第0章　はじめに

これから、Java 言語によるプログラミングを学習します。と言っても、プログラミングを始めて経験する人、そして「そもそもプログラミング言語って何？」と戸惑っている人もいることと思います。そこで、具体的なプログラミング学習に入る前に、その前提となっている基礎知識を簡単に整理あるいは確認しておきましょう。

0-1　プログラミング言語とは？

プログラミング言語とは、コンピュータに命令する処理内容を記述するために開発された**人工言語**です。世の中には、科学技術計算に用いる FORTRAN（フォートラン）やシステム開発によく用いられる C++(シー･プラス･プラス)そして初心者向けと言われる BASIC（ベーシック）など、様々なプログラミング言語があります。

それでは、なぜプログラミング言語が必要なのでしょうか？端的に言えば、それは、人間がプログラムを書きやすくする、あるいは読みやすくするためです。実はコンピュータが理解できるのは、0 と 1 からなる２進数の並びだけです。コンピュータの開発当初のプログラムは

101010 0000 00000001101110
101000 0000 00000001110111
・・・

などという、２進数のコードが延々と続くものでした。これを、機械（コンピュータ）が直接理解できる言語という意味で**機械語**と呼びます。言うまでもなく、人間にとってこのような２進数の並びを理解することは困難です。機械語は機械（コンピュータ）に優しく、人間に厳しい言語と言えます。そこで、人間にとって分かりやすい言語が開発されました。上の機械語の命令を、例えば FORTRAN という言語で記述すると、

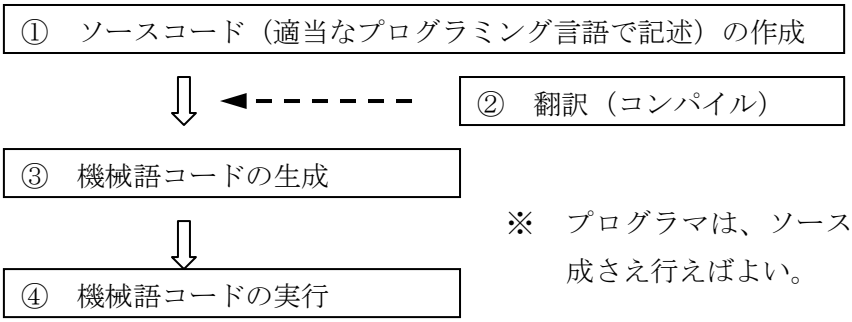
A=3
B＝A+4

などとなります（正確な対応ではありませんが、ここでは気にしないで下さい)。これを見ると、正確には分からなくても、B というモノ（変数）には７が入るのだろう・・・という予測はつくでしょう!?このように、(機械にではなく)”人間に優しく”、つまり分かりやすくするために開発された言語がプログラミング言語なのです。言語である以上、その記述方法には約束事があります。それを**文法**と言います。基本的な文法が分かれば、様々な処

理内容をコンピュータに伝えることができます。

ところで、コンピュータが 0 と 1 との並びしか理解できないという事情は、実は今も（昔と）全く変わっていません。コンピュータは相変わらず機械語しか理解できないのです。そこで、適当なプログラミング言語で記述したプログラム（**ソースコード**と言います）を機械語に変換する過程が必要になります。この過程を翻訳（**コンパイル**）と言います。これは、各プログラミング言語用に開発されている翻訳システム（**コンパイラ**と言います）によって行われるので、人間（プログラム作成者）は気にする必要はありません。つまり、プログラマは少なくともプログラミング言語の文法を理解すれば、コンピュータに処理を実行させることができるようになっているのです。

＜プログラムの作成・実行の流れ＞



※ プログラマは、ソースコードの作成さえ行えばよい。

本テキストでは、Java 言語というプログラミング言語を採り上げ、その基本的な文法を学習します。もっとも、言語の文法だけを知っていてもそれを活用することはできません（文法だけでは話せない、という事は恐らく英語で経験済みでしょう!?）。やはり、実際の問題に適用した適用例（のプログラム）を通じて実践的に理解を積み重ねて行く事が必要です。次章以降に出てくる課題プログラムがその適用例に当たります。どうか、着実に課題プログラムをマスターして行って欲しいと思います。

0-2 Java 言語とは？

これから学習する Java 言語は、1995 年に Sun Microsystems（サン・マイクロシステムズ）社[1)]が開発したプログラミング言語です。まだ誕生から 16 年程度という新しい言語です。上で挙げた FORTRAN が開発されたのが 1957 年であったことを思うといかに新しい言語であるかが分かるでしょう。にもかかわらず、多くの人は、雑誌や広告などで”Java 言語”というキーワードを目にしたことがあるのではないでしょうか？これは、Java 言語がそれだけ注目されていることを意味します。また、2003 年のある新聞記事に「Java 言語に習熟した技術者が不足－IT 業界の需要に追いつかず」という内容の記事がありました。これをみると、どうやら、ただ注目されているだけではなく、社会的な需要も大きいようです（この傾向は現在でも続いています)。

その理由は幾つかあります。しかし、現段階ではその詳細を説明してもまだピンと来ないと思いますので、ここでは、Java 言語でどのようなことができるのか、ということを述べることでその答えに替えましょう。Java 言語を用いれば以下のようなプログラムを作ることができます。

① Windows 上で動作するプログラム

ワープロや表計算ソフトなど、開いたウィンドウ上で動作するアプリケーションソフトウェアのことです。

② Web ブラウザ上で動作するプログラムーアプレット

Web ブラウザ上で動作するプログラムのことを**アプレット**(Applet)と呼びます。Java 言語が現れた当初は最も注目された機能です。

③ Web サーバ上で動作するプログラムーサーブレット、JSP

Web 上で履修登録を行うシステムなどのような、Web アプリケーションのことです。これを作成するためには Web サーバ上で動作するプログラムを作成する必要がありますが、そのようなプログラムを**サーブレット**(Servlet)と呼びます。また、HTML 言語の中に Java 言語の処理を記述できる **JSP**(JavaServer Pages)というプログラムもあります。これらの詳細については本テキストでは触れません。

④ データベースと連携するプログラムーJDBC

ビジネスの現場では、顧客情報データベースなど、何らかのデータベースを扱う処理が必須となります。JDBC(Java DataBase Connectivity)とは Java プログラムで色々なデータベースを操作するための命令群のことで、これらを用いることによって、データベースを操作するシステムを Java 言語によって作成することが容易になります。

---

1) 2010 年 1 月に Sun は米国 Oracle 社に買収されました。したがって、Java も現在は Oracle 社に属しています。

⑤　携帯端末上で動作するプログラム

携帯電話や、PDA 端末などの携帯機器上で動作するプログラムです。皆が持っている携帯電話上でも恐らく Java プログラムが動作しているはずです。最近はスマートフォン用のアプリケーション開発にも用いられています。

何だか難しそうな用語が並びますが、今はその詳細を気にする必要はありません。まずは、「Java 言語で色々なことができるのだ・・・。」というイメージを持ってくれれば十分です。この、色々なことが（比較的容易に）できる、というあたりに Java 言語の人気の一因があると思われます。

本テキストでは、主に、①の Windows 上で動作するプログラム（Windows アプリケーション）の作成を採り上げます。③以降は少し専門的な内容になるので、本テキストでは扱いませんが、これらについては様々なテキストが市販されています。本テキスト学習後、興味のある人は学習してみて下さい。本学部で学習するデータベースやネットワークプログラミング論（③の Web サーバ上で動作するプログラムを学習します）などの講義と並行する形で理解を深めて行けば、在学中に Java 技術者のエキスパートになれると思いますよ・・・。

さて、Java 言語には３つのエディション（Edition）という分類があります。Java 言語の解説書に良く出てくる言葉なので、それについて次ページのコラムにまとめましたので目を通しておいて下さい。本テキストでは、このコラムの中の **J2SE** を用いて Java プログラミングの学習を行います。これらエディションのバラエティをみても、Java 言語の応用範囲の広さを想像することが出来でしょう。色々と語り出すと、Java 言語には他にも特徴的な点があるのですが、最後に次節で Java 言語の翻訳（コンパイル）の特徴についてふれておきましょう。

コラム　J2SE、J2EE そして J2ME とは？

Java 言語プログラミングに必要なシステム一式はJDK(Java Development Kit)と言う名前で開発元のサン・マイクロシステムズ社より公開・配布されています。そして 1995 年春の JDK1.0 からバージョンアップを重ねてきました。特に 1998 年には JDK1.2 がリリースされ、このとき大幅な変更がなされたので、これ以降を **Java2** と呼ぶようになりました。さらに、1999 年には、Web サーバシステムの構築や、携帯端末向けのアプリケーションの開発など、多様化する用途に合わせて次の３つの Edition（エディション）に分類して配布されるようになりました。

Ⅰ．J2SE(Java2 Platform Standard Edition)

従来の Java2 を引き継いだもので、主にパソコン上で動作させるプログラムを作成するために必要な機能が備わっています。このエディション内の機能を使って上で述べた①、②そして④を作成することができます。

Ⅱ．J2EE(Java2 Platform Enterprise Edition)

J2SE の他にサーバ上で動作するプログラムを作成する機能を加えたものです。このエディション内の機能を使って、上の③のプログラム、いわゆる**サーバサイドシステム**を構築することができます。さらに、Enterprise（企業）版という名前から推測できるように、企業向けの高度なアプリケーションやシステムを構築するための機能も含まれています。

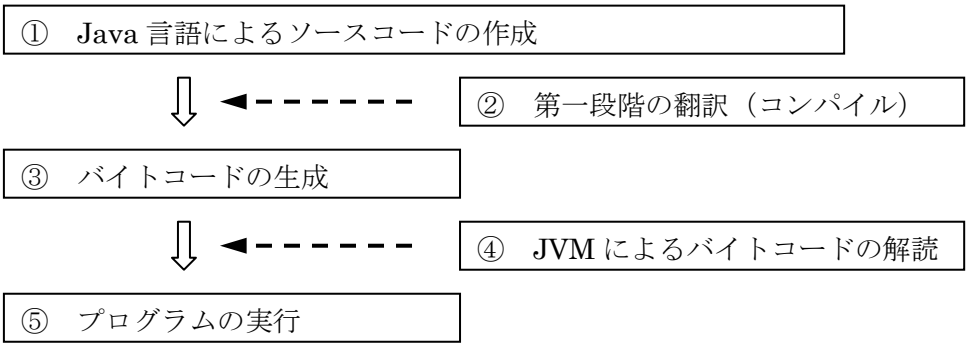
Ⅲ．J2ME(Java2 Platform Micro Edition)

携帯電話や PDA 端末など、小型携帯機器用のプログラムを作成するためのエディションです。Java２から余分な機能を取り除く反面、限られた CPU やメモリに対応するための機能が用意されています。携帯電話の普及に伴い、注目を集めるようになったエディションです。

0-3 Java 言語の翻訳(コンパイル)について

0-1 節で説明したように、Java 言語の場合もソースコードを実行するには、翻訳を行います。しかし、実はいきなり機械語に翻訳する訳ではありません。次に示すように、一旦、**バイトコード**と呼ばれる中間言語に翻訳して、実行時に、Java 仮想マシン(**JVM**:Java Virtual Machine)と呼ばれる(2段階目の)翻訳システムがバイトコードを解読し、プログラムをする、という 2 段階方式を採っています。

＜Java プログラムの作成・実行の流れ＞



一見、無駄なことをしているようですが、実はこういったことをする背景には、Java 開発チームの戦略があるのです。順を追って説明しましょう。まず、機械語は、Windows や Linux など、使用している OS によって異なります。国が異なれば言語が異なるように、土台となる OS が異なれば、(それが理解できる)機械語も異なるのです。ところで、コンピュータの世界では OS やハードウェア環境のことを**プラットフォーム**(Platform)と言います。この言葉を用いると、「機械語に翻訳してしまうと、特定のプラットフォームでしか動作しない」ということになるのです。Java 開発チームはこれを嫌いました。そこで、一旦、共通語に相当するバイトコードという中間言語を作り、翻訳をそこで止めたのです。そして一方で、Windows、MacOS そして Linux など様々なプラットフォーム毎に、バイトコードを解読して実行できる Java 仮想マシン(JVM)というシステムを用意しました。つまり、JVM がインストールされているマシンであれば、どのプラットフォームでも動作する環境を、この 2 段階の翻訳システムで実現したのです。Java 開発チームはこのことを「Write once, run anywhere(一旦ソースコードを書けば、どこでも動作する)」というキャッチフレーズで宣伝しました。そしてこれは JAVA 言語の特徴の一つになっています。

実はこの性質は、コンピュータネットワークの世界では極めて重要です。なぜならネットワークにつながっているのは、Windows マシンや Linux マシンなど様々であるため、特定のプラットフォームに依存するプログラムでは利用範囲が限定されてしまうからです。Java が普及した大きな理由には、この「どのプラットフォームでもプログラムを実行できる」という性質があります。