わかりすぎる C言語の教科書

0.1
0.2
0.3
0.4
0.5
0.6
0.7
1.0
1.5
2.0

C言語習得のポイントが見えてくる！

中島省吾 著

SCC

わかりすぎる

C言語の教科書

中島省吾 著

SCC

・本書に記載されたURL等は執筆時点でのものであり、予告なく変更される場合があります。
・本書の使用（本書のとおりに操作を行う場合を含む）により、万一損害が発生しても、出版社、著者（著作権者）は一切の責任を負いかねますので、あらかじめご了承下さい。

はじめに

本書は、プログラミング初学者に向けたC言語の入門書です。

プログラミング言語といえば、現在は「Java」を筆頭とする「オブジェクト指向言語」が、開発現場の主流を占めています。

対してC言語は、世に出て既に40年以上が経過し、当然の如くオブジェクト指向にも対応していません。このような言語が、現在でもなお高い人気を博しているその理由は、「言語仕様がシンプルであること」や、「高級言語でありながらハードを直接操作できる低レベルな記述が可能であること」といった特徴以外に、「C++」「C#」「Objective-C」といった代表的なオブジェクト指向言語が、どれもC言語の派生言語であるという点が大きいのではないかと思われます。つまり、これら最新言語を学ぶ前に、そのルーツとなったC言語を学ぶことで、まずはプログラミングの基本を理解しようという訳です。その意味において、これからプログラミングを学ぶ初学者にとって、C言語はうってつけの言語といえるのではないでしょうか。

では、そのC言語を学ぼうとしたとき、従来の入門書では「テキストエディター」と「コンパイラ（入力したプログラムを実行できる形に変換するプログラム）」を利用するというのが半ば常識と考えられてきました。しかし実際の開発現場では、この2つのツールのみで開発を行うことは非常に希です。では、どのような環境で開発を行っているのかというと、それは「統合開発環境（とうごうかいはつかんきょう）」と呼ばれるアプリケーションです。統合開発環境とは、コンパイラだけでなく、入力ミスや入力候補を表示する高機能なエディターを搭載し、バグを修正したりプログラミングの支援をするさまざまなツールが統合された、開発専用のアプリケーションのことです。

このような開発環境の中には、もちろん驚くほど高価なものもありますが、無償で利用できるソフトも数多く存在します。そこで本書では、できるだけ実践に即した環境で学習することを目的として、「Eclipse（エクリプス）」というフリーの統合開発環境を利用します。Eclipseは、多くの開発プロジェクトが実際に採用している本格的な統合開発環境で、「ｇｃｃ（ジーシーシー）」というフリーのコンパイラと組み合わせることでC言語の開発にも対応します。しかも、これらのツールは既に統合された状態で用意されており、インターネット経由でダウンロードすることが可能になっています。

さらに本書の特徴として、C言語の文法を解説する際には、図やシンプルなプログラムを多

用するようにしています。こうすることで、イメージと実際の動作が直接結び付き、理解度がより高くなります。また、プログラミングの考え方を身に付けるため「アルゴリズム」の解説や、そのアルゴリズムをＣ言語でどのように実装するのか、その実装方法も紹介しています。ぜひ本書を活用して、Ｃ言語とプログラミングの基本をマスターしてください。

本書が、皆さんのＣ言語の学習に少しでもお役に立てれば幸いです。

著　者

目次

第１章

C言語を学習するための準備

この章では、プログラミングやC 言語の概要と、開発環境のインストール方法を紹介してC 言語を学習するための準備をします。

1.1 プログラミングとは

C 言語の学習を始める前に、そもそも「プログラミング」とは何をすることなのかを説明します。

1.1.1 プログラミングとは

プログラミングを言葉にすると、次のようになります。

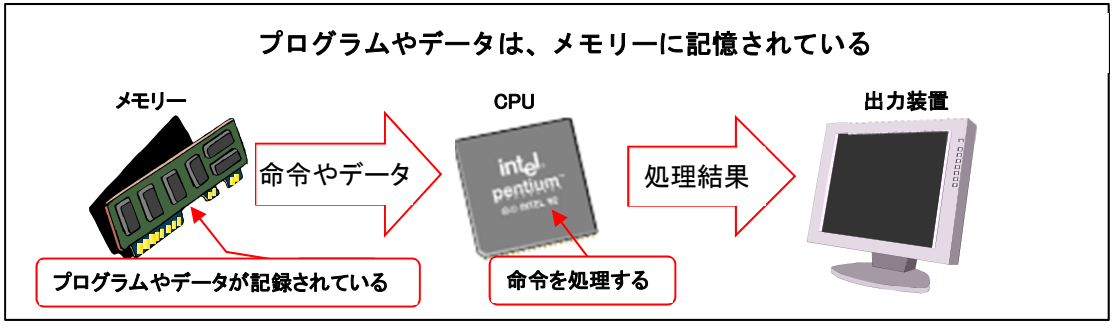
> 目的の処理をコンピュータに実行させるため、必要な命令を組み立てる行為

こうして、組み立てた命令のまとまりを「プログラム」と呼び、プログラムを記述する人を「プログラマー」と呼びます。

プログラムは、コンピュータに内蔵されたＣＰＵ（シーピーユー）（Central Processing Unit）（コントロール　プロセッシング　ユニット）と呼ばれる装置へ送信され、CPU がプログラムの命令を解釈しながら実際の処理を行います。

CPU は、「中央処理装置（ちゅうおうしょりそうち）」や「中央演算処理装置（ちゅうおうえんざんしょりそうち）」と呼ばれるように、与えられた命令に従い演算やデータ転送を行います。そして、その結果は別の処理で利用されたり、「ディスプレイ」などの出力装置へ出力されたりします。

CPU に処理を行わせるプログラムやデータは、その都度人間が入力していくのではなく、あらかじめ「メモリー」と呼ばれる記憶装置に保存しておきます。そして、プログラムを「実行」すると、メモリー内の命令が CPU へ順次転送され処理が進んでいきます。



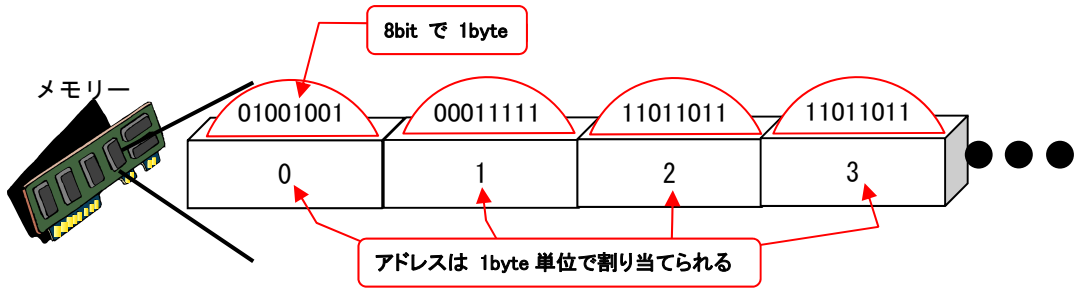
1.1.2 メモリーの構造

メモリーは、電気のあるなしを記憶する素子の集まりでできています。そして、記憶素子に電圧がある状態を 1、ない状態を 0 と考え、0 と 1 で 2 進数を表現します。この、メモリーが記憶できる 1 番小さな単位を「bit（ビット）」と呼びます。

1bit で記憶できる値は、0 と 1 のみですが、これを 8 つ並べて 00000000 ～ 11111111（10 進数で考えると 0 ～ 255 ）まで記憶できるようにした単位を「byte（バイト）」と呼び、メモリーはこの byte 単位でデータを管理します。

さらに、メモリーには byte 単位で「アドレス」という「データの場所を表す番号」が割り当てられていて、プログラムがメモリー内のデータにアクセスするときは、このアドレスを使ってデータが記憶されている場所を特定し、データを読み出したり書き込んだりしています。

メモリーには 1byte 単位でアドレスが割り当てられている



プログラム、ソフトウェア、アプリケーション

「プログラム」という呼び名は、実行する環境や規模によって呼び方が変わる場合があります。例えば、CPU が読み込んで実行するものは、すべて「プログラム」と考えることができますが、コンピュータを「ハードウェア」と「ソフトウェア」に分けて考えた場合、CPU は「ハードウェア」の一種、プログラムは「ソフトウェア」を構成する要素になります。

また、コンピュータで表計算を実現したり、ワードプロセッサとして印刷物を作成できるソフトウェアを、「アプリケーション」と呼ぶ場合があります。

2 進数、8 進数、10 進数、16 進数

通常、私達人間は 10 進数で数字を扱います。しかし、コンピュータは 0 と 1 しか扱えないため、数値を扱う際には 2 進数で考えます。

しかし、2 進数で表現する数値は桁数が多くなるため、人間が行うプログラムでは 10 進数のほかに、8 進数や 16 進数がよく用いられます。

ただし、プログラムの中で登場する数字が何進数で表現されているのか、数字だけでは判断しにくいため、C 言語では次のように数字の頭に印を付けます。

8 進数 数字の頭に 0(数字のゼロ)を付ける。
例:07 012 076

16 進数 数字の頭に 0(数字のゼロ)と小文字の x、もしくは大文字の X を付ける。
例:0x1c 0X3B

また、一般的な C 言語のコンパイラではできませんが、2 進数を以下のように表記できる場合もあります。

2 進数 数字の頭に 0(数字のゼロ)と小文字の b、もしくは大文字の B を付ける。
例:0b00001101 0B1000111100001010

プログラミングで、8 進数や 16 進数がよく利用されるのは、2 進数との相性が良いためです。

例えば 10 進数では、9 の次に桁があがり 10 になります。同じように、8 進数は 07 の次に桁が上がり 010 になります。2 進数の場合 0b111(10 進数の 7)の次に桁が上がると 0b1000(10 進数の 8)になります。つまり、8 進数の 1 桁と 2 進数の 3 桁が同じになります。また、16 進数は 0x0 ～ 0x9、そして 0xa、0xb、0xc、0xd、0xe、0xf(10 進数の 10 ～ 15)の次に桁が上がります。つまり、16 進数の 1 桁と 2 進数の 4 桁が同じになります。

このように、桁上がりの区切りが良いので、8 進数や 16 進数がプログラミングではよく利用されるのです。

1.2 プログラミング言語

プログラミング言語は、C 言語だけではありません。ここでは、C 言語以外のプログラミング言語について説明します。

1.2.1 アセンブリ言語

CPU が、直接理解できる命令のことを「マシン語」とか「機械語（きかいご）」と呼びます。

メモリーから CPU へマシン語を送信する際、マシン語は対応する番号（数値）で転送されます。しかし、プログラマーが、この番号でプログラミングするのはとても大変なので、マシン語に対して 1 対 1 で「アルファベット文字列や記号（これを「ニーモニック」と呼びます）」を対応付けし、プログラムが書けるようにしました。このような言語を「アセンブリ言語」と呼びます。

アセンブリ言語のプログラムは、完成するとマシン語（数値）に変換され（この変換プログラムを「アセンブラ」と呼びます）、メモリーに記憶されます。この「アセンブリ言語」が、プログラミング言語の元祖といえます。

ところが、アセンブリ言語は CPU の命令と直に対応しているため、ほかの CPU で同じプログラムを動かすには、その CPU のアセンブリ言語でプログラムを作り直す必要があります。

このように、ほかの CPU やコンピュータで同じプログラムが動くようにすることを「プログラムを移植する」といいます。

やがて、プログラムが高度に複雑化するにつれ、アセンブリ言語よりも移植のしやすい（移植性が高い）プログラミング言語の開発が求められるようになりました。

ただし、そのような言語を広めるためには、抽象的な（人間が理解しやすい）文で、プログラムが記述できなければいけません。例えば、「A という文字を画面に表示せよ」とか「もし、ボタンがクリックされたらファイルを印刷しろ」のように、誰でもわかりやすい文法で記述できることが求められます。

このような、人間が理解しやすい文法で記述できるプログラミング言語を「高級言語（または、高水準言語）と呼びます。

現在の高級言語は、英語や日本語のような言語に比べれば遙かに単純で簡素な文法ですが、そ

れでもアセンブリ言語の単語と記号だけの言語よりも遙かに人間が理解しやすく、移植性や可読性（プログラムの読みやすさ）が向上しています。

もちろん高級言語でできたプログラムは、人間にとって理解しやすいというだけで、CPU の命令とは異なります。そこで、高級言語でできたプログラムを CPU に理解させるために、マシン語に変換する必要があります。この変換作業を「コンパイル」、変換するプログラムを「コンパイラ」と呼びます。

また、このようにコンパイルを行うプログラミング言語のことを「コンパイル方式のプログラミング言語」と呼びます。

1.2.2 高級言語

現在、高級言語として代表的なものには、次のような種類があります。

現在の主な高級言語

言語の名前	説明
COBOL	COmmon Business Oriented Language の略。主に企業の会計処理など、大型計算機のプログラムに使われる言語。
Fortran	FORmula TRANslater の略。1956 年に IBM 社によって開発された科学技術計算用言語。
PASCAL	Philips’ Automatical Sequence Calcurator の略。1968 年にスイス連邦工科大学の Niklaus Wirth 教授によって開発された教育用言語。
Basic	Beginner’s All-purpose Symbolic Instruction Code の略。主に初心者向けの言語として開発される。当初は、命令を逐次解釈しながら実行する「インタープリタ形式」を採用していたが、後に「コンパイル方式」の Basic も開発されている。
C	1972 年にアメリカ AT&T 社のベル研究所で、OS の開発用に D. M. Ritchie 氏と B. W. Kernighan 氏によって開発された。
C++	1992 年に仕様が策定されたオブジェクト指向言語。C 言語の上位互換なので、C 言語のコンパイラは C++ のコンパイラで代用できる。
objective-C	C 言語と上位互換のオブジェクト指向言語。主にアップルの Mac OS X や iOS 上で動作するソフトの開発に利用される。
Java	1995 年にサン・マイクロシステムズ（現在は、オラクル社に吸収され存在しない）が発表したオブジェクト指向言語。プログラムは、Java バイトコードにコンパイルされ、Java 仮想マシン上で動作するのが特徴。
C#	マイクロソフト社が 2000 年に発表したオブジェクト指向言語。C++、Java のよい点をミックスして開発した。プログラムは中間言語にコンパイルされ、.NET Framework 環境で動作する。

1.3 アルゴリズム

プログラムは、プログラミング言語の文法だけ理解しても、効率の良いプログラムは作れません。ここでは、プログラムの考え方である「アルゴリズム」について学びます。

1.3.1 アルゴリズムとは

アルゴリズムとは、次のようなものです。

プログラミングにおいて「問題を解決するための手順」を表現したもの

また、「問題を解決するための手順」に名前を付け、その総称を「アルゴリズム」と呼ぶ場合もあります。

例えば、バラバラに並んだ数値を、大きい順、または小さい順に並べ替えるプログラムの場合、「どのような手順で値を移動させると並べ替えが完了するか」を、表現したものがアルゴリズムです。

したがって、同じ問題を解決するアルゴリズムは、1 つとは限りません。つまり、同じ結果になるのであれば、複数のアルゴリズムが存在してもかまいません。

どのアルゴリズムを使って、数値を並べ替えるのかは、処理の効率やリソース（メモリーやファイルなど、プログラムにとっての「資源」のことを「リソース」と呼びます）を考慮して、プログラムを行う人（プログラマー）が判断します。

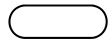
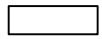
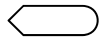
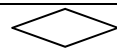
1.3.2 フローチャート

プログラマーが考えたアルゴリズムを、別のプログラマーに説明するとき、通常は文章よりも図や記号を用いて表現した方がわかりやすくなります。

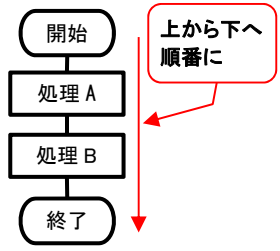
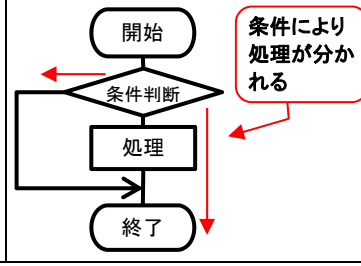
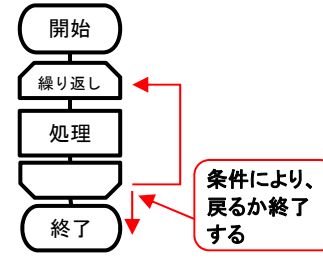
そこで、アルゴリズムを説明する際に処理の流れを「フローチャート」と呼ばれる、標準化された記

号で表現することがあります。

フローチャートとは、プログラムの流れを決められた図形で表現するもので、主に次のような記号を用います。

記号	呼び名	説明
	端子	処理の始まりと終わりを表す
	処理	演算や代入など
	入力	キーボードなどからの入力
	表示	コンソールなどへの出力
	ループの開始	繰り返しの始点
	ループの終了	繰り返しの終点
	判断	条件判断により分岐する
	結合子	別の流れとの接続を表す

また、プログラムの流れ（処理の流れ）には大きく分けて「直線」「分岐」「繰り返し」の３種類があり、次のように表現されます。

直線	分岐	繰り返し
上から下に処理が進む	条件の判断により、処理が分かれる	条件が成り立つまで、もしくは成り立たなくなるまで繰り返す
		

Ｃ言語のプログラミングでは、この３種類の流れを組み合わせることで、処理全体の流れを組み立てます。

1.4 C 言語とは

いよいよ、C 言語の説明を始めます。まずは、C 言語がどのような理由で開発されたのかをお話しましょう。

1.4.1 C 言語の歴史

C 言語は、1972 年にアメリカＡＴ＆Ｔ社のベル研究所で、Ｄ．Ｍ．Ritchie氏とＢ．　Ｗ．Kernighan氏によって開発された言語です。

当時のベル研究所では、UNIXと呼ばれるオペレーティングシステム（以降ＯＳ）の開発をアセンブリ言語で行っていましたが、ほかの CPU へプログラムを移植するのが非常に困難でした。

※OSとは、コンピュータの「基本ソフトウェア」と呼ばれるもので、キーボートからの入力やディスプレイの表示、ファイル操作といったコンピュータの基本機能を、アプリケーションに提供するソフトウェアのことです。

そこで、UNIX をほかのコンピュータへ移植しやすくするために、Ritchie 氏と Kernighan 氏が開発した高級言語が「C 言語」です。

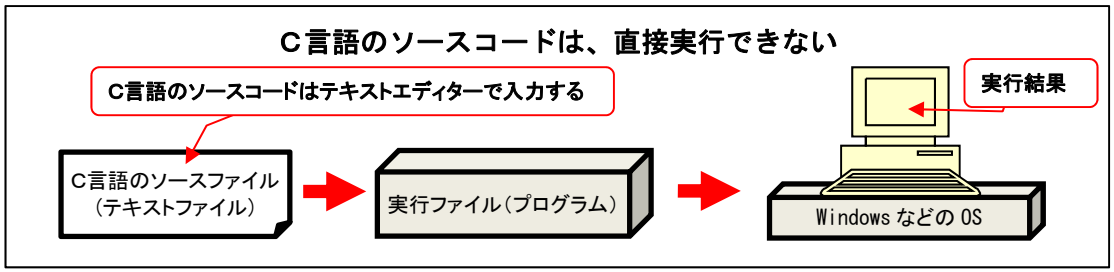
さらに二人は、1978 年に「The　C　Programming　Language」という、C 言語のバイブルともいうべき本を出版します。この書籍により、C 言語を学ぶ技術者が増え、さまざまなコンピュータでＣ言語によるプログラムの開発ができるようになっていきました。

その後、1989 年にはアメリカの標準規格である「ANSI（American　National　Standards　Institute）」でも制定され、現在でも、オブジェクト指向言語である C++言語の基になる言語として、さまざまな用途に利用され続けています。

1.4.2 Ｃ言語でプログラムを作る方法

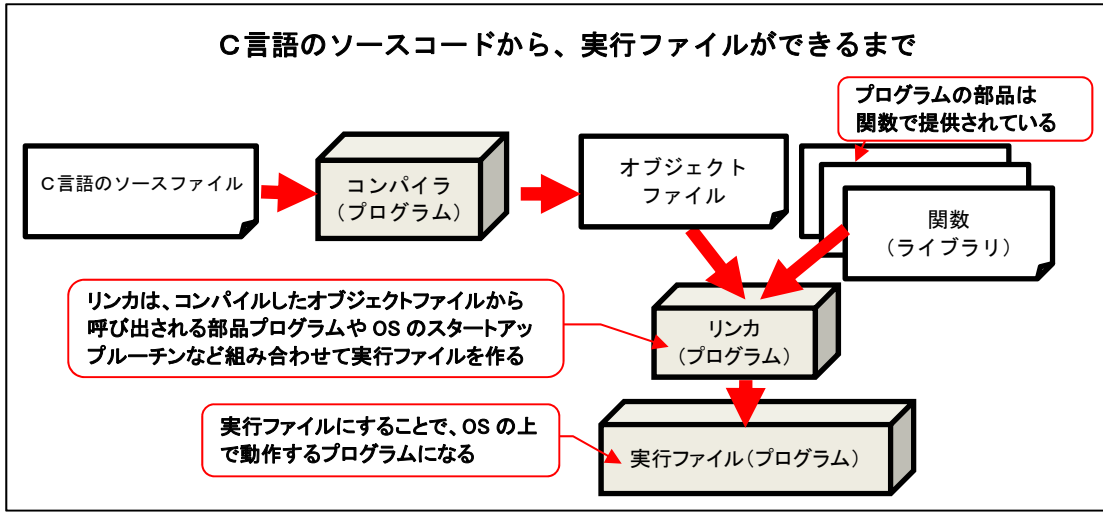
Ｃ言語でプログラムを作るには、まずＣ言語で目的の処理をするように記述したテキストファイル（文字だけでできたファイル）を作成します。このテキストファイルは、通常「テキストエディター（キーボードなどで文字を入力してテキストファイルを作成することができるアプリケーション）」で作成します。そして、このプログラムを記述したテキストのことを「ソースコード」、できたテキストファイルを「ソースファイル」と呼びます。

ソースファイルの中にはＣ言語でプログラムが記述されていますが、直接コンピュータ上で実行できるわけではありません。ソースファイルは、OS などのプラットフォーム上で動作する「実行ファイル」に変換する必要があります。



ソースファイルを実行ファイルにするには、まずソースファイルをマシン語でできたオブジェクトファイルに変換します。この変換作業のことを「コンパイル」、変換するプログラムを「コンパイラ」と呼びます。

さらに、オブジェクトファイルにあらかじめ用意されているプログラムの部品（C 言語では、関数ライブラリとして提供されます）を組み込み、OS から起動できるようにするプログラム「スタートアップルーチン」などを組み込みます。この組み込み作業を「リンク」、組み込み作業を行うプログラムを「リンカ」と呼びます。



1.5 開発環境のインストール

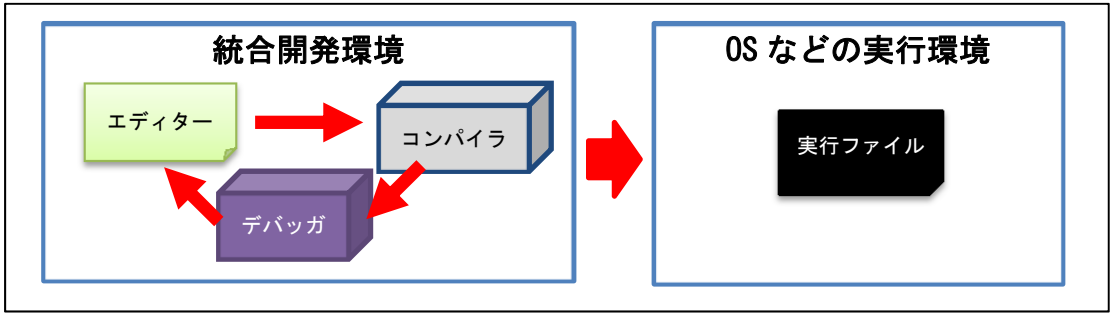
それでは、C 言語でプログラムが作れるように、ツールをインストールしましょう。本書では、C 言語の開発に統合開発環境であるEclipse(エクリプス)を使います。

1.5.1 統合開発環境

C 言語のプログラムは、コンパイラ、リンカ、テキストエディターなど、動作させる環境に合わせたいくつかのツールが必要です。そこで、本書では「統合開発環境(とうごうかいはつかんきょう)」を利用して、C 言語のプログラムを学習することにします。

統合開発環境とは、ソフトウェア開発に必要な、コンパイラ、リンカ、エディター、バグの修正をするときに役立つツール(これを「デバッガ」といいます)など、さまざまなツールをあたかも 1 つのアプリケーションであるかのように統合した環境のことで、ＩＤＥ(アイディーイー) (Integrated インテグレーテッド Development デベロップメント Environment エンバイロメント)と呼ばれます。

これらのツールが同じ環境から利用できることによって、ソースコードの入力、実行、デバッグといった一連の作業を、ストレスなく繰り返すことができ開発のスピードアップが図れます。



C 言語の開発で利用される IDE には、有料のものからフリーのものまで多くの種類がありますが、今回は、オープンソースの代表的な統合開発環境である、「Eclipse」を使用します。

1.5.2 Eclipse をインストールする

Eclipse は、実際の開発現場でも利用されている高機能な IDE で、本来は「Java」というプログラミング言語の開発用に作られたものですが、「C」「C++」「ＰＨＰ（ピーエイチピー）（Web サイトの作成などに用いられる言語）」など多くの言語を扱えます。

このように、Eclipse がさまざまなプログラミング言語に対応するのは、「プラグイン」と呼ばれる Eclipse の部品を組み合わせることで、機能が拡張できるためです。

プラグインの中には、メニューなどの表示を変更するものもあり、さまざまな国の言語で画面表示をすることができます。もちろん、日本語表示用のプラグインもあるので、表示を日本語に変更することも可能です。

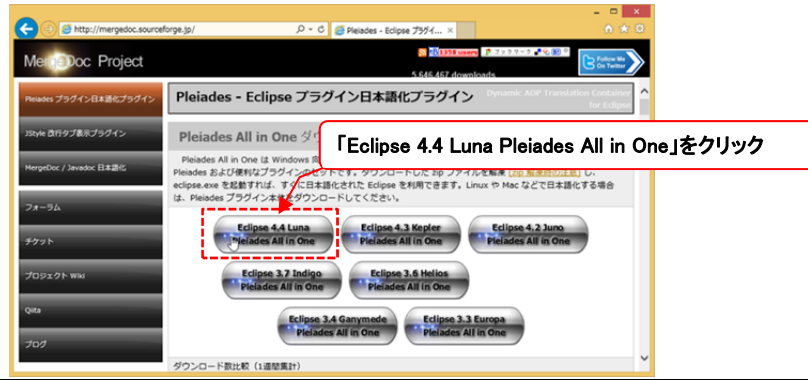
また、インストールしただけでは英語表示であるため、日本語にするためのプラグインも必要です。そこで、本書では Windows 環境に限定されますが、「Pleiades（プレアデス） All（オール） in（イン） One（ワン）」という Eclipse のパッケージを利用します。

Pleiades All in One は、プログラミング言語別にパッケージングした Eclipse のことです。本体と日本語プラグインである「Pleiades」がセットになっていて、zip ファイルをダウンロードして解凍するだけで、日本語化された Eclipse を使うことができます。

それでは、以下のサイトにブラウザでアクセスしましょう。

```
http://mergedoc.sourceforge.jp/
```

Pleiades All in One のダウンロードページ



最新の Eclipse は、「Eclipse 4.4 Luna Pleiades All in One」です。リンクになっているのでクリックしましょう。

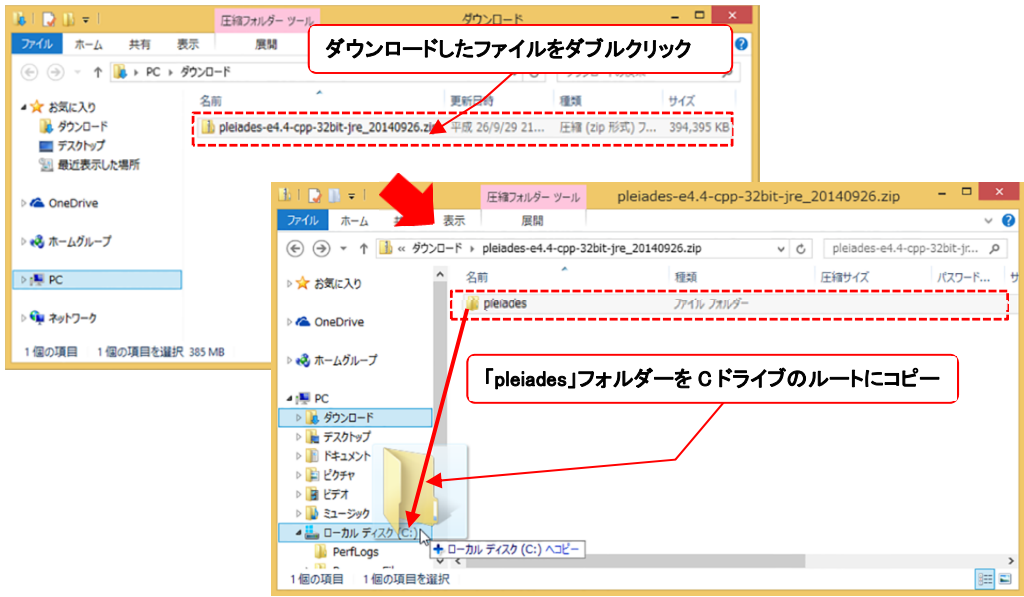
各言語に対応したパッケージの一覧画面になります。

選択する言語は「C/C++」を選択します。「Full Edition」と「Standard Edition」の 2 種類ありますが、本書の画面紹介は Windows 8.1 Update 32bit 版で行っているため、ここでは 32bit 版の Full Edition をクリックします(64bit の Windows をお使いの方は、64bit の方を選択してください)。

「Download」ボタンをクリックすると、ブラウザのダウンロード確認になるので、32bit の Full Edition の場合「pleiades-e4.4-cpp-32bit-jre_20140926.zip」がダウンロードされます。

注意　ダウンロードするファイル名は、バージョンによって多少異なります。図のファイル名は、2014 年 10 月にダウンロードしたものです。

ダウンロードが完了したら、エクスプローラーでダブルクリックして開きます。



開くと、「pleiades」フォルダーがあるので、Ｃドライブのルートにドラッグアンドドロップしてコピーします。解凍しながらコピーするので、５分～20 分ほど時間がかかる場合があります。完了までしばらく待ちましょう。

※コピーが終了したら、ダウンロードした「pleiades-e4.4-cpp-32bit-jre_20140926.zip」は削除してかまいません。

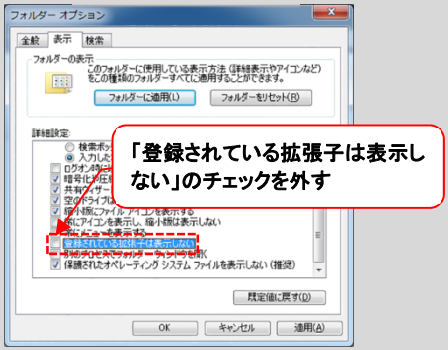
Windows で拡張子を表示する

Windows は、デフォルトでファイル拡張子が非表示になっています。「拡張子」とは、ファイル名の最後に「.(ドット)」と共に追加するファイルの種類を表す文字列のことです。

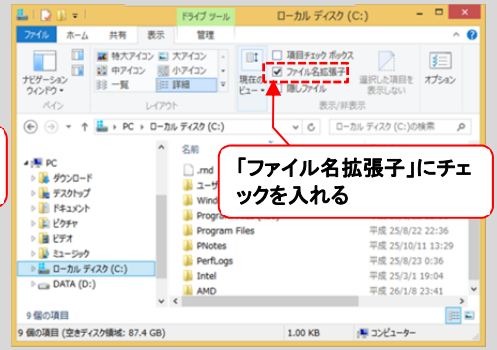
この「拡張子」を表示するには、Windows 7 の場合、エクスプローラーの[整理]メニューにある「フォルダーと検索のオプション」をクリックし「フォルダーオプション」ダイアログを表示します。[表示]タブをクリックして、「登録されている拡張子は表示しない」のチェックをクリックして外し、[適用]ボタンをクリックしてダイアログを閉じます。

Windows 8 以降では、エクスプローラーの「表示」リボンにある[ファイル名拡張子]のチェックボックスにチェックをいれます。

Windows 7



Windows 8 以降



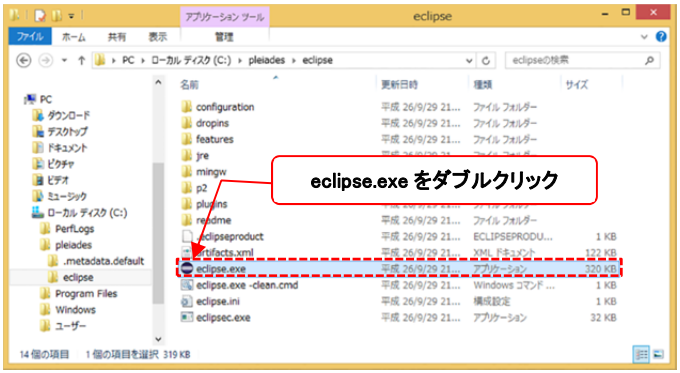
1.6 はじめてのＣ言語プログラミング

Eclipse のインストールが完了したら、正しくインストールされているか簡単な C 言語のプログラムを作ってみましょう。

1.6.1 Eclipse を起動する

Eclipse を起動するには、エクスプローラーで、C:¥pleiades¥eclipse フォルダーを開き、eclipse.exe をダブルクリックします。

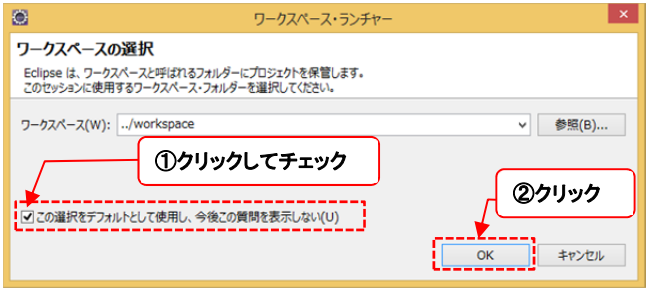
この eclipse.exe が Eclipse の実行ファイルです。



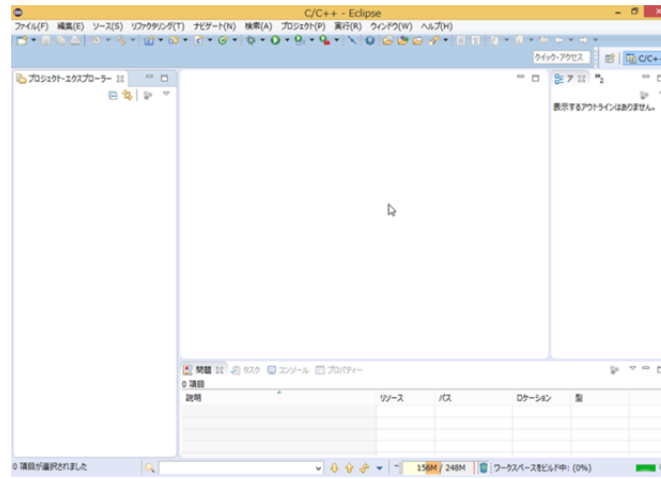
※eclipse.exeは、今後何度も起動するので、デスクトップにショートカットを貼り付けておくとよいでしょう。

「ワークスペース・ランチャー」というウィンドウが開きます。「ワークスペース」とは、Eclipse が作業するフォルダーのことです。

本書では、デフォルトのフォルダー「C:¥pleiades¥workspace」で作業するので「この選択をデフォルトとして使用し、今後この質問を表示しない」チェックボックスをクリックしてチェックを入れ、[OK]ボタンをクリックします。



Eclipse が、起動します。

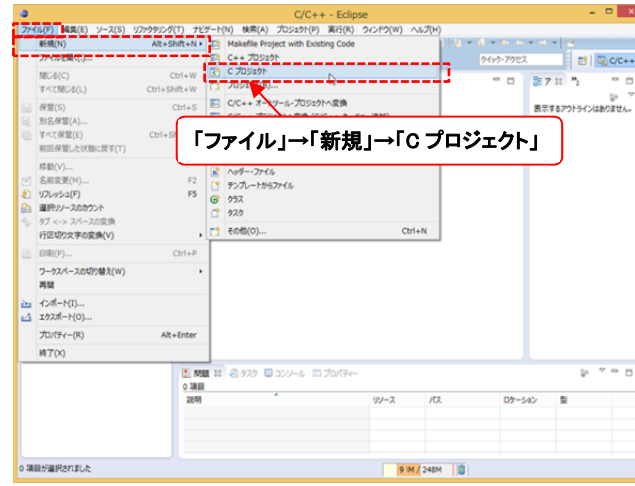


1.6.2 Ｃプロジェクトを作成する

Eclipse で C 言語のソースコードを入力しコンパイルして実行ファイルを作成するには、ソースファイル以外にも多くのファイルが必要です。これらのファイルを用意してプログラムがコンパイル、実行できる環境を整えるため、Eclipse では「プロジェクト」を作ります。

それでは、「test」という名前のプロジェクトを、作ってみましょう。

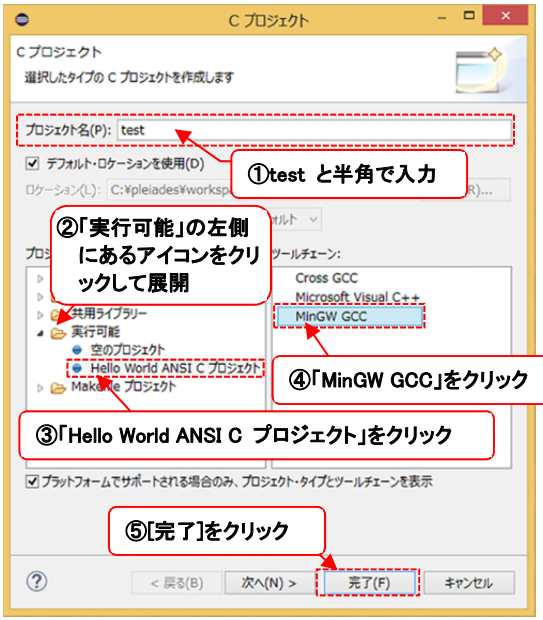
Eclipse の「ファイル」メニューをクリックして、表示されたメニューから「新規」→「C プロジェクト」を選択します。



「C プロジェクト」ダイアログが開くので、プロジェクトの名前、プロジェクトの種類、コンパイラなどのツールの種類を決めます。

どんな名前でもかまいませんが、ここでは「test」と半角で入力し、「プロジェクトタイプ」欄で「実行可能」－「Hello World ANSI C プロジェクト」をクリックして選択します。

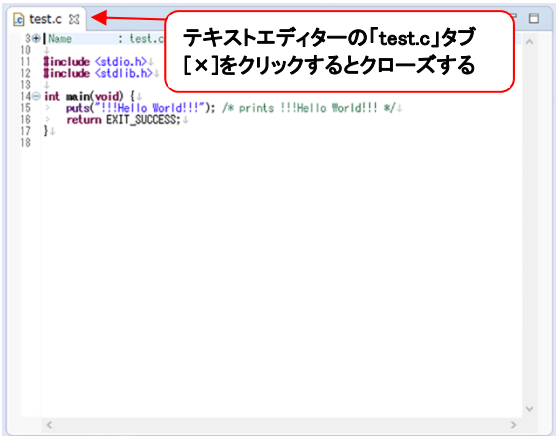
次に、「ツールチェーン」欄にある「MinGW GCC」をクリックして選択したら、[完了]ボタンをクリックします。



Eclipse の中央にテキストエディターが出現し、「test.c」というタブが表示されます。C 言語のソースファイルの拡張子は、通常「.c」にします。

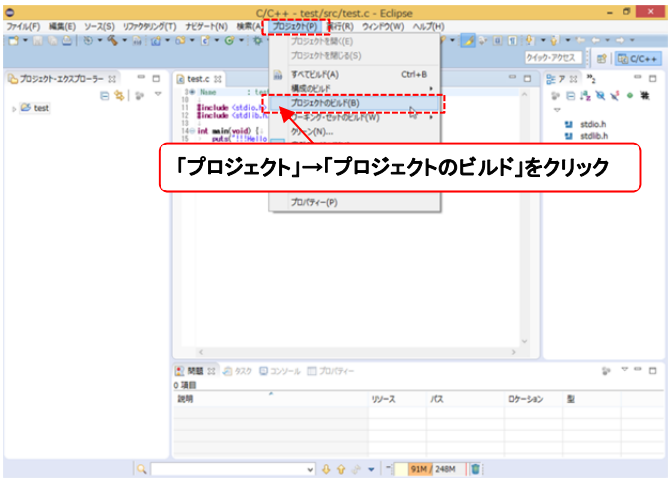
また、Eclipse のエディターは、タブで複数のソースファイルを切り替えながら編集ができます。また、編集を行わない場合は、タブの右側にある[x]ボタンをクリックしてクローズすることができます。

エディター上のソースファイルには、すでにソースコードのひな型が表示されます。



1.6.3 ビルドする

Eclipse では、ソースファイルをコンパイル、リンクして実行ファイルを作ることを「ビルド」と呼んでいます。そこで、Eclipse の「プロジェクト」メニューをクリックして、表示されたメニューから「プロジェクトのビルド」を選択します。

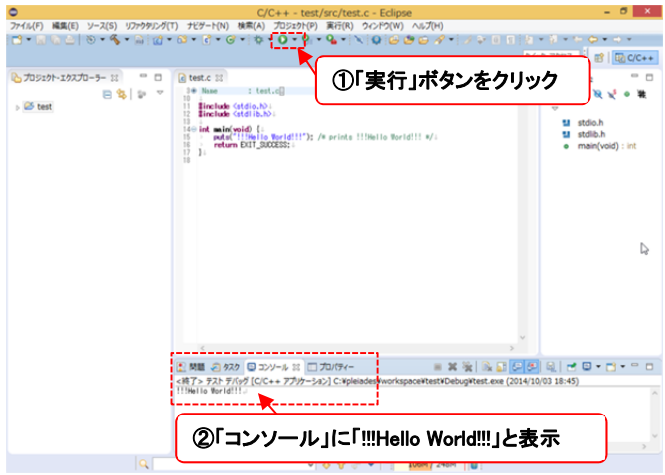


下の「コンソール」タブをクリックすると、「Build Finished」という文字が表示されコンパイルを行った様子が表示されます。

表示されている「src¥¥test.o」がオブジェクトファイル、「test.exe」が、実行ファイルです。このように、Windows の場合、実行ファイルは、拡張子が「.exe」になります。

「gcc」と表示されているのは、コンパイラとリンカの役割をするプログラムの名前です。

これで、実行ファイルができたので[実行]ボタンをクリックしてみましょう。コンソールに「!!!Hello World!!!」と表示されます。



注意　ここで[実行]ボタンをクリックするとエラーになる場合は、Eclipse のメニューにある「実行」メニューをクリックして、表示されたメニュー項目から「実行構成」を選択してください。「実行構成」ダイアログが表示されるので、左側にある「C/C++アプリケーション」をクリックして展開し「テストデバッグ」項目を選択します。すると、右側には「メイン」タブが現れます。「C/C++アプリケーション」が「Debug¥test.exe」になっていることを確認してください。
もしなっていない場合は、[参照]ボタンをクリックして「C:¥pleiades¥workspace¥test¥Debug¥test.exe」を選択します。また、「プロジェクト」が「test」であることも確認してください。もし、違う場合は[参照]ボタンをクリックして、「プロジェクトの選択」ダイアログを表示し、test プロジェクトを選択して[OK]ボタンをクリックします。これで、[実行]ボタンで実行できるようになるはずです。

1.6.4 プログラムを変更してみよう

それでは、「"!!!Hello World!!!"」の部分を自分の名前に変更して、自分の名前がコンソールに表示されるようにしてみましょう。

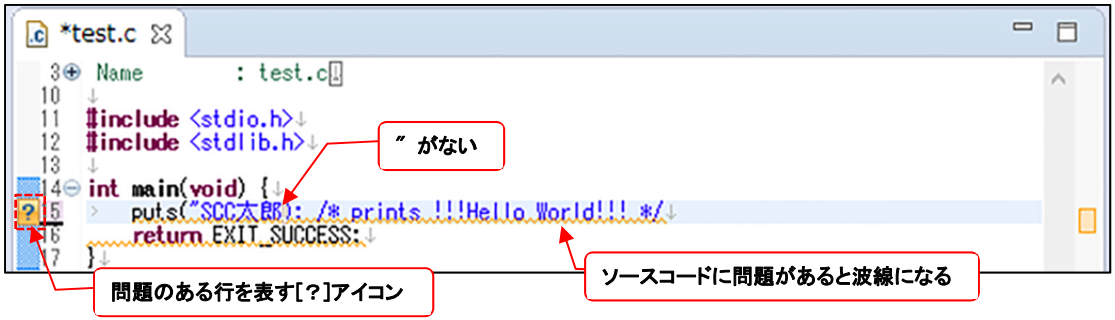
Eclipse の中央に表示されているのがテキストエディター（以降エディター）です。エディターを見ると「puts("!!!Hello World!!!"); /* prints !!!Hello World!!! */」と記述されている部分があります。ここの「"!!!Hello World!!!"」の部分を、以下のように自分の名前に書き換えてみましょう。

ここは、あなたの名前にしてください

```
puts("SCC 太郎"); /* prints !!!Hello World!!! */
```

このとき、コンソールに表示する文字列は「"（ダブルクォーテーション）」で括るので注意してください。

入力の際、誤って「"(ダブルクォーテーション)」を消してしまったような場合、エディターはエラーを表示します。



エラーの表示は、アイコンとソースコード上の波線で表現され、マウスカーソルを上にのせると「ヒント」がポップアップして表示されます。

また、Ｃ言語のソースコードは、すべて半角文字で入力します。全角文字は使用できません。ただし、「″(ダブルクォーテーション)」で括った部分は「文字列」として判断され全角文字が使用できます。ですから、コンソールに表示する文字列は「″」で括る必要があります。

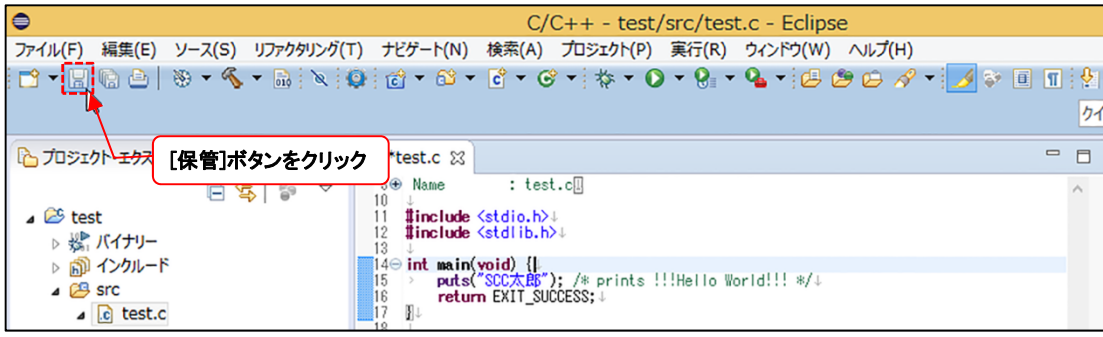
全角文字と半角文字

日本語の文字をワープロやエディターなどで入力する場合、日本語変換ソフトを起動して入力する「全角文字」と、オフにして(直接入力状態)入力する「半角文字」があります。日本語のひらがなと漢字は全角文字ですが、アルファベットや数字は全角文字と半角文字の両方が存在します。

この全角と半角のアルファベットと数字は、見た目は非常によく似ていますが、コンピュータにとっては、まったく別の文字になります。

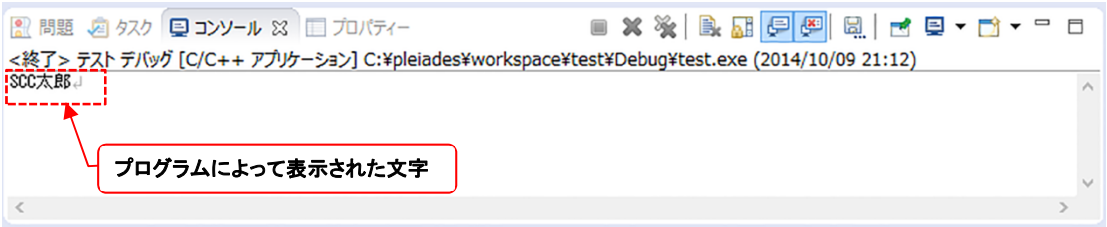
コンピュータは、文字を数値(番号)で識別しています。この文字の番号のことを「文字コード」と呼びます。人間にとっては全角の「A」も半角の「A」も同じ文字ですが、それぞれ異なる番号が与えられているため、違う文字として判断されます。

それでは、ソースコードの入力が完了したら、[保管]ボタンをクリックして保存します。



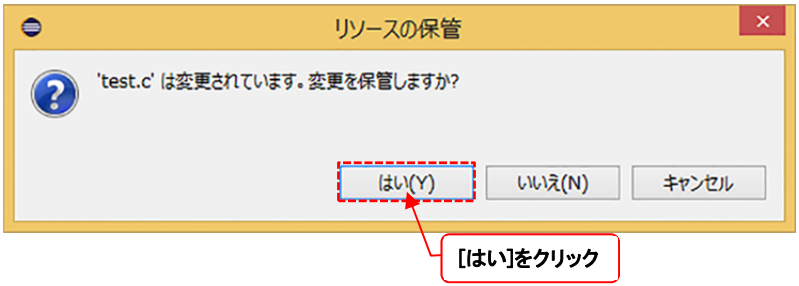
ソースコードが保管されると、エディターのファイル名のタブにある「*(アスタリスク)」が消え、保管済みであることを表します。

それでは、Eclipse の[実行]ボタンをクリックして実行しましょう。名前がコンソールに表示されるはずです。



これで、動作検証は終了です。

Eclipse を終了するときは、Eclipse のクローズボタンをクリックします。プロジェクトで利用したファイルが保存済みの場合そのまま終了します。もし、未保存のときは、ファイルを保存するかどうかを確認するメッセージボックスが表示されます。[OK]もしくは[はい]ボタンをクリックして保存してください。



1.6.5 サンプルコードのダウンロード

本書で紹介するサンプルコードのテキストが、下記のサイトからダウンロードできます。

URL　http://www.scc-kk.co.jp/scc-books/support/B-379/support.html

ダウンロードするファイルは、Sample.zip です。Sample.zip のダウンロードが完了したら、エクスプローラーで表示してダブルクリックします。

圧縮ファイルの中には「Sample」というフォルダーが入っているので、マウスで Ｃドライブのルートにドラッグアンドドロップしてコピーしておきます。

1.7 第１章のまとめと練習問題

第１章で説明してきた内容のまとめと、正しく理解しているか簡単な三択のテストをしてみましょう。

1.7.1 第１章のまとめ

1. プログラミングとは、「目的の処理をコンピュータに実行させるため、必要な命令を組み立てる行為」のことです。
2. プログラミングで組み立てた命令のまとまりを「プログラム」、プログラムを作成する人を「プログラマー」と呼びます。
3. コンピュータは、メモリーにプログラムを保存し、CPUに転送しながら、処理を実行していきます。
4. メモリーの記憶素子は、電圧のあるなしを１と０で表現するため、コンピュータは２進数でデータを記憶します。
5. メモリーが記憶する１番小さな単位を「bit」、bit が８個で「byte」と呼びます。メモリーの記憶容量は「byte」で表現します。
6. メモリーの記憶素子は、1byte 単位に「アドレス」というデータの場所を表す番号が割り当てられています。
7. CPUが直接理解できる命令を、「マシン語」と呼びます。
8. マシン語に １ 対 １ で対応するアルファベットや記号（ニーモニック）を使いプログラムを書く言語を「アセンブリ言語」と呼びます。
9. ほかのCPUやコンピュータで、同じプログラムが動くように作り直すことを「プログラムを移植する」といいます。
10. 人間が理解しやすい文法で記述できるプログラミング言語を「高級言語（または、高水準言語）」と呼びます。
11. 高級言語のプログラムを、マシン語に変換する作業を「コンパイル」、変換するプログラムを「コンパイラ」と呼びます。

12. コンパイルを行うプログラミング言語を、「コンパイル方式のプログラミング言語」と呼びます。
13. アルゴリズムとは、プログラミングにおいて「問題を解決するための手順」を表現したものです。
14. アルゴリズムを説明する際、処理の流れを「フローチャート」で表現することができます。
15. プログラムの処理の流れには、大きく分けて「直線」「分岐」「繰り返し」の 3 種類があります。
16. Ｃ言語は、1972 年にアメリカ AT&T 社のベル研究所で、D. M. Ritchie 氏とB. W. Kernighan 氏によって開発された言語です。
17. OSとは、コンピュータの「基本ソフトウェア」のことで、キーボート入力やディスプレイ表示といった基本機能をアプリケーションに提供します。
18. プログラムを記述したテキストを「ソースコード」、そのファイルを「ソースファイル」と呼びます。
19. ソースファイルのプログラムをOSなどのプラットフォーム上で動作させるには、「実行ファイル」にする必要があります。
20. ソースファイルはコンパイルすると「オブジェクトファイル」になります。このオブジェクトファイルに、関数ライブラリやスタートアップルーチンを組み込み実行ファイルにします。このような作業を「リンク」、リンクするプログラムを「リンカ」と呼びます。
21. アプリケーション開発に必要なさまざまなツールを統合した環境のことを「統合開発環境（IDE）」と呼びます。
22. Eclipse とは高機能な IDE のことで、プラグインにより機能を拡張することができます。
23. Ｃ言語のソースファイルの拡張子は、通常「.c」です。

1.7.2 練習問題

■ 問題 1

0 と 1 だけで数値を表現できるものは、次のうちどれですか。

1. 2 進数
2. 10 進数
3. 16 進数

■ 問題 2

メモリーの中に記憶されているデータの場所を調べるには、次のどれを利用しますか。

1. bit
2. byte
3. アドレス

■ 問題 3

マシン語に 1 対 1 で対応するニーモニックを使いプログラムを作る言語は、次のうちどれですか。

1. アセンブリ言語
2. 高級言語
3. コンパイル

■ 問題 4

高級言語のプログラムをマシン語に変換するプログラムを、なんと呼びますか。

1. コンパイル
2. コンパイラ
3. アセンブラ

■ 問題 5

アルゴリズムの説明などで、処理の流れを表現するには、次のどれを利用することができますか。

1. アセンブラ
2. コンパイル
3. フローチャート

■ 問題 6

3 種類あるプログラムの処理の流れに、含まれないものはどれですか。

1. 曲線
2. 直線
3. 分岐

■ 問題 7

OS などのプラットフォーム上で、実際に動作するファイルは、次のうちどれですか。

1. テキストファイル
2. ソースファイル
3. 実行ファイル

■ 問題 8

オブジェクトファイルに、関数ライブラリやスタートアップルーチンを組み込むことを、なんと呼びますか。

1. コンパイルする
2. リンクする
3. 実行する

■ 問題 9

統合開発環境を利用する利点で、間違っているものはどれですか。

1. 統合開発環境を使うことで、開発に必要なツールを個別に用意しなくても良くなるから。
2. 統合開発環境を使うことで、コードの入力、デバッグ、実行の作業がスピードアップするから。
3. 統合開発環境でコンパイルした方が、プログラムの処理速度がアップするから。

■ 問題 10

C 言語の拡張子は、次のうちどれですか。

1. .exe
2. .c
3. .txt