

C#とは？

.NETとC#

C# は、Microsoft が提唱する .NET にアクセスする手段の一つです
.NET とは、Java 同様に OS やプラットフォームを抽象化したアプリケーションの開発を提供する新しい動作環境であり、コンパイラやクラスライブラリを提供するサービスとも言えます

この .NET にアクセスし .NET アプリケーションの開発に使用するのが C# 言語なのです
C# は C/C++ から派生しながら完成度の高いオブジェクト指向をサポートします
(オブジェクト指向については、この言語仕様を知れば理解できるでしょう)
Microsoft は、C# を C++ の威力と Visual Basic の生産性を兼ね備える言語と位置付けています

.NET アプリケーションはコンピュータではなく .NET の**仮想マシン**を動作させます
この仮想マシンを **CLR**(Common Language Runtime) と呼び、.NET の中核に位置します
そして、.NET は **BCL**(Base Class Library) と呼ばれる基底クラスをサポートします
このクラスライブラリは、ファイルやデータベース、XML など様々な機能を提供しています

MSIL

Java 言語は、コンパイルするとバイトコードを生成し
Java 仮想マシンはこのバイトコードを受けとって実行する中間言語仕様でした
Microsoft .NET も仮想マシンという形態を取っていますが、コンパイラは何を吐き出すのでしょうか？

.NET も Java のように、仮想マシン仕様の中間言語を持ちます
これが **MSIL**(Microsoft Intermediate Language) と呼ばれる言語です
アセンブリ言語に類似した言語で、MSIL だけで開発することも可能です

Microsoft は CLS (Common Language Specification) と呼ばれる共通言語仕様を定めていて
CLR で動作するアプリケーション開発用のコンパイラが守るべき規則を定義しています
あなたが、CLS 互換のコンパイラを作成し MSIL にコンパイルする独自の環境を作ることもできるのです

逆に言えば、C# は .NET アプリケーション開発の一つの手段であり
.NET アプリケーションは MSIL を生成できる言語ならばなんでも良いとも言えます
これは Java 言語にはない魅力で (Java アプリケーションは Java 言語からしか生成できない)
.NET アプリケーションは C# の他に VB .NET (Visual Basic を CLS に対応させた言語) や
C++, J# (Microsoft 独自の Java .NET 対応版)、JScript など、多くの言語を選択できます

ところで、Microsoft の C# コンパイラはバイトコードのようなものは出力しません
実際にやってみると **EXE 実行ファイル** を生成します
.EXE 実行ファイルはネイティブな機械語であり、中間言語ではありません
なぜ、ネイティブな実行ファイルが生成されるのでしょうか？

Java はコンパイルするとバイトコードを出力し
実行するにはコンピュータにインストールされている VM で実行する必要があります
ところが Microsoft の開発環境は .EXE ファイルの内部にリソースとしてこれを書き込むのです

ネイティブプラットフォームは通常 MSIL を解釈することはできないので
実行ファイルを実行すると、CLR が **JIT コンパイラ** で MSIL をコンパイルします
JIT は中間言語をネイティブな機械語に翻訳し、これをメモリにキャッシュします
これは大きな利点があり、インタプリタのように実行する度に変換するのではなく
実行前に、一度の処理で中間言語がコンパイルされるということ意味します

開発環境

C# と .NET という技術の概要はなんとなく理解できたと思います
.NET 開発環境は CLR が理解できる MSIL を出力し
CLR が実行時にこれをコンパイルすることで OS やプラットフォームを抽象化します

この新技術は現在 (2001/10/22) まだ開発中であり、完成していません
しかし、すでに Microsoft はこの開発環境をβ版として配付しています
この開発環境を使用して C# 言語を学習することができるようになりました

開発環境には、Microsoft .NET Framework SDK を推奨します
これは、コンパイラなどの開発環境やドキュメントが付属されており
コマンドラインから C# ソースコードを実行ファイルにコンパイルすることができます

Microsoft .NET Framework SDK は Microsoft の Web サイトや雑誌に付属している CD から手に入るでしょう
現在、Windows 2000 を推奨しています。9x や Me でもインストールできますが
動作が不安定になる可能性があります (筆者は Windows 98SE で開発)

Microsoft 対 Sun、C# 対 Java

C# 及び Microsoft .NET の話題に触れるなら、Java にも触れる必要があります
あらかじめ意思表示をするならば、筆者は Java と Sun Microsystems を強く支持している

一時期、実用できるオブジェクト指向言語といえば C++ と Java だった
C++ は高速だが、生産性が低く不完全なオブジェクト指向だとか
Java は C++ に比べ動作が遅くシステムの完成度も低く実用性がないとか
C++ 対 Java というような図式がそれぞれのプログラマの中にありました

しかし、時代は変わり**C# 対 Java**という新たな戦争が始まりました
二つの言語には企業が言語仕様と開発環境を制定したという共通の特徴があります
(Java は Sun Microsystems が95年に発表したプログラミング言語)

Microsoft は、これまで幾度となく Java を倒すための政策を行いました
まず有名な事件が **Microsoft Visual J++** という開発環境です
これは、開発環境を名乗ったアンチ Java 兵器とすら言ってもよい代物です

Java は OS や CPU に依存しない、「一度書いたらどこでも動く」が最大の売りです
そこで、Microsoft は独自の仕様を含めた Java を名乗った開発環境を売ること
Java の統一性を破壊し、Java の最大の魅力を破壊しようと試みたのです
結果、Sun Microsystems に起訴され、Microsoft は敗訴し独自の拡張を認められませんでした

次に、**Active X**と呼ばれるWeb技術も対Javaの技術といって良いでしょう
これは、Java よりも高速にコンテンツやサーバー運用ができる技術とされましたが
Java に比べセキュリティが貧弱で Microsoft の一人よがりな技術で終わることになりました
結局、Active X は Java を打ち破る技術にはならなかったのです

こうして、幾度にわたる Microsoft の Java 侵略は失敗に終わり敗退しているのです
逆に Java は Microsoft 以外の多くの企業の協力の元に進化しつづけています
ルックアンドフィールを実現した Java2 の前に、ついに Windows OS すら危機にさらされているのです
(Java2 はすでに、OS として独立できるだけの機能と思想、システム、ユーザーを持っている)

こうして Microsoft は(おそらく)Javaの2番煎じと呼ばれる覚悟で .NET 構想を提唱しました
事実、CLR は Java VM、MSIL は Java バイトコード、C# は Java という技術に酷似しています

[戻る](#)

[次のページへ](#)