

# Co nowego pojawi o si ostatnio?

Referat na laboratoria z **Teorii Kompilacji**  
**Akademia Górniczo Hutnicza 2006**

## **Autorzy**

Mateusz Makowiec

Artur Andrzejewski

# Omawiane Zagadnienia

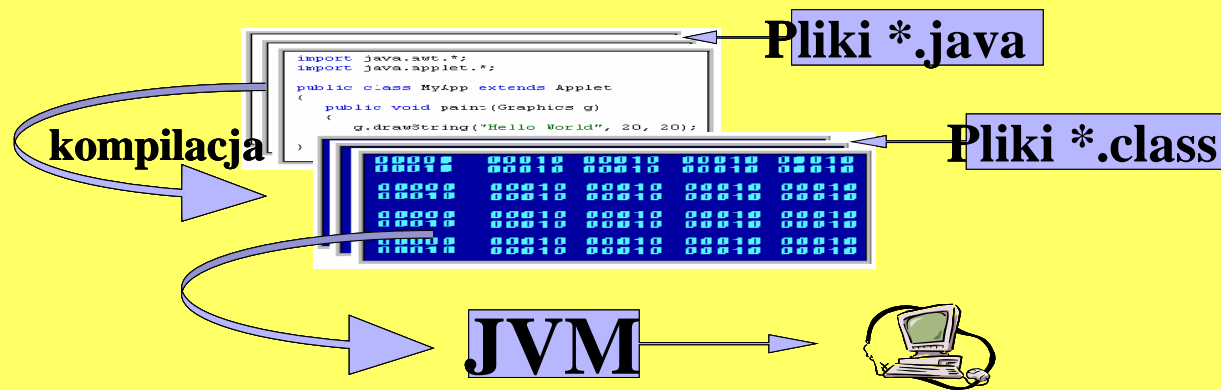
- Java Platform
- .NET Framework
- Kompilatory dla kart graficznych
- Narzędzia do optymalizacji kodu
- Narzędzia do generacji parserów
- Odnośniki

# Java Platform

- Java Platform – platforma Javy, jest to środowisko firmy Sun Microsystems dzięki któremu możemy tworzyć, rozwijać i uruchamiać aplikacje napisane w Javie.
- Kod programu (plik \*.java) jest kompilowany do kodu pośredniego (plik \*.class) (ang. bytecode). Dzięki temu powstały program jest niezależny od systemu operacyjnego. Jego wykonaniem zajmuje się wirtualna maszyna Javy (JVM).
- Wirtualna maszyna Javy (ang. *Java Virtual Machine*) – system dzięki któremu możemy uruchomić programy napisane w Javie.

# Java Platform - model wykonania kodu

- Kod po redni (bytecode) jest wykonywany przez JVM która t umaczy ten kod na kod dopasowany do konkretnego systemu operacyjnego.



- Cz sto JVM uzupe nia si o kompilator JIT (ang. Just-In-Time ), wynika to z faktu szybko ci wykonywania kodu skompilowanego do kodu maszynowego , a interpretacji kodu po redniego przez JVM

# Java Platform - JIT

- Kompilacja JIT (ang . Just-In-Time ) to metoda wykonywania programów polegaj ca na kompilacji do kodu maszynowego w locie, czyli bezpo rednio przed wykonaniem danego fragmentu kodu JVM decyduje które fragmenty kodu kompilowa (z regu y te które si mog powtarza np. wn trza p tle) a które fragmenty interpretowa .

# Java Platform – JVM

- **Różne implementacje JVM :**
- JDK – firmy Sun zawiera szereg narzędzi do tworzenia oprogramowania (w odróżnieniu od JRE) np.: kompilator czy debugger. aktualna wersja : JDK 5.0
- Jikes – zapoczątkowany przez IBM , aktualnie rozwijany przez ochotników (open source) , aktualna wersja : Jikes 1.22 wydana w 3 października 2004 , posiada dodatki np.: generator parserów (2001)
- JRockit – optymalizowane pod względem szybkości pod procesory Intela (Itanium oraz Xeon EMT64) autorstwa firmy Bea, aktualna wersja JRockit 5.0 R26 JDK

# Narzędzia powiązane z Java

- **AMPC** (ang. Axiomatic Multi-Platform C) – środowisko które generuje bytecode na podstawie programu napisanego w C. Dostępny na różne systemy operacyjny (Linux, MS Windows, Mac OS),
  - strona domowa <http://www.axiomsol.com/>
  - aktualna wersja to 1.3.7 wydana 20 stycznia tego roku
  - komercyjne narzędzie (99\$)
- **Excelsior JET** - narzędzie służące do przetworzenia plików Javy (\*.class) do samodzielnych plików wykonalnych dla MS Windows lub Linux.
  - aktualna wersja 4.1 15 grudnia 2005
  - strona domowa <http://www.excelsior-usa.com/home.html>
- - komercyjne narzędzie (cena tak od 240\$ do 700\$ w zależności od dodatków)

# Narzędzia do optymalizacji

- **JMPC** – free java class compresor – narzędzie optymalizuje bytecode oraz kompresuje pliki class.
  - optymalizuje kod redukując niepotrzebne instrukcje load/store oraz goto
  - zmniejsza nazwy metod i pól
  - skraca nazwy klas
  - obcina informacje dla debugera

Ostatnia wersja została wydana: 1 czerwca 2005

Strona domowa: <http://viscomp.utdallas.edu/FACADE/jcmp/>

**Przykłady użycia:**

```
java -jar jcmp.jar test.class
```

```
java -jar jcmp.jar -jars a.jar b.jar
```



# Narzędzia do optymalizacji

- **J.CLASS OPTIMIZER** — optymalizuje codebyte
  - usuwa nieużywane pola, metody, stringi, stałe, atrybuty
  - optymalizuje bytecode
  - usuwa informacje dla debugerastrona domowa: <http://www.uni-vologda.ac.ru/~c3c/jco/index.html>
- Komercyjne : licencja kosztuje 29\$
- **Przykładowe użycie:**  
C:\jdk1.2\demo\applets\Animator>jco \*.class -max

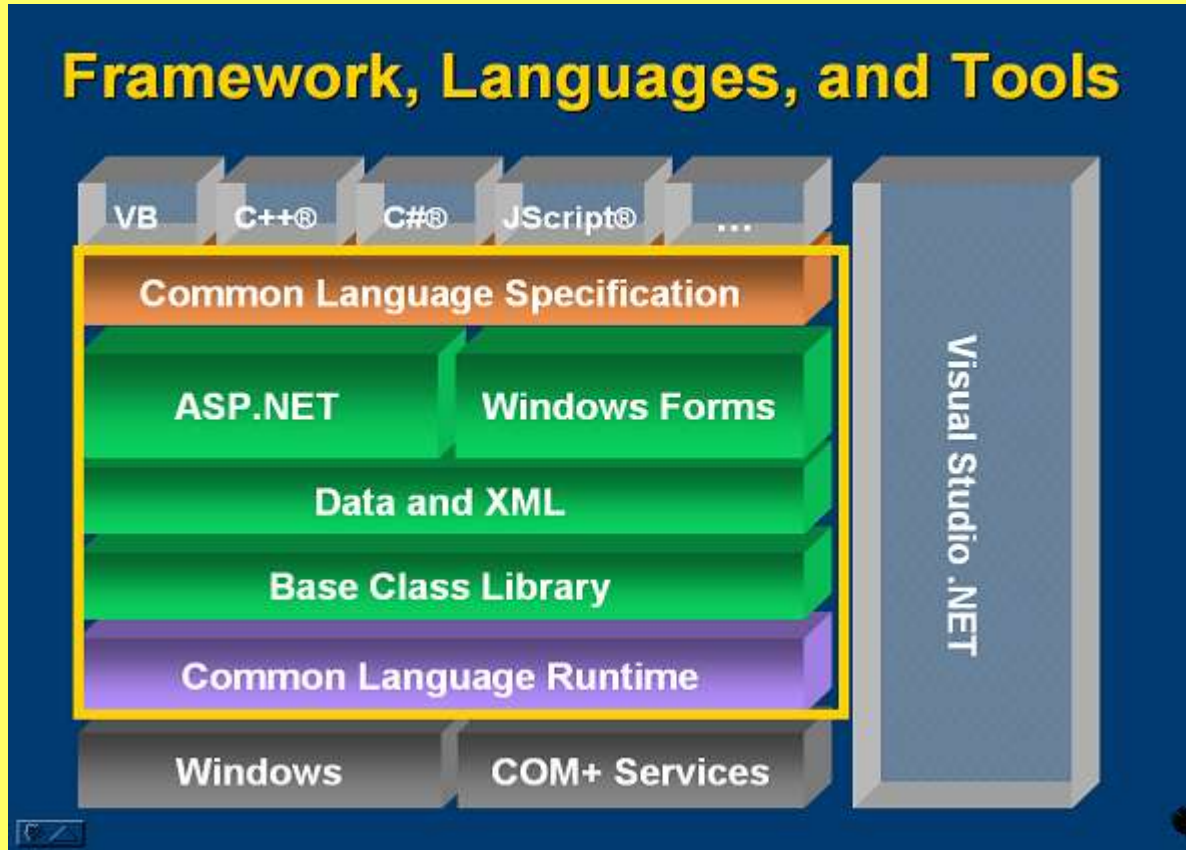


# .NET Framework

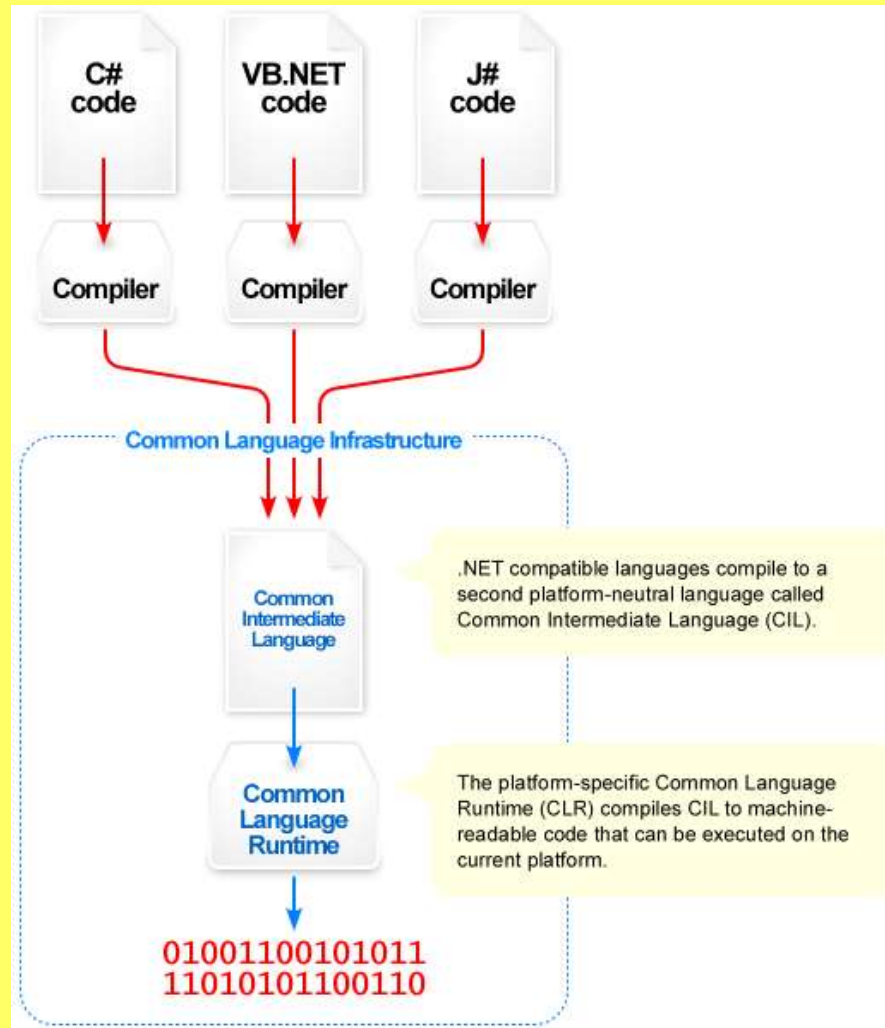
- .NET to platforma pozwalająca na współpracę dowolnej ilości języków programowania w zależności od sprzętu i systemu operacyjnego. Opisana standardem CLI (Common Language Infrastructure)
- Wersja 1.0 została stworzona przez Microsoft w 2002 roku, jako odpowiedź na technologii Java i dołączona do Visual Studio 2002.
- Wersja 2.0 (Ukazała się w Październiku 2005) jest dołączona do Visual Studio 2005. Od tej wersji zapewnia pełne wsparcie 64 bitowych architektur. Duża część kodu MS Windows Vista jest pisana w C# 2.0
- Wykorzystuje podstawowy język CIL (Common Intermediate Language) oraz środowisko uruchomieniowe CLR (Common Language Runtime)
- Wiele gotowych front-endów, na przykład: C#, J#, Managed C++, VB, JScript (Microsoft), Python, Perl, COBOL, Oberon, Haskell, Eiffel, Scheme (Inne firmy)



# .NET Framework



# .NET Model wykonywania kodu



# .NET Common Intermediate Language

- Po redni j zyk do którego kompilowane s wszystkie j zyki .NET, zamieniany bezpo rednio na bytecode. Pierwotnie znany jako MSIL (Microsoft Intermediate Language), wraz z wprowadzeniem CLI przemianowany na CIL.

Przyk adowy kod (Hello World) w CIL:

```
.method private hidebysig static void Main() cil managed
{
    .entrypoint
    // Code size          11 (0xb)
    .maxstack 8
    IL_0000: ldstr          "Hello, world!"
    IL_0005: call           void [mscorlib]System.Console::WriteLine(string)
    IL_000a: ret
} // end of method HelloWorld::Main
```

# .NET Common Language Runtime

- Pełni rolę wirtualnej maszyny, zarządza kodem podczas jego wykonywania
- Zarządza pamięcią (Managed Heap, Garbage Collection)
- Zarządza wątkami
- Kompiluje kod CIL *w trakcie* uruchomienia programu (JIT : Just-In-Time) lub *przed* (NGEN : Native Image Generator).
- Zapewnia bezpieczeństwo (rzutowanie typów danych, indeksowanie tablic, ..)

# .NET Assembly (Podzespó )

- To pakiet zawieraj cy kod CIL (W implementacji MS dla Windows przechowywany jako PE (Portable Exe))
- Posiada manifest zawieraj cy nazw podzespó u (mo e by silna, wtedy podzespó posiada unikalny id dla systemu), numer wersji, ustawienia regionalne, list wymaganych podzespó ów z ich numerami wersji i list plików tworz cych podzespó wraz z ich CRC
- Posiada metadane opisuj ce typy znajduj ce si w pakiecie (Co umo liwia u ycie mechanizmu refleksji dla podzespó u)
- Wyró niamy dwa rodzaje podzespó ów: process assemblies (EXE) i library assemblies (DLL)
- Podzespó mo e si sk ada z wielu plików, nazywanych modu ami
- Mo e by prywatny lub dzielony przez wiele aplikacji

# .NET Base Class Library (BCL)

- Standardowa biblioteka dostępna dla wszystkich języków w obrębie .NET Framework. Poniżej lista przestrzeni BCL.

**Collections**

**Configuration**

**Diagnostics**

**Globalization**

**IO**

**Net**

**Resources**

**Reflection**

**Security**

**ServiceProcess**

**Text**

**Threading**

**Runtime**

**InteropServices**

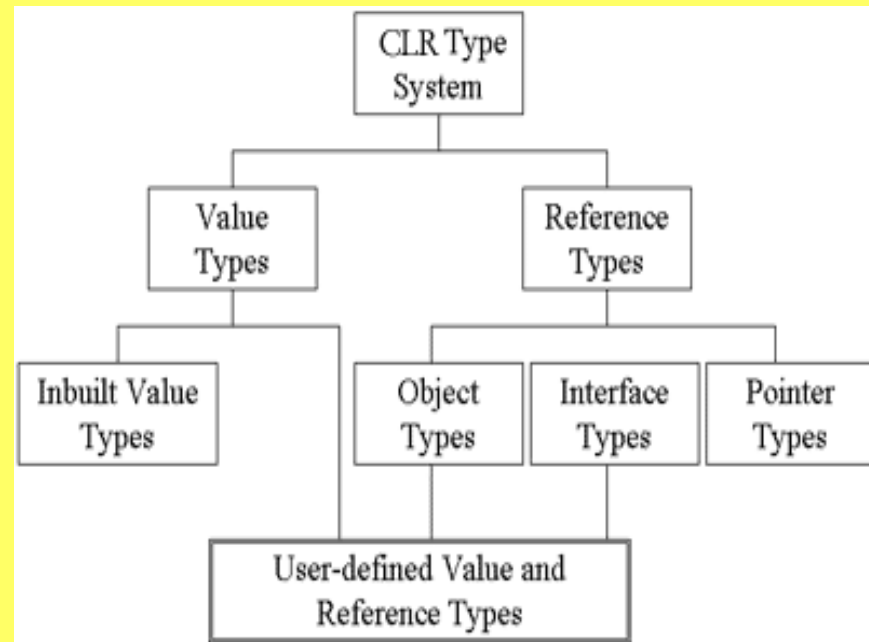
**Remoting**

**Serialization**



# .NET Common Type System (CTS)

- Wszystkie języki .NET posiadają te same typy proste
- Na poziomie binarnym typy proste są ze sobą zgodne
- Typy mogą być *wartościami* lub *referencjami*
- Brak wskaźników
- Boxing/Unboxing



# .NET CTS: Boxing/Unboxing

```
public static void Main()
{
    Int32 v = 5; // creating unboxed value type variable
    Object o = v; // o refers to boxed version of v
    v = 123; // changes the unboxed value to 123

    Console.WriteLine (v + ", " + (Int32) o); // displays "123, 5"
}
```

*Boxing* zachodzi kiedy zmiennej-referencji przypisujemy zmienną – wartość, tworzony jest wtedy nowy obiekt i przypisywana jest mu odpowiednia wartość

*Unboxing* to proces odwrotny (zachodzi tylko pod warunkiem, że referencja nie jest równa null i typy są zgodne)

# .NET Common Language Infrastructure

- Specyfikacja dostępna publicznie, standaryzowana przez ECMA (Ecma-335) oraz ISO (IEC 23271)
- Określa elementy z których składa się .NET Framework i pozwala na napisanie własnej implementacji tej platformy.
- CLI podzielono na 4 części: CTS (Common Type System), Metadata, CLS (Common Language Specification) oraz VES (Virtual Execution System)
- Dostępne implementacje to m.in. MS .NET Framework, .NET Compact Framework, Mono development platform, Portable.NET

# .NET implementacje

## **Mono Development Platform**

[http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)

Stworzony przez firmę Novell.

Licencja: GPL/LGPL/MIT

Dostępny na systemy: FreeBSD, MacOS X i Windows

## **DotGNU**

<http://www.dotgnu.org/>

Licencja: GPL+/LGPL

Dostępny na systemy: GNU/Linux (on PCs, Sparc, iPAQ, Sharp Zaurus, PlayStation 2, Xbox,...), \*BSD, Cygwin/Mingw32, Mac OS X, Solaris, AIX i Windows.

# Kompilatory dla Kart Graficznych

- Rozwój kart graficznych 3d doprowadzi do powstania i upowszechnienia programowalnych jednostek. Pierwsza karta graficzna obsługująca PS i VS to GeForce3 firmy Nvidia
- Pixel Shader: program sterujący teksturami (otrzymuje interpolowane dane od VS i dla każdego wypelnianego punktu potrafi ustawić kolor)
- Vertex Shader: program sterujący wierzchołkami (vertexami)
- Geometry Shader: program tworzący wierzchołki (DX10/SM 4.0)
- Program sterujący Shaderem, może przesyłać do niego dane. (W SM 4.0 także odebrać dane od shadera)
- DirectX 10 / Shader Model 4.0 . Pierwsze karty obsługujące nowy standard mają się pojawić w drugiej połowie 2006 (Chip G80 firmy Nvidia)

# Vertex Shader 1.1 - przyk ad

```
vs.1.1

dcl_position v0

def c40, 0.0f,0.0f,0.0f,0.0f;

;transform into projection space
m4x4 r0,v0,c8
max r0.z,c40.z,r0.z //clamp to 0
max r0.w,c12.x,r0.w //clamp to near clip plane
mov oPos,r0
add r0.w,r0.w,-c12.x
mul r0.w,r0.w,c12.y
;Load into diffuse
mov oD0.xyzw,r0.w
;load into texture
mov oT0.x,r0.w
mov oT0.y,c12.x
```

# Pixel Shader 1.1 - przyk ad

```
ps.1.1
```

```
def c1, 1.0f,0.0f,0.0f,0.0f  
def c4, 0.0f,0.0f,1.0f,0.0f  
def c2, 1.0f,1.0f,1.0f,1.0f  
def c3, 1.0f,1.0f,1.0f,1.0f  
def c0, 0.0f,0.0f,0.0f,0.0f
```

```
tex t0  
tex t1  
tex t2
```

```
sub r1.rgb,t1,t0  
+sub_x4 r1.a,t0,t1  
mov_x4 r1.a,r1.a  
mad r1,t2,r1.a,r1
```

```
dp3 t0.rgba,r1,c4
```

```
mov_x4 r1.rgb,r1  
dp3_x4 t1.rgba,r1,c1 //move into the alpha  
add_x2 r0.a,t0,t1  
+mov r0.rgb,c3
```

# HLSL (High Level Shader Language), CG (C for Graphics)

- Standard języka wysokiego poziomu dla kart graficznych
- CG i HLSL to te same języki, opracowane wspólnie, różni się jedynie nazwą: pierwsza promowana jest przez firmę Nvidia, druga przez Microsoft w pakiecie DirectX
- Składnia zbliżona do C
- Spina stany *urządzenia renderującego*, pixel i vertex shader (niebawem także geometry shader) w jednym pliku. Pozwala na tworzenie *technik*.
- Typy danych i funkcje umożliwiające łatwe prowadzenie obliczeń matematycznych (takie jak float4, czy float4x4) Struktury, tablice.



# HLSL – fragment kodu

```
float4 mainPS(vertexOutput IN,
    uniform samplerCUBE EnvironmentMap,
    uniform half reflectStrength,
    uniform half refractStrength,
    uniform half3 etas
) : COLOR
{
    half3 N = normalize(IN.WorldNormal);
    float3 V = normalize(IN.WorldView);

    // reflection
    half3 R = reflect(-V, N);
    half4 reflColor = texCUBE(EnvironmentMap, R);

//    half fresnel = fresnel(dot(N, V), 0.2, 4.0);
    half fresnel = tex2D(fresnelSampler, dot(N, V));

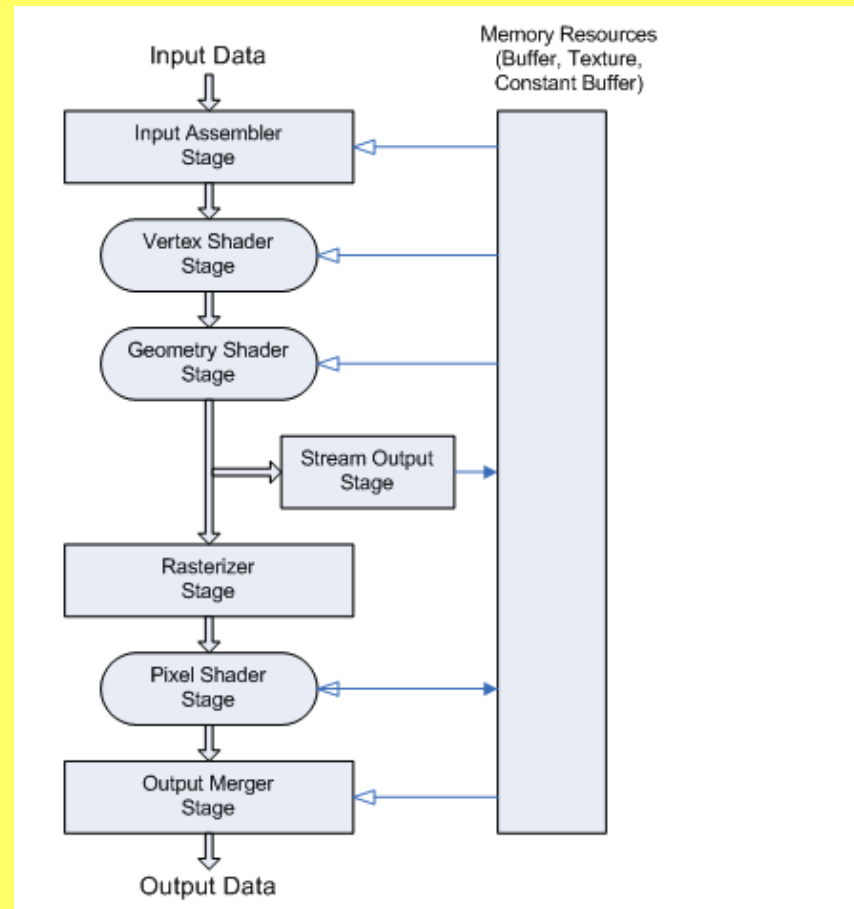
    // wavelength colors
    const half4 colors[3] = {
        { 1, 0, 0, 0 },
        { 0, 1, 0, 0 },
        { 0, 0, 1, 0 },
    };

    // transmission
    half4 transColor = 0;
    bool fail = false;
    for(int i=0; i<3; i++) {
        half3 T = refract2(-V, N, etas[i], fail);
        transColor += texCUBE(EnvironmentMap, T) * colors[i];
    }

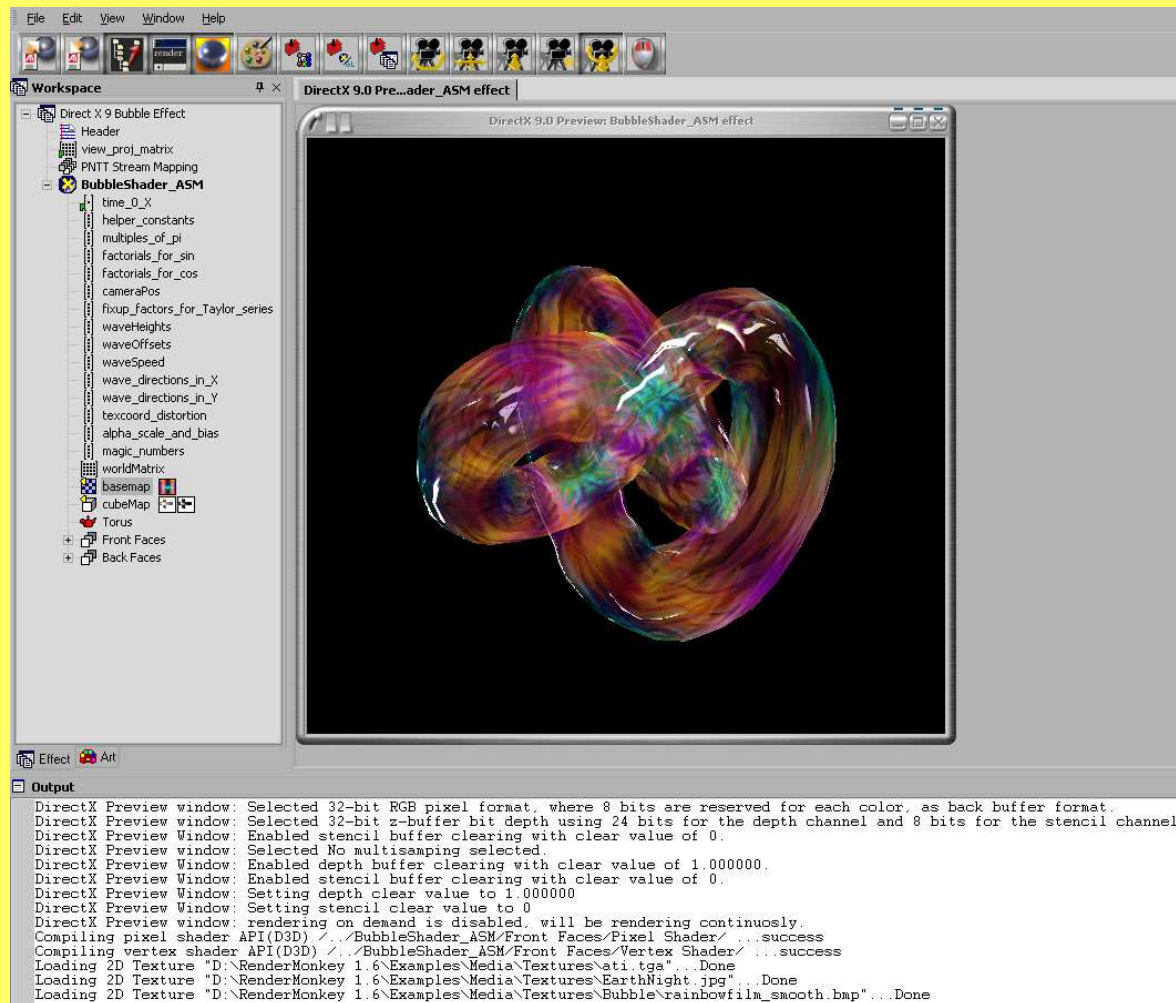
    return lerp(transColor*refractStrength, reflColor*reflectStrength, fresnel);
}
```

# Geometry Shader

- Nowo która zostanie wprowadzona razem z DirectX 10
- Shader *Produkuje* wierzchołki
- Na wejściu Shader otrzymuje *prymityw* (3 punkty, 2 punkty lub punkt)



# Shadery w praktyce : RenderMonkey



# Shadery w praktyce : Unreal Engine 3



# Odniki

Nowo ci ze wiata kompilatorów: <http://www.compilers.net>

Katalog narz dzi do konstruowania kompilatorów: <http://catalog.compilertools.net/>

Konferencja CC2006: <http://www.st.cs.uni-sb.de/cc/cc2006cfp.pdf>

Porównanie .NET 2.0 i Java 1.5: <http://www.bentuser.com/article.aspx?ID=323>

MSDN Library: <http://msdn.microsoft.com/>

CG Shaders: <http://www.cgshaders.org/>

DirectX 10 SDK (Technology Preview): <http://msdn.microsoft.com/directx/sdk/>

Portal Nvidii dla developerów: <http://developer.nvidia.com/page/home.html>

Wikipedia: <http://www.wikipedia.org>

PYTANIA ?

KONIEC