第1章　Java

1.1　Java とは

1995 年、Sun Microsystems 社から公表された、比較的新しい言語である。文法は、C あるいは C++ と似ているが、＿＿＿＿＿＿＿＿。C++と同様、＿＿＿＿＿＿＿＿＿＿言語であるが、C++に比べて＿＿＿＿＿＿＿仕様になっている。なお、JavaScript (ECMAScript) とは文法は似ている（JavaScript が Java に文法を似せている）が、それ以外の関係はなく全く別の言語であるので注意する必要がある。

1.2　Java の特徴

Java は、誕生当時は Web ページにアニメーションと ＿＿＿＿＿＿＿＿＿＿＿＿をもたらすための仕組みとして、世に広まった

WWW の基本的な応答

アプレットを使った場合の応答

HTML だけを用いて書かれた Web 文書の場合は、ユーザがマウスをクリックするなどのアクションがあると、ブラウザはそのアクションを遠隔地の WWW サーバに伝え、その応答を待って新しい表示をする必要がある。

Java を使っている場合は、＿＿＿＿＿＿（Applet）と呼ばれる Java のプログラムをサーバからブラウザへダウンロードする。するとユーザのアクションに対して、ブラウザの中で実行されているアプレットが即時に反応することができる。このように、インタラクティブ性の高いページを記述することが可能になる。

Java の情報のページ

`http://java.sun.com/`	（Java の故郷）
`http://sdc.sun.co.jp/java/`	（日本語の Java 関連情報）
`http://javanews.jp/`	（日本語 Java News）

このように、アプレットと呼ばれる Java のプログラムはネットワークを通じて別のコンピュータに移動して実行されることになる。このような使い方をするためには _____ と _____ という特徴が重要になる。

安全性　これは、簡単にいえばアプレットを使って他人のコンピュータに悪戯をすることができない、ということである。もし、ホームページに任意のプログラムを埋め込んでブラウザ上で実行させることができれば、ハードディスク中のデータを完全に消去してしまうなどのイタズラが簡単に行なえる。

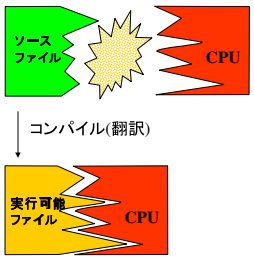
安全性を保障するためには、まずプログラムにファイル操作などをさせない、などの制限を課する必要があるが、C のような言語では、ポインタ（アドレス）操作や無制限な型変換などの仕組みを通じて、いくらでも抜け道を作ることができる。Java はこのような抜け道がないよう設計されている。

> 一方で、アプレットでファイル操作がまったくできないというのでは困る場合もある。そこで作成者が明確なアプレットにファイル操作などを許す**署名つき（signed）アプレット**という仕組みも用意されている。

機種非依存性　Web ページに埋め込まれるということは、さまざまな機種のコンピュータで実行される可能性があるということである。つまり、Java のアプレットに機種依存性があってはいけない。_____ を用いる実行方式ではプログラムが機械語に翻訳されるため、機種依存性は避けられない。一方、_____ を用いる方式では、各機種毎にインタプリタを実装するだけで良いが、効率が犠牲になる。このため、Java では _____ という方法をとる。

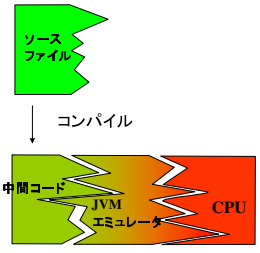
Java のプログラムは Java コンパイラによって _____ という仮想 CPU のコードに翻訳される。この仮想コードを各 CPU 上の JVM エミュレータ（一種のインタプリタ）が解釈・実行する。

コンパイラ

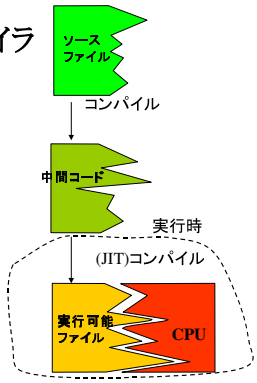


インタプリタ

中間コード
方式



JITコンパイラ
方式



この方法は Java プログラムを直接インタプリタで解釈・実行するよりは高速である。しかし、現在では JVM コードをより高速に実行するために＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿というものを用いて、JVM コードを実行しながら各 CPU の機械語へ翻訳する、という方法を用いる。

また、機種非依存のグラフィックスライブラリやネットワークに関する標準ライブラリを持つことも、今までのプログラミング言語にはなかった重要な特徴である。

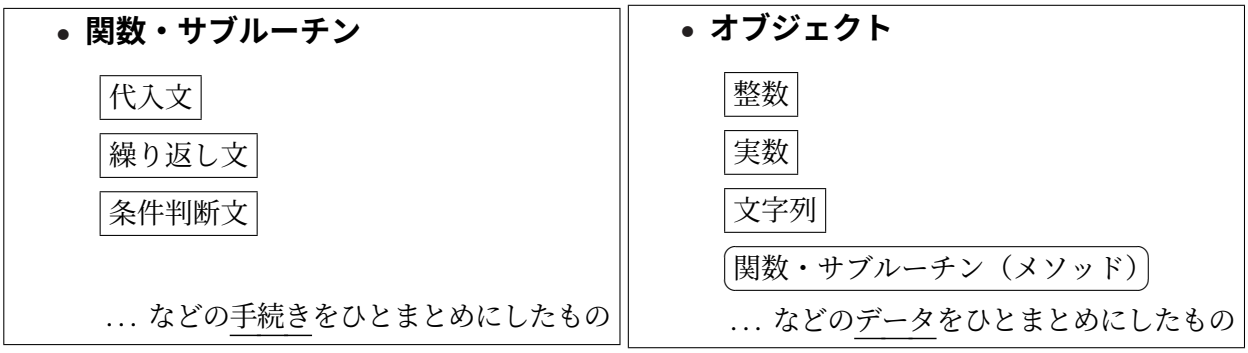
このように当初、Java はアプレットを作成するための言語として広まった。現在では、インタラクティブな Web ページを作成するためのブラウザ側の仕組みとしては、Macromedia Flash などが主流となって、Java アプレットは比較的マイナーな存在になっている。一方で Java の上記のような性質は、他の分野のアプリケーションでも役に立つため、現在はむしろアプレット以外のアプリケーション（例えば WWW **サーバ側**で動作するサーブレット（Servlet）などのプログラム）を作成するために、広く用いられるようになってきている。しかし、オブジェクト指向など Java のさまざまな特徴を理解するためには、現在でもアプレットは良い教材である。

1.3 オブジェクト指向プログラミング

Java はオブジェクト指向型プログラミング（Object-Oriented Programming, OOP）言語である。＿＿＿＿＿＿言語・＿＿＿＿＿言語・＿＿＿＿＿言語・オブジェクト指向型言語などと、プログラミング言語を分類することがあるが、このような言語の分類は、主にプログラミング言語が備える部品化の仕組みに基づいている。

オブジェクト指向型言語に限らず、プログラムの部品を設計することは、単に利用することよりも格段に難しい。まずは、自分で独自のプログラム部品を設計するよりも、オブジェクト指向という仕組みのおかげで豊富に用意された Java の部品群を利用することを学ぶことが必要であろう。この節では、オブジェクト指向型言語が用意する部品を利用するために必要な術語を紹介する。

オブジェクト指向（object-oriented）とは簡単に言えば、従来の手続きを中心としたプログラム部品（＿＿＿＿＿＿＿＿、＿＿＿＿）の利用に加えて、データを中心とした部品（＿＿＿＿＿＿＿＿）の利用を支援することである。関数（サブルーチン）はいくつかの手続きをまとめて一つの部品としたものだが、オブジェクトは、いくつかのデータ（関数 — ＿＿＿＿＿＿ （method）と呼ばれる—も含む）をまとめて一つの部品としたものである。



実際には、プログラム部品として提供されるのは、オブジェクトそのものではなく、オブジェクトの雛型とでもいうべき＿＿＿＿＿ （class）である。クラスは、そこから生成されるオブジェクトが（具体的なデータ （つまり、1 とか 3.14）ではなく）どのような名前と型の構成要素を持つか、のみを指定したものである。クラスを具体化 （instantiate — つまり、`x` という名前の `int` 型の構成要素は 1

で、y という名前の `float` 型の要素は、3.14 などと定めること）したものがオブジェクトである。この時、このオブジェクトはもとのクラスの _____（instance, 具体例）である、という。オブジェクトを構成している個々の構成要素を _____（instance variable）、あるいは _____（member）、_____（field）という。ただし、関数型の要素は特別に _____（method）と呼ぶのが普通である。オブジェクトのメソッドを起動することを、擬人的にオブジェクトに _____（message）を送る、と表現することがある。

> 正確に言えば、メソッドついてはインスタンスごとに定義するのではなく、クラスに対してコードを定義する（ようになっているオブジェクト指向言語が多い）。オブジェクトは各フィールドのデータの他に、どのクラスに属しているか、という情報を持っていて、それによって適切なメソッドのコードが起動される。しかし、メソッドから参照される変数の指している内容はインスタンスごとに異なるので、オブジェクトごとに別々のメソッドを持っていると理解してもはずれているわけではない。なお、Java では匿名内部クラスという仕組みを用いれば、インスタンスごとにメソッドに別々のコードを定義する（のに相当する）ことができる。

従来の手続き型言語では、部品の再利用方法は、既存の部品を関数・サブルーチンとして呼び出すだけだったが、オブジェクト指向型言語では、それに加えて既存の部品（つまりクラス）を少しだけ書き換える（_____, inherit）、という形の再利用の方法が可能になる。これによって、手続き型言語ではプログラムの “幹” の部分を変えて “枝” の部分だけを再利用することができたが、オブジェクト指向型言語では、“枝” の部分を変えて “幹” の部分を再利用することもできるのである。

手続き型言語のイメージ	オブジェクト指向言語のイメージ

最近のソフトウェアではユーザーインタフェースの部分（“枝” の部分）が重要であることが多いので、オブジェクト指向という考え方が特に必要となってきている。

キーワード:

Java, C, C++, オブジェクト指向、アプレット、中間言語方式、JIT コンパイラ、サーブレット、オブジェクト、クラス、インスタンス、インスタンス変数（フィールド）、メソッド、継承