**4. Внешний интерфейс CLI**

Common Language Infrastructure

Разработан Microsoft. Менее безопасный чем JNI. В C# и C++ есть «небезопасное» множество размером примерно с С (?)

* Прямое управление раскладкой полей объекта
  + Уменьшает гибкость системы
* Закрепление объекта в памяти и получение указателя
* Уменьшает гибкость менеджера памяти
* Потенциальная угроза безопасности
* Прямая работа с указателями из управляемого кода
* Потенциальная угроза безопасности
* pinvoke (.NET)
  + Связывание произвольной внешней функции
  + копирование объектов с изменением раскладки полей

Функции, реализованные в неуправляемых библиотеках DLL, могут вызываться из управляемого кода с помощью вызова неуправляемого кода P/Invoke. Если исходный код библиотеки DLL недоступен, вызов P/Invoke является единственным вариантом обеспечения взаимодействия. В отличие от других языков .NET в Visual C++ предусматривается альтернатива вызову P/Invoke.

Все совместимые языки компилируются в Common Intermediate Language(CIL), который является промежуточным языком, абстрагированным от платформенного «железа». Во время исполнения кода CIL будет перекомпилирован в машинный язык, соответствующий конкретному «железу» и ОС.

Промежуточное представление(IR):

* Язык регистровой машины
  + На 45% меньше операций доступа к памяти
  + Возможность оптимизации на уровне IR
  + Используется в статических и оптимизирующих JIT компиляторах
* Язык стековой машины
  + На 25% более компактен
  + Сохраняет семантику исходного кода (проще восстановить исходный код)
  + Проще для интерпретации и шаблонной компиляции
* Используется в JBC(Java Byte Code) и CIL

Оба языка полны по Тьюрингу (эквивалентны машине Тьюринга)

*Сравнение CIL и JBC*

* CIL по семантике расширение JBC
  + Добавлены целые без знака
  + Поддержка рефлексии (ldtoken)
  + Размещение памяти на стеке (localloc)
  + Работа с внутренними ссылками
  + Работа с указателями
  + обертывание (box/unbox)

+

Common Language Infrastructure (CLI) enables an application program written in any of several commonly-used programming languages to be run on any operating system using a common runtime program rather than a language-specific one.

Common Language Infrastructure provides a virtual execution environment comparable to the one provided by Sun Microsystems for Java programs. In both environments, CLI and Java use a compiler to process language statements (also known as source code) into a preliminary form of executable code called bytecode. Later, when a program is run, its bytecode is compiled on the fly into the native code required by the machine architecture of a given computer. Common Language Infrastructure has been accepted as an open standard by ECMA, an international organization for the promotion of technology standards. CLI is designed to make it easier to develop programs that can be used as Web services.