6.0
НАИБОЛЕЕ ПОЛНОЕ РУКОВОДСТВО
ДЛЯ ПРОФЕССИОНАЛЬНОЙ РАБОТЫ В СРЕДЕ
VISUAL BASIC 6.0
Стандартные элементы управления и формы
Средства ActiveX Designers и компоненты ActiveX™
Разработка мощных совместимых приложений
Создание дистрибутива приложения
МАСТЕР
РУКОВОДСТВО ДЛ
ОФЕССИОНАЛОВ

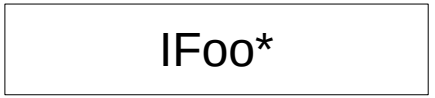
А давайте попробуем использовать нечто, что поймут как C#, так и C++?

А давайте попробуем использовать нечто, что поймут как C#, так и C++?

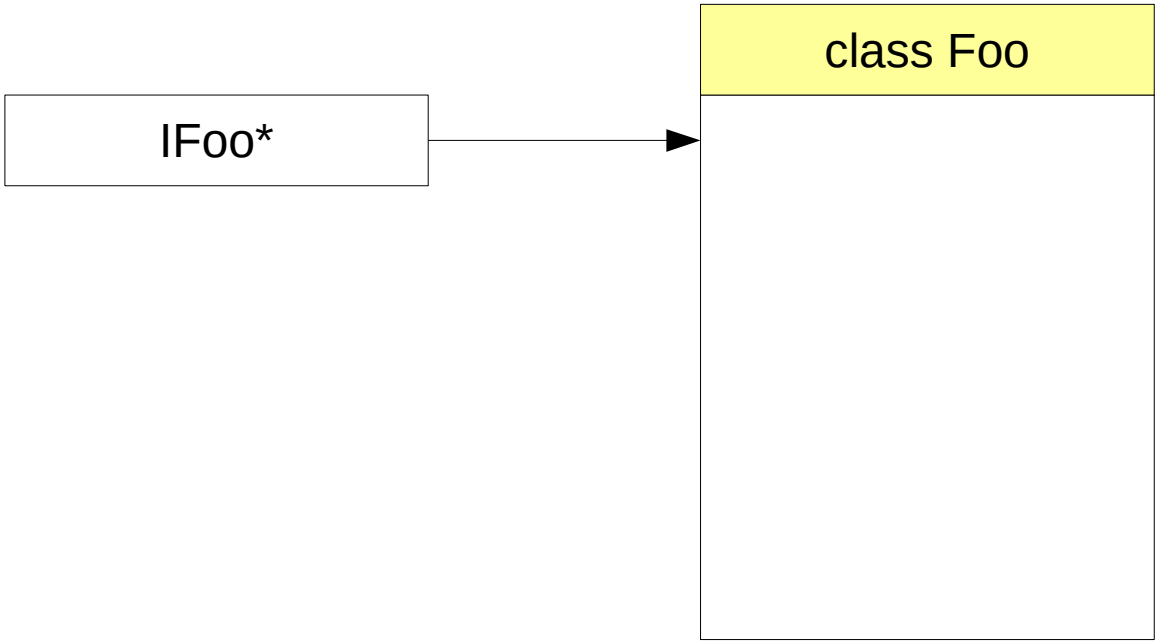
- Этим «нечто» будет интерфейс

Демо

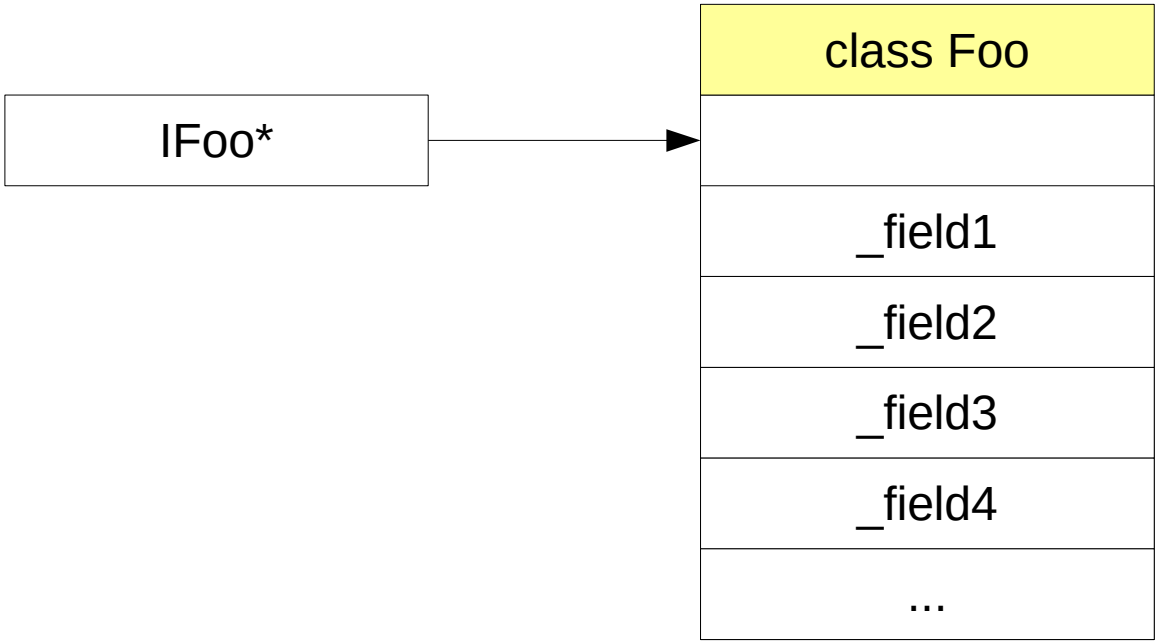
IFoo*



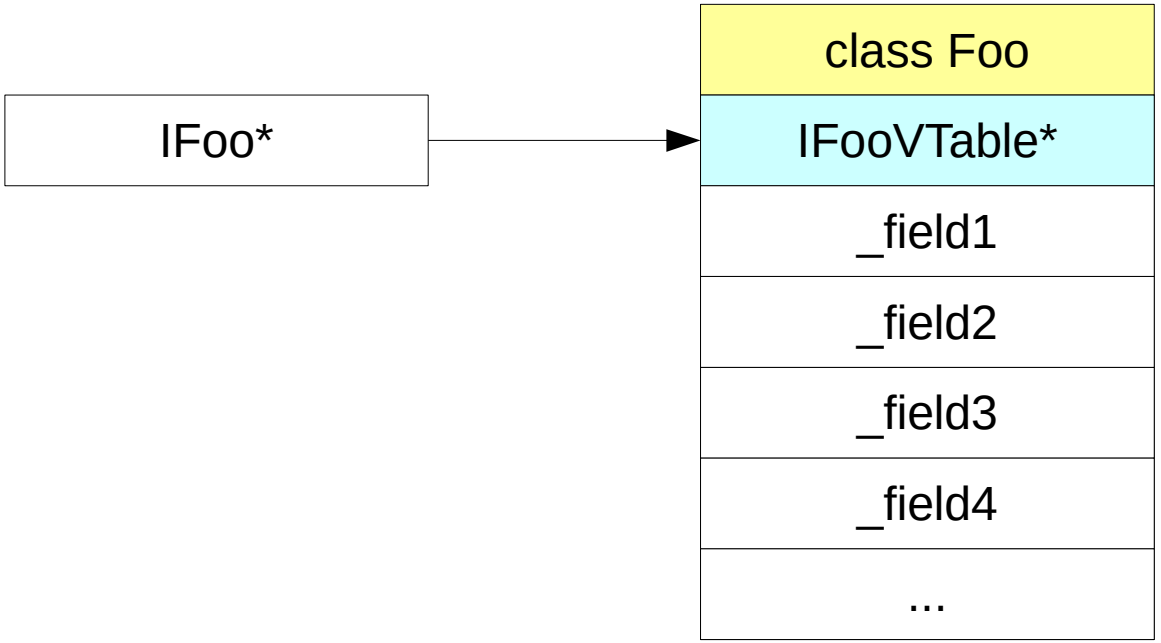
IFoo*



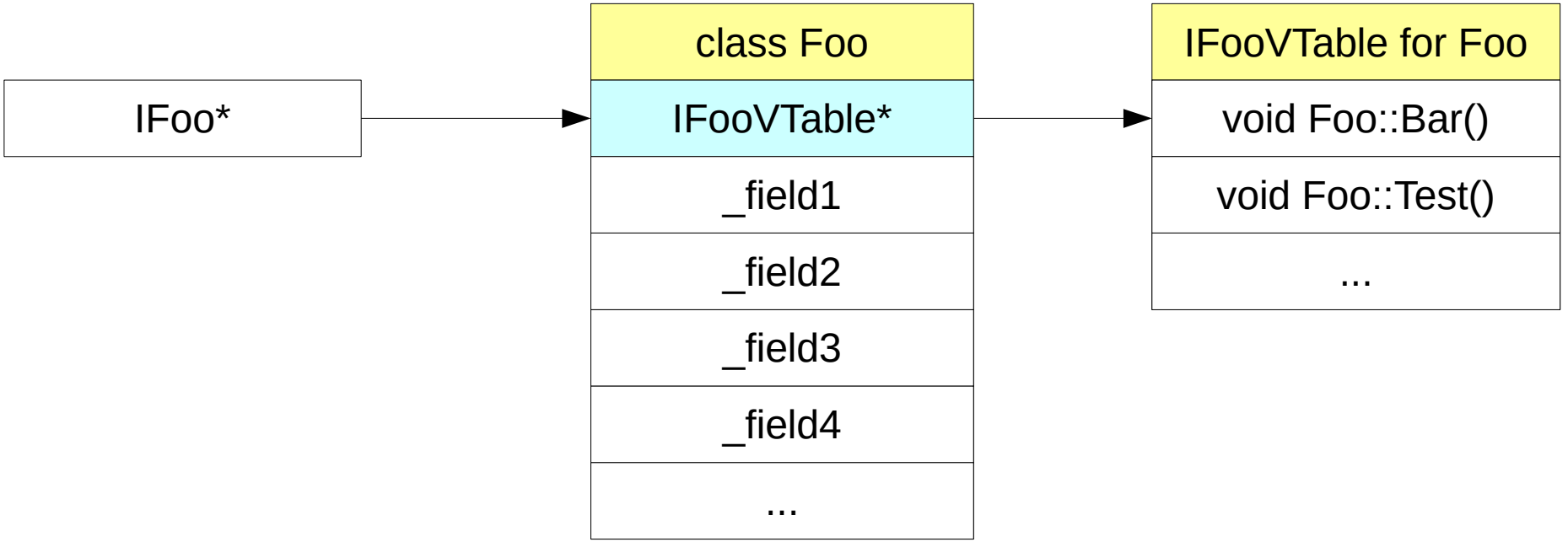
IFoo*



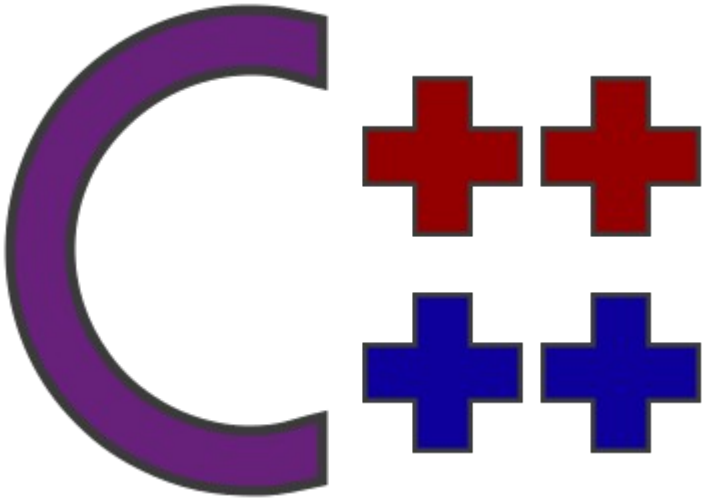
IFoo*



IFoo*


IFoo*
class Foo
IFooVTable*
_field1
_field2
_field3
_field4
...
IFooVTable for Foo
void Foo::Bar()
void Foo::Test()
...

Продолжаем демо



Остались 2 проблемы

- Не понятно, как управлять памятью
- Не понятно, как кастовать интерфейсы

Reference counting

```
void Obtain();
void Release();
```


Специальный метод для каста

```
void* GetInterface(int id);
```


Базовый интерфейс

```
struct IInteropBase
{
  virtual InteropBase* GetInterface(int id) = 0;

  virtual void Obtain() = 0;

  virtual void Release () = 0;
};
```


Вперед в прошлое

IUnknown

```
struct IUnknown
{
  virtual HRESULT STDMETHODCALLTYPE QueryInterface(
          REFIID riid,
          void **ppvObject) = 0;

  virtual ULONG STDMETHODCALLTYPE AddRef( void) = 0;

  virtual ULONG STDMETHODCALLTYPE Release( void) = 0;

};
```


Демо

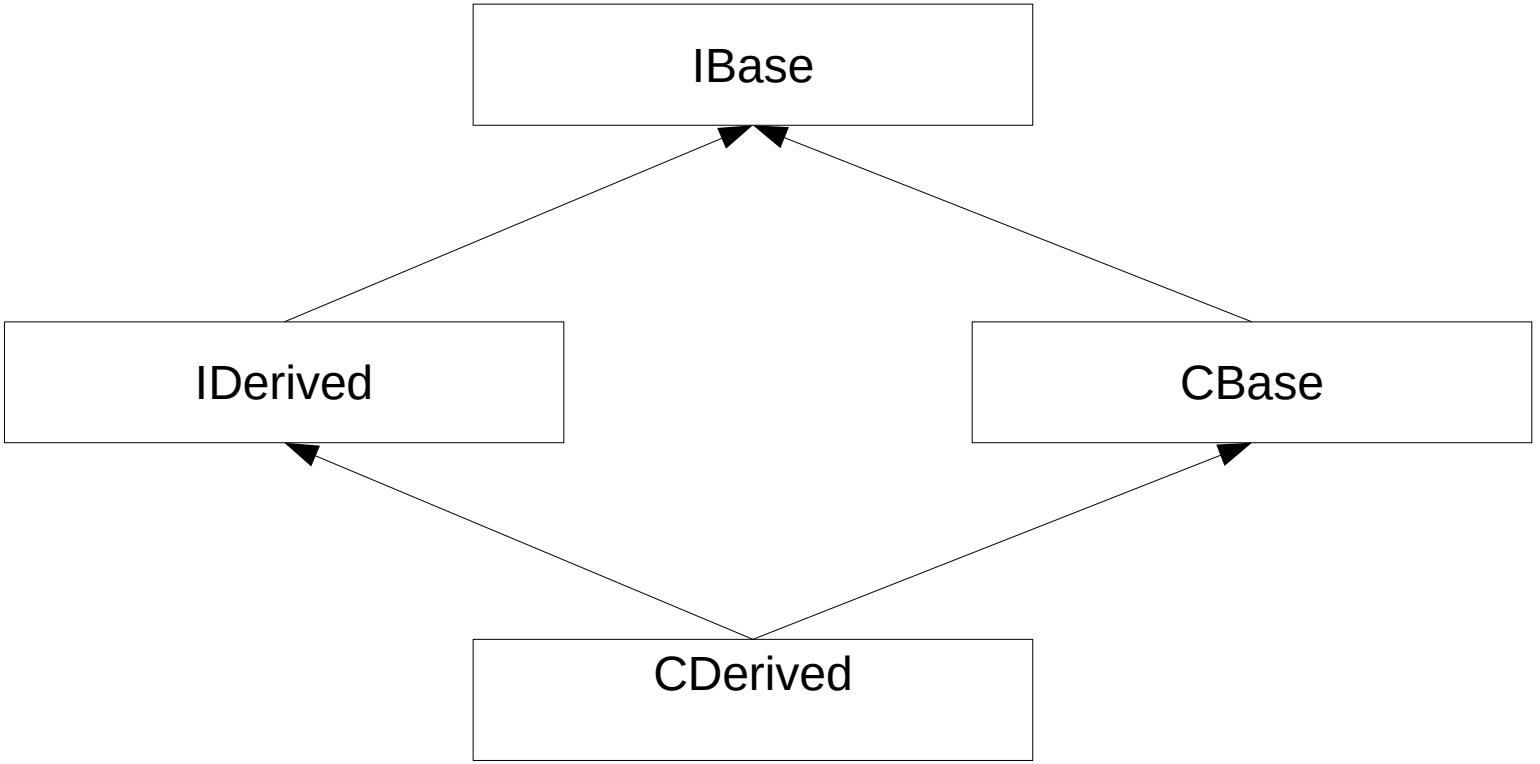
На кой нам FORWARD_IUNKNOWN()?

```
struct IBase {
  virtual void BaseMethod() = 0;
};

class CBase : public IBase {
public:
  virtual void BaseMethod() override {
      /* Do something */
    }
};
```

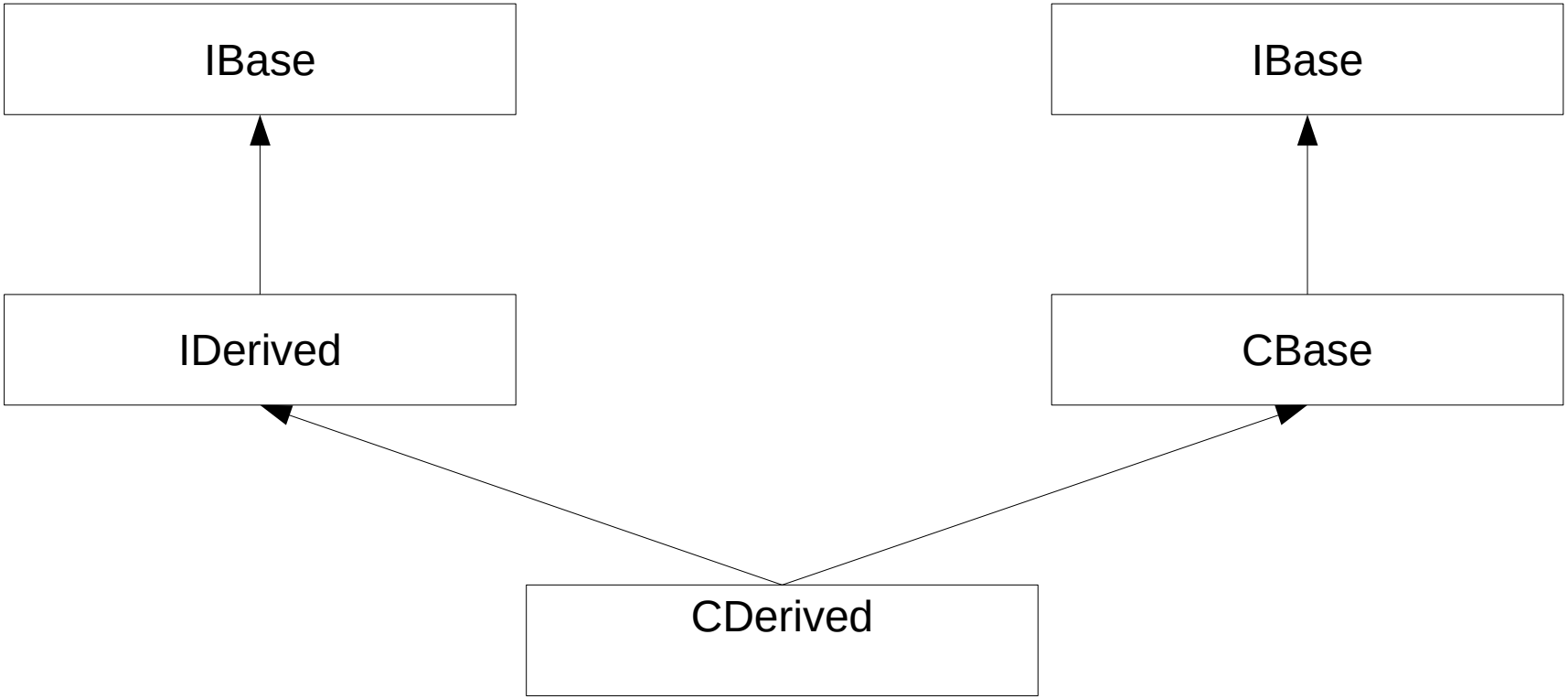

На кой нам FORWARD_IUNKNOWN()?

```
struct  IDerived : IBase {
   virtual void DerivedMethod() = 0;
};

class CDerived : public IDerived, public CBase {
public:
   virtual void DerivedMethod() override {
     /* Do something */
    }
};
```

На кой нам FORWARD_IUNKNOWN()?



На кой нам FORWARD_IUNKNOWN()?



На кой нам FORWARD_UNKNOWN()?

```
struct IBase {
  virtual void BaseMethod() = 0;
};

class CBase : public virtual IBase {
public:
  virtual void BaseMethod() override {
      /* Do something */
    }
};
```

На кой нам FORWARD_IUNKNOWN()?

```
struct  IDerived : virtual IBase {
   virtual void DerivedMethod() = 0;
};

class CDerived : public IDerived, public CBase {
public:
   virtual void DerivedMethod() override {
     /* Do something */
    }
};
```


SIGSEGV while executing native code. This usually indicates
fatal error in the mono runtime or one of the native libraries
used by your application.
/proc/self/maps:
41329000-41399000 rwxp 00000000 00:00 0
55e9aa684000-55e9aaad0000 r-xp 00000000 08:01 2229399
55e9aaccf000-55e9aacd6000 r--p 0044b000 08:01 2229399
55e9aacd6000-55e9aacdb000 rw-p 00452000 08:01 2229399
55e9aacdb000-55e9aacf2000 rw-p 00000000 00:00 0
55e9ac414000-55e9ac669000 rw-p 00000000 00:00 0
7f73f4000000-7f73f44f1000 rw-p 00000000 00:00 0
7f73f44f1000-7f73f8000000 ---p 00000000 00:00 0
7f73fc000000-7f73fc021000 rw-p 00000000 00:00 0
7f73fc021000-7f7400000000 ---p 00000000 00:00 0
7f74019a3000-7f7401b1c000 r-xp 00000000 08:01 2230523
7f7401b1c000-7f7401d1c000 ---p 00179000 08:01 2230523
7f7401d1c000-7f7401d26000 r--p 00179000 08:01 2230523
/usr/bin/mono-sgen
/usr/bin/mono-sgen
/usr/bin/mono-sgen
[heap]
/usr/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gn
/usr/lib/x86_64-linux-
/usr/lib/x86_64-linu

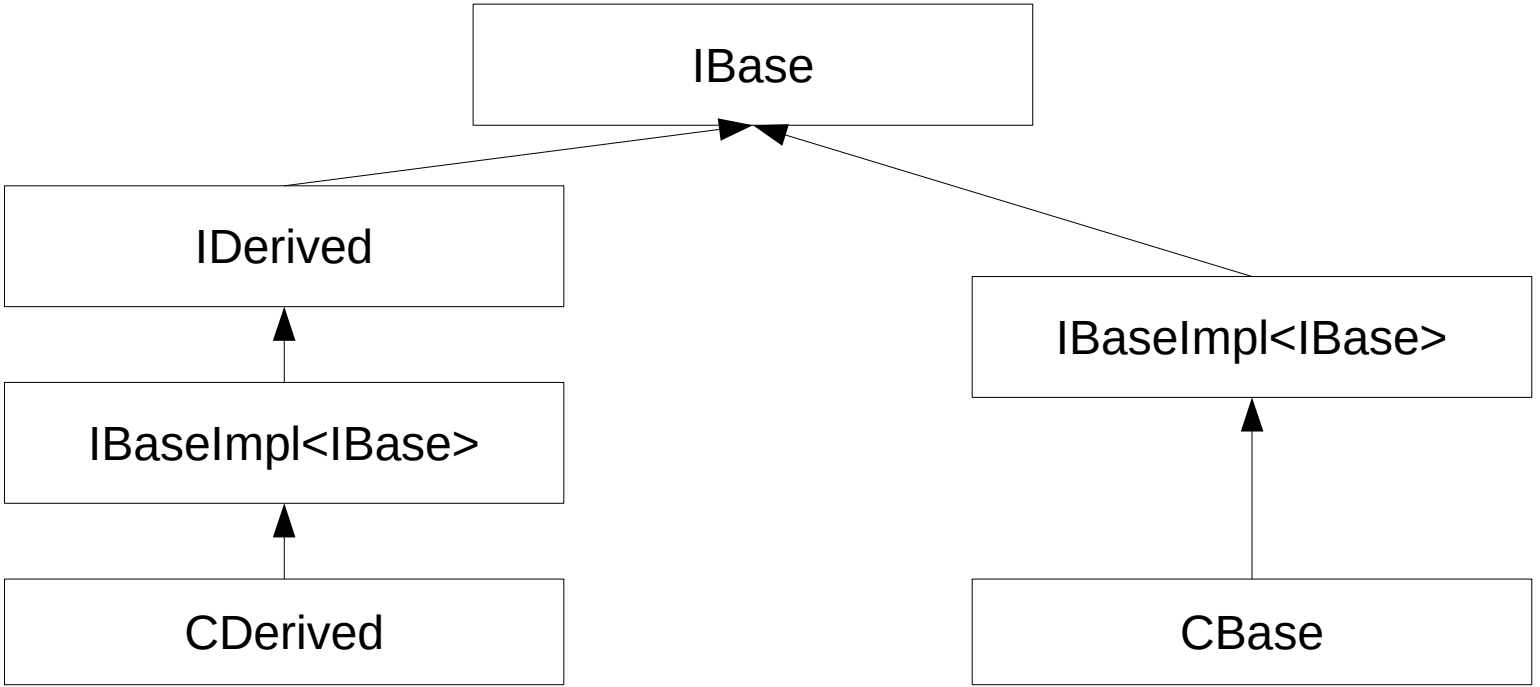
На кой нам FORWARD_IUNKNOWN()?

```
class CDerived : public IDerived, public CBase {
public:
    virtual void BaseMethod() override
    {
        CBase::BaseMethod();
    }
    virtual void DerivedMethod() override
    {
        /* Do something */
    }
};
```


ATL делает так:

```
template <typename Itf>
class IBaseImpl : public Itf {
public:
    void BaseMethod() override;
};

class CBase : public IBaseImpl<IBase> {};

class CDerived : public IBaseImpl<IDerived> {
public:
    void DerivedMethod() override;
};
```

ATL делает так:



Ист: https://stackoverflow.com/q/42576982

Продолжаем демо

Проблемы встроенной поддержки COM

Проблемы встроенной поддержки COM

- Неудобно освобождать unmanaged-объекты

Проблемы встроенной поддержки COM

- Неудобно освобождать unmanaged-объекты
- Нет уведомления об ненужности managed-объекта

Проблемы встроенной поддержки COM

- Неудобно освобождать unmanaged-объекты
- Нет уведомления об ненужности managed-объекта
- Не поддерживается в CoreCLR на не-Windows

Проблемы встроенной поддержки COM

- Неудобно освобождать unmanaged-объекты
- Нет уведомления об ненужности managed-объекта
- Не поддерживается в CoreCLR на не-Windows
- Тормозит

sharpdx

Демо

Ограничения

- Нет событий, придётся работать как на Java (в тот момент, когда в COM добавили события всё покатилось в трубу)
- Неудобно делать иерархии наследования интерфейсов и классов в C++-коде

Где используется?

Где используется?

- Ну, например, мхм, Visual Studio

Где COM используется на *nix?

Где COM используется на *nix?

- Отладчик CoreCLR (libdbgshim.so/dylib)

Где COM используется на *nix?

- Отладчик CoreCLR (libdbgshim.so/dylib)
- Avalonia.Native (OSX-бэкэнд)

Спикер

Никита Цуканов
nikita.d.tsukanov@gmail.com
Telegram: kekekeks

Примеры

https://github.com/kekekeks/dontext2019spb

SharpGenTools

https://sharpgentools.readthedocs.io/en/latest/
https://github.com/SharpGenTools/SharpGenTools/