

課題 2

～ C 言語と Java 言語の違いについて ～

Report

提出日： 2009年2月15日
所属： 情報工学部一年次
学籍番号： 095745D
氏名： 西島本 純

1.コンパイラ言語とインタプリタ言語の違い

C言語で書いたコードは、コンパイルされた後、マシンの上で直接実行される。つまりコンパイラ言語である。機械語の効率的な実行のために設計される。これに対してJavaはインタプリタ言語（ソースコードを逐次解釈しながら実行するソフトウェア）になる。Javaで書いたコードは、コンパイル時には中間言語コードへとなおされ、実行時にはバーチャルマシンの上で実行される。

ここでC言語のようなコンパイラ言語は、コードを直接マシンの上で実行するので、実行させるマシンの環境に依存することになる。例えばC言語では、符号なしのint型の大きさを最低16bit(0~65535)と定義している。ここで「最低」というのがポイントで、マシンによっては24bitでも、32bitでも良いということになる。これは、int型だけに限った話でなく、C言語のさまざまな部分に影響している。

一方Javaではバーチャルマシンを利用しているので、プログラムはなるべく近い環境で実行されることになり、こういった問題はかなり少なくなる。

2.オブジェクト指向か

Javaはオブジェクト指向、C言語はオブジェクト指向ではない。オブジェクト指向とは、オブジェクト同士の相互作用としてシステムの振る舞いをとらえる考え方である。

この違いが大きく表れているのは、関数の定義場所であろう。Javaはオブジェクト(クラス)の内部に、メソッドとして関数を定義する。ちなみに、ここではオブジェクトの持つ関数をメソッドと呼んでいる。一方、C言語はオブジェクト指向ではありませんので、グローバルな領域に関数を定義する。これによって、関数へのアクセス方法が異なる。オブジェクト指向は、関数(メソッド)は何らかのオブジェクトに含まれているので、オブジェクト名.メソッド名(引数);

としてアクセスする。

これに対してC言語では、

関数名(引数);

として関数にアクセスする。また、C言語では、クラスからインスタンスを生成する仕組みがないので、newやinstanceof演算子がない。しかし、C言語の構造体の仕組みは、少しくラスと似ている。構造体はメソッドを持たないのが、クラスと大きく違う点である。

3.ポインタの仕組みがあるか

ポインタの仕組みといったが、要するにメモリを直接操作できるかどうかです。C言語ではメモリを直接操作できますが、Javaではできません。そのためメモリサイズを調べるsizeof演算子、メモリを確保するmalloc、ポインタに関する演算子、*や&がJavaにはありません。

4.メモリの利用

C言語とJavaの違いの一つでガベージコレクションがあるかどうかがある。ガベージコレクションとは、プログラムが動的に確保したメモリ領域のうち、不要になった領域を自

動的に解放する機能である。つまりメモリを再利用するための仕組みです。Javaにはこれが実装されており、メモリの領域を考えなければならないC言語より優れているといえる。しかし、ガベージコレクションがあっても、たくさんのインスタンスを確保したためにメモリが本当に不足したら、メモリ不足のエラーになります。

	C言語	Java
オブジェクトと指向	×	○
多重継承	×	なし (interface)
例外	setjmp, longjmpで実装	あり
friend関数	×	なし
ガベージコレクタ	自分で書く	あり
ポインタ	あり	なし (すべてクラスに統合)
演算子オーバーライド	×	× (Stringクラスのみ+を再定義)
マルチスレッド	ライブラリを使う	言語仕様に含む
デストラクタ	×	GCが一元t系に扱う
テンプレート (汎用型)	×	なし
メソッド定義	×	上書きされないものをfinalで指定
変数サイズ	実装依存	規格で決めている
プリプロセッサ	cpp	なし
構造体・共用体	あり	なし (クラスに統合)
無所属変数・関数	これだけ	不可
パッケージ	×	あり
ライブラリ	豊富	豊富