

Оникийчук Антон

ВВЕДЕНИЕ В MANAGED C++

Возможности managed C++

- CLR – совместимый диалект C++
- Возможность использовать как native C++ библиотеки, так и .NET библиотеки
- Возможность хранить в managed типах ссылки на unmanaged типы и наоборот.
- Использование директивы `import` для работы с COM.

Новый синтаксис managed C++

- `ref` – перед классом означает что класс `managed` (`public ref MyClass`)
- `gcnew` – оператор для создания объектов `managed` типов (`throw gcnew Exception();`)
- `^` - показывает что надо использовать ссылку на `managed` объект (`System::String^ helloString`)

СВОЙСТВА

```
property System::String^ HelloText{
    System::String^ get(){
        return helloString;
    }
    System::Void set(System::String^ str)
    {
        helloString = str;
    }
};
```

СОБЫТИЯ

```
System::AsyncCallback^ callBack;  
event System::AsyncCallback^ AsyncCallback{  
    System::Void add(System::AsyncCallback^ src)  
    {  
        callBack+=src;  
    }  
    System::Void remove(System::AsyncCallback^ src)  
    {  
        callBack-=src;  
    }  
private:  
    System::Void raise(System::IAsyncResult^ res)  
    {  
        callBack(res);  
    }  
};
```

И что? 😊

- ⦿ Возможность работать с COM объектами максимально контролируя взаимодействие
- ⦿ Возможность переводить иерархии COM объектов в .NET иерархии разумно, с использованием только 1й библиотеки.
- ⦿ Возможность использовать C++/C библиотеки в .NET коде без выяснения сигнатур.

Но как? Просто!

Используем COM

- Директива `#import` – автоматическое построение C++ классов на основе библиотеки типов (tlb)
- Переименовываем конфликтующие имена
- Импорты должны идти в строгом порядке – сначала те от которых зависят остальные, потом зависимые

Используем stl

- Не забываем о работе с памятью (класс `System::Runtime::InteropServices::Marshal`)
- Не используем `managed` типы в параметрах шаблонов

Кое что о производительности

- Вызов через C++ на 10-30 машинных инструкций более производителен чем `pthread_t`

Минусы

- Обязательный ручной маршалинг данных.
- Сложность и громоздкость кода.
- Местами очень неявные ошибки компиляции и рантайма.