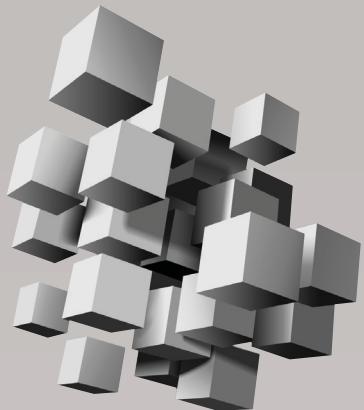


オンラインジャッジではじめる C/C++プログラミング入門

会津大学 渡部有隆 [著]

Online Programming Challenge!



日本最大級の"オンラインジャッジ"サイトで
ゲーム感覚で楽しみながらC/C++をマスターしよう！

（アイヅ オンライン ジャッジ）



AOJ:AIZU ONLINE JUDGE

コース問題公式解説書！



オンラインジャッジではじめる C/C++プログラミング入門

会津大学 渡部有隆 [著]

Online
Programming
Challenge!



本書のサポートサイト

本書に関する追加情報等について提供します。続編「アルゴリズムとデータ構造入門」の情報についてもこちらでご案内予定です。

<http://book.mynavi.jp/support/pc/5110/>

- ・本書は提出されたプログラムの正しさ・効率の自動判定を行うオンラインジャッジシステム（Webアプリケーション）を利用したC/C++プログラミング学習書籍です。オンラインジャッジシステムとして、会津大学が開発するAizu Online Judge (AOJ) を活用します。

Aizu Online Judge (AOJ) <http://judge.u-aizu.ac.jp>

- ・利用は無料ですがAOJサイトで登録する必要があります。詳しくは書籍の第2章「オンラインジャッジ」を参照してください。サイトはいつでも自由に自分のペースで学習可能です。
- ・本書の内容は2014年5月現在利用できるAizu Online Judge(AOJ)で解説をしています。書籍発刊後、サイトの情報や利用方法が大きく変更される可能性があります。あらかじめご了承ください。

Aizu Online Judge Version 1.0 ©2004 - 2014 AIZU Competitive Programming Club, Database Systems Lab.
University of Aizu

- ・本書に記載された内容は情報の提供のみを目的としています。本書の制作にあたっては正確な記述に努めましたが、著者・出版社のいずれも本書の内容について何らかの保証をするものではなく、内容に関するいかなる運用結果についてもいっさいの責任を負いません。本書を用いての運用はすべて個人の責任と判断において行ってください。
- ・本書に記載の記事、製品名、URL等は2014年5月現在のものです。これらは変更される可能性がありますのであらかじめご了承ください。
- ・本書に記載されている会社名・製品名等は、一般に各社の登録商標または商標です。本文中では©、®、™等の表示は省略しています。



はじめに

本書は、オンラインジャッジシステムといわれるプログラムの自動採点システムを用いてプログラミングを学習するための参考書です。本書は上級者を対象とした敷居の高いものではなく、オンラインジャッジのコース問題を題材とした入門書となっています。また、今後発刊予定の「オンラインジャッジではじめるアルゴリズムとデータ構造入門」とあわせて、C/C++によるプログラミングの入門から、精選したアルゴリズムの実装までを途切れなく学習することができます。これからプログラミングを始める方が、モチベーションを維持しながらプログラマとしての基礎体力を身に付けられる本を目指しました。

プログラミングとアルゴリズムを学ぶことは次の点で意義があります：

■**専門職** 全てのものがコンピュータで制御されている今、ソフトウェアを作成するプログラマという職業がますます重要になると考えられます。競争に勝つためには、数学力に加え、より創造的なプログラマであることが望ましいですが、求められる資質は平凡な問題に対して正確に仕様を理解し、バグ¹や性能の欠陥がないプログラムをより速く作成できる能力です。

■**教育ツール** 問題を解く手順をアルゴリズムといいます。アルゴリズムをコンピュータに伝える作業がプログラミングです。より効率的なアルゴリズムの設計には論理的思考力が必要になり、パズルを解くようにプログラムを組み立てます。これは、小中学生をはじめ、より若い世代の人たちが触れるべき教育ツールのひとつで、問題解決能力を養うために役立つと考えられます。

■**自ら作る** 何かやりたいことがあるときに、プログラムを書くことができれば、コンピュータの計算力と資源をフルに活用し、自分が形にしたいものを自ら実現することができるようになります。そのためには、ツールとしての言語やライブラリ²だけでなく、アルゴリズムの知識やそれらを自ら考え出す力が必要になってきます。

プログラミングのはじめ方はいろいろあります。ゲームなどのアプリケーション開発を通してプログラミングを学べば、目標が明確になり、完成した作品から達成感も得られるでしょう。既存のプログラミング言語やアルゴリズムの書籍からは多くのことを学ぶことができますし、演習問題のサンプルコードはすぐに実用にも応用できるでしょう。

一方、本書の特徴は、オンラインの判定システムを通して与えられた課題を1つ1つ解していくことです。本書からは、成果として本格的なアプリケーションは得られません。

¹ プログラムに含まれる誤りをバグ（虫）といいます。

² 使用頻度の高いプログラムを再利用可能な形で集めたものをライブラリといいます。

判定システムから合格（正解）をもらうことは簡単ではなく、時には苦しむこともあるでしょう。多くの場合、書いたプログラムにはバグが埋め込まれてしまいます。実行に時間のかかってしまう効率の悪いプログラムを書いてしまうこともあります。オンラインジャッジは、このようなプログラムの欠陥の有無を教えてくれます。一方で、問題を自力で解決した末に受ける正解判定は、ささやかな喜びを与えてくれます。それらの積み重ねにより、達成感と楽しみを感じながら学習を進めることができるでしょう。さらに、このような判定システムを用いた学習を反復することで、モチベーションが維持できるだけでなく、以下の資質が得られることが期待できます：

- 文書から要件を正確に理解し、仕様に忠実でバグのないプログラムを意識したコーディングができるようになる。
- 計算効率・メモリ使用量などの、コンピュータの資源を意識した設計・コーディングができるようになる。
- 情報処理技術者・プログラマとして最低限必要な基本的アルゴリズムとデータ構造の幅広い知識が身に付く。

なにより魅力的なことは、オンラインジャッジを使ったプログラミングではゲーム感覚で学習が進められることです。プログラミングをしたいけど、何を作っていくか分からぬ、という学生さんを多く見かけます。そのような方にはオンラインジャッジの活用をお勧めします。プログラミングを1つの趣味として見出すこともできるでしょう。

本書では、Aizu Online Judge のコースの問題を題材としています。本書は、続編の「オンラインジャッジではじめるアルゴリズムとデータ構造入門」とあわせて右のような構成になっています（図1：本書の内容を参照）。

図の矢印は主な前提知識を表します。ある章を始める前に、それを学習するために必要な知識がおおむね身についているよう配慮しました。■はオンラインジャッジのトピック番号に対応しています。トピック問題を順番に解きやすいように、章の適切な位置に演習問題を配置しました。道のりは長いですが、入門からでも基本の積み重ねで高度なアルゴリズムが実装できるようになります。



本書の内容

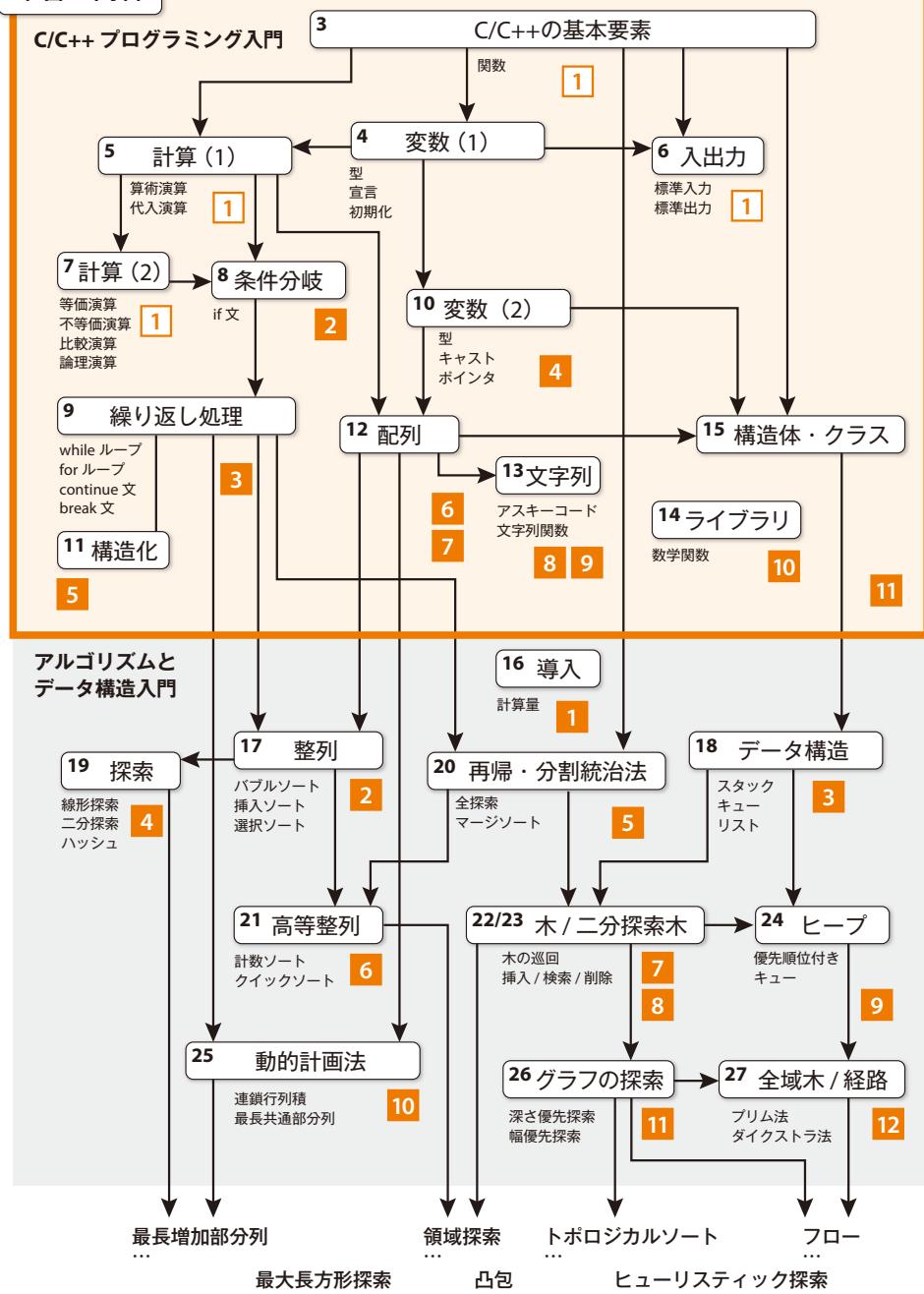


図1：本書の内容

本書「オンラインジャッジではじめるC/C++プログラミング入門」は以下の第1部と第2部から構成されており、次の特徴を持っています。

- 第1部では、プログラミングについての基礎知識とオンラインジャッジについて紹介し、Aizu Online Judge のチュートリアルをベースにその使い方を簡単に説明します。
- 第2部では、プログラミング入門として、体系的に整理されたC/C++言語の基礎的な言語要素を解説します。簡単な入出力例付きのサンプルプログラムを用いてプログラムの動作原理を理解した上で、オンラインジャッジの入門レベルの問題を自力で解いていきます。

本書では、基礎的なものに限りますが、C/C++言語の主な構文や機能を一通り身につけることができます。

具体的には、変数、入出力、条件分岐、繰り返し、配列を用いたプログラムを作成できるようになります。これらの知識を身につければ、プログラマとしての次のステップである「アルゴリズムとデータ構造」を自ら実装することができるようになります。

続編として今後発刊予定の「オンラインジャッジではじめるアルゴリズムとデータ構造入門」は以下の第3部と第4部から構成されています。

- 第3部では、体系的に整理された基本的なアルゴリズムとデータ構造に関する問題を解いていきます。解説では、疑似コードからアルゴリズムの動作を理解し、本書「オンラインジャッジではじめるC/C++プログラミング入門」の第2部で得た知識のみで、問題を解くアルゴリズムを実装することを目標とします。
- 第4部では、精選トピックとして、いくつかの応用問題を取り上げ、第3部で身に付けたアルゴリズムとデータ構造の応用やそれらの組み合わせで、難しい問題を解くことを目標とします。



本書「オンラインジャッジではじめるC/C++プログラミング入門」および発刊予定の「オンラインジャッジではじめるアルゴリズムとデータ構造入門」を通して課題をクリアした後は、プログラマとしての最低限の基礎体力が身についていることと思います。一方、プロとしてのプログラマを目指すためには、より言語仕様に精通していかなければなりません。C言語をさらに学ぶためには例えば「C言語によるプログラミング」(オーム社)などがおすすめです。Webアプリ、各種ゲームを作成するためには、異なるプログラミング言語やライブラリの使い方など新しい知識が必要になりますが、目的に特化した参考書は豊富にあります。

一方、より高度なアルゴリズムを知りたい、アルゴリズムの研究をしてみたい、思考力を鍛えたい、あるいは趣味としてプログラミングを楽しみたい方は、競技プログラミングの世界をお勧めします。本書では扱わない各種プログラミングコンテストの過去問題にチャレンジしたり、定期コンテストに参戦するとよいでしょう。「プログラミングコンテストチャレンジブック [第2版]～問題解決のアルゴリズム活用力とコーディングテクニックを鍛える～」(マイナビ) や「最強最速アルゴリズマー養成講座 プログラミングコンテストTopCoder攻略ガイド」(SBクリエイティブ)などの参考書も出版されており、学習環境も整っています。

本書を活用していただくために必要な前提知識は、エディタやWebブラウザの使い方、ソフトのインストールなどのパソコンの基本操作と、高校レベルの数学だけです。またオンラインジャッジによる演習だけでなく、解説・疑似コード・サンプル解答コードが豊富なので、オンライン環境がなくても十分活用していただけます。

本書は、著者が会津大学公開講座・出前講義用に作成したプログラミング入門・アルゴリズム入門テキスト（過去5年）および会津大学学部科目アルゴリズムとデータ構造の演習問題（過去3年）の内容を参考にして作成されました。



目 次

第Ⅰ部 学習の準備	1
第1章 プログラミング	3
1.1 プログラミング言語	4
1.2 プログラムの作成から実行まで	5
1.3 プログラミングの学習方法	9
1.4 学習環境の設定	10
1.4.1 Cygwin	10
1.4.2 オンラインコンパイラー	10
第2章 オンラインジャッジ	11
2.1 オンラインジャッジシステムとは？	12
2.2 ユーザ登録する	14
2.3 問題を観覧する	16
2.3.1 問題の種類	16
2.3.2 ファインダーから探す	17
2.3.3 コースから探す	17
2.4 問題を解く	19
2.4.1 問題文を読む	19
2.4.2 プログラムを提出する	21
2.4.3 判定結果を確認する	22
2.5 マイページ	26
2.6 その他の機能	28
2.7 本書での活用方法	28
第Ⅱ部 プログラミング入門	29
第3章 基本構造	31
3.1 プログラムの基本要素：関数	33
3.1.1 関数の構造	34
3.2 C++言語の基本構造	37
3.2.1 main関数	37
3.2.2 関数の呼び出し	37
3.2.3 C++言語の書式	38
3.2.4 コメント	39
3.3 ライブラリ	40
3.4 演習問題	42



第4章 変数（1）	43
4.1 変数	44
4.1.1 変数の型	45
4.1.2 変数の宣言	45
4.1.3 変数の初期化	46
4.1.4 参照と代入	46
第5章 計算（1）	47
5.1 計算式	48
5.1.1 演算子	48
5.1.2 算術演算子	49
5.1.3 代入演算子	50
5.1.4 優先順位と結合法則	50
5.2 演習問題	52
第6章 入出力	53
6.1 標準出力	55
6.1.1 printf関数による出力	55
6.1.2 coutによる出力	57
6.2 標準入力	60
6.2.1 scanfによる入力	60
6.2.2 cinによる入力	62
6.3 演習問題	64
第7章 計算（2）	65
7.1 代入演算子	67
7.2 インクリメント・デクリメント演算子	69
7.3 等価演算子・不等価演算子	70
7.4 比較演算子	73
7.5 論理演算子	75
7.6 演算子の優先度と結合法則	77
7.7 演習問題	79
第8章 構造文:条件分岐	81
8.1 条件分岐	83
8.1.1 if文	83
8.1.2 if-else文	84
8.1.3 if-else if-else文	86
8.1.4 三項演算子	88
8.2 演習問題	89
第9章 構造文：繰り返し処理	93
9.1 繰り返し処理	95
9.1.1 while ループ	95
9.1.2 for ループ	96

9.2	演習問題（1）	98
9.3	繰り返し処理の制御	99
9.3.1	break文	99
9.3.2	continue文	102
9.3.3	return文	104
9.4	演習問題（2）	106
第10章 変数（2）		109
10.1	型の表現範囲	111
10.2	型変換	113
10.2.1	混合演算の型変換	113
10.2.2	代入演算の型変換	114
10.2.3	キャスト演算	115
10.3	ポインタ	116
10.4	演習問題	119
第11章 プログラムの構造		123
11.1	文の構造	125
11.2	変数のスコープ	127
11.2.1	大域変数	128
11.2.2	局所変数	128
11.3	演習問題	129
第12章 配列		135
12.1	配列	137
12.1.1	配列の宣言	138
12.1.2	配列の要素へのアクセス	138
12.1.3	配列の初期化	138
12.2	演習問題（1）	140
12.3	多次元配列	141
12.3.1	多次元配列の宣言	142
12.3.2	多次元配列要素へのアクセス	143
12.3.3	多次元配列の初期化	143
12.4	演習問題（2）	144
12.5	演習問題（3）	150
第13章 文字列		155
13.1	文字	157
13.1.1	文字とアスキーコード	157
13.1.2	文字の関数	158
13.2	演習問題（1）	160
13.3	文字列: char型配列	161
13.3.1	char型配列	161
13.3.2	文字列の宣言	162



13.3.3 文字列の初期化	162
13.3.4 文字列の入出力	162
13.3.5 文字列の基本操作	163
13.3.6 文字列関数	164
13.4 演習問題	167
13.5 <code>string</code> クラスによる文字列	171
13.5.1 文字列の宣言	172
13.5.2 文字列の初期化	172
13.5.3 文字列の入出力	172
13.5.4 文字列の基本操作	173
13.5.5 文字列の関数	175
13.6 演習問題	179
第14章 ライブラリ	183
14.1 数学関数	185
14.2 演習問題	187
第15章 構造体とクラス	191
15.1 構造体	193
15.1.1 構造体の定義	193
15.1.2 構造体の使用	195
15.1.3 構造体の配列	196
15.2 クラス	198
15.2.1 クラスの定義	198
付録 演習問題模範解答	207
1 Getting Started	208
ITP1 1 A: Hello World	208
ITP1 1 B: X Cubic	208
ITP1 1 C: Rectangle	209
ITP1 1 D: Watch	209
2 Branch on Condition	210
ITP1 2 A: Small, Large, or Equal	210
ITP1 2 B: Range	210
ITP1 2 C: Sorting Three Numbers	211
ITP1 2 D: A Circle in a Rectangle	211
3 Repetitive Processing	212
ITP1 3 A: Print Many Hello World	212
ITP1 3 B: Print Test Cases	212
ITP1 3 C: Swapping Two Numbers	213
ITP1 3 D: How Many Divisors?	213
4 Computation	214
ITP1 4 A: A/B Problem	214
ITP1 4 B: Circle	214
ITP1 4 C: Simple Calculator	215
ITP1 4 D: Min, Max and Sum	215

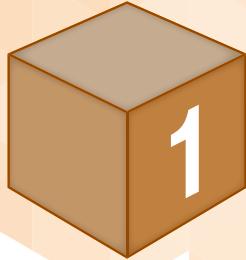
5 Structured Program I	216
ITP1 5 A: Print a Rectangle	216
ITP1 5 B: Print a Frame	217
ITP1 5 C: Print a Chessboard	218
ITP1 5 D: Structured Programming	219
6 Array	220
ITP1 6 A: Reversing Numbers	220
ITP1 6 B: Finding Missing Cards	221
ITP1 6 C: Official House	222
ITP1 6 D: Matrix Vector Multiplication	223
7 Structured Program II	224
ITP1 7 A: Grading	224
ITP1 7 B: How many ways?	225
ITP1 7 C: Spreadsheet	226
ITP1 7 D: Matrix Multiplication	227
8 Character	228
ITP1 8 A: Toggling Cases	228
ITP1 8 B: Sum of Numbers	229
ITP1 8 C: Counting Characters	230
ITP1 8 D: Ring	230
9 String	231
ITP1 9 A: Finding a Word	231
ITP1 9 B: Shuffle	232
ITP1 9 C: Card Game	233
ITP1 9 D: Transformation	234
10 Math Functions	235
ITP1 10 A: Distance	235
ITP1 10 B: Triangle	235
ITP1 10 C: Standard Deviation	236
ITP1 10 D: Distance II	237
11 Structure and Class	238
ITP1 11 A: Dice I	238
ITP1 11 B: Dice II	239
ITP1 11 C: Dice III	240
ITP1 11 D: Dice IV	241
参考文献	242
索引	242

第Ⅰ部

学習の準備

1章 プログラミング 3

2章 オンラインジャッジ 11



プログラミング

Programming

- 1.1 プログラミング言語**
- 1.2 プログラムの作成から実行まで**
- 1.3 プログラミングの学習方法**
- 1.4 学習環境の設定**

パソコン、スマートフォン、ゲーム機などのコンピュータは、入力に応じて計算結果を出力する機械です。文字が表示され、キャラクターが動き、シミュレーション結果が得られるのは、コンピュータの中に入力に応じて異なる出力をを行うソフトウェア、つまりプログラムが組み込まれているからです。

この章では、ソフトウェアを作成するためのプログラミング言語とその作業（学習）の流れを確認します。



1.1 プログラミング言語

コンピュータに命令を伝えるために使用するのがプログラミング言語です。人とコミュニケーションをとるための日本語や英語にも文法があるように、プログラミング言語にも文法が定義されています。プログラムを書いてコンピュータに命令を伝えるためには、言語のルールを覚えなければなりません。しかし心配する必要はありません。新しい言語を覚えるといつても、英語やロシア語などの自然言語を習得するよりも、プログラミング言語を習得することはずっと簡単です。プログラミング言語では、限られたキーワードと構文の使い方を憶えれば多くのアルゴリズムをプログラムとして実装¹することができます。また、必要に応じてライブラリの使い方を辞書を引くようにして調べるのが一般的なので、一度に多くのことを覚える必要はありません²。

一方、プログラミング言語は自然言語よりもずっと厳格に解釈され、少しでも文法を間違えればコンピュータは理解してくれません。人間のように相手の意図をくみ取ることも得意ではありません。それどころか、文法が正しければたとえバグがひそんでいても、そしてその内容が自分の意図しない内容であってもコンピュータは与えられた命令文をなんの間違いもなしに実行します。

これまでに様々なプログラミング言語が開発されて、用途によって使いわけられています。代表的なプログラミング言語を紹介します：

■C。「シー」と呼びます。マイクロコンピュータから大型コンピュータまで、多くのプラットフォーム³におけるアプリケーションがC言語で開発されています。プログラミングの自由度が高く、実行速度が速いことが特徴です。

¹ ソフトウェアの分野で、設計やアルゴリズムに基づいてプログラムを作成することを実装といいます。

² 新しいプログラミングパラダイムを習得することはある程度の努力が必要です。

³ アプリケーションを動作させる基盤となるOSや環境のことです。

■ **C++**。「シープラスプラス」と呼びます。C言語にオブジェクト指向⁴といわれるプログラミングパラダイムの考え方や開発効率を上げる標準ライブラリが追加されたものです。

■ **Java**。「ジャバ」と呼びます。機種に異存せずに様々なハードウェアで動かすことができるため、スマートフォンなどの携帯機器から大規模サーバまで、多くのシステムで活用されています。C/C++言語から多くのものを引き継いでおり、基本的な文法などはC/C++に類似しています。オブジェクト指向に基づいて設計された言語です。

本書で解説する言語はC++です。しかし、C++言語はC言語の拡張版であり、C言語の要素を引き継いでいます。C++言語を習得するということはC言語を知ることにもなります。C++言語では充実したライブラリを使うことができますが、一方で純粋なC言語でのプログラミングも行うことができます。ただし、汎用的な機能であってもC言語にはできない部分もあるので、C++言語を最初に学んでC言語を使う場合には注意が必要です。



1.2 プログラムの作成から実行まで

様々なプログラミング言語の中でも、人間にとて分かり易いプログラミング言語を高級言語といいます。C、C++、Javaなどは高級言語に属します。人間は訓練をすればこれらの高級言語の構文・意味を理解し使用できるようになります。

一方、コンピュータは高級言語を直接理解することができません。高級言語で書かれたプログラムはコンピュータが理解できる機械語に翻訳してから実行する必要があります⁵。プログラムの実行方法には主に以下の2つのタイプがあります：

コンパイラによる翻訳・実行



図1.1：コンパイラによる翻訳・実行

⁴ データとそれに対する操作を「オブジェクト」として管理し、それらの組み合わせによってソフトウェアを構築する考え方。

⁵ 機械語と高級言語の中間に位置するアセンブリ言語というものがあります。

高級言語で書かれたプログラムを機械語にコンパイル(翻訳)してから実行する方式です。コンパイルはコンパイラといわれるソフトウェアを使って行います。コンパイル・実行という2つのステップが必要ですが、プログラムの実行速度が速いという利点があります。

インタプリタによる逐次実行



図1.2：インタプリタによる逐次実行

高級言語で書かれたプログラムをインタプリタというソフトウェアによって逐次翻訳して実行する方式です。命令を1つ1つ解釈して実行するので、コンパイラ方式よりも実行速度が遅いという欠点があります。

C/C++言語は前者の方式で実行します。C/C++言語のコンパイラにはそれぞれgccとg++を使います。多くの場合g++でC言語とC++言語で書かれたプログラムをコンパイルすることができます。



図1.3：C++ プログラムのコンパイル・実行

本書のプログラムは全てg++でコンパイルすることにします。通常、プログラムはまとった機能ごとに1つのテキストファイルとして作成します。C言語で書かれたプログラムのファイルには「.c」、C++言語で書かれたプログラムのファイルには、「.cpp」という拡張子付けています。

実際のプログラムがどのようなものになるか見てみましょう。次のProgram 1.1は入力された2つの整数の和を出力するプログラムです。

Program 1.1：2つの整数の和を計算するプログラム

```
1 #include<cstdio>
2
3 int main(){
4     int a, b;
5     scanf("%d %d", &a, &b);
6     printf("%d\n", a+b);
7     return 0;
8 }
```

ここでは、コンパイル・実行の方法に焦点を置くため、プログラムの詳しい内容にはふれません。adder.cpp というファイルに、Program 1.1 の内容がコピーされているものとします。コンパイルはコマンドとして

```
g++ adder.cpp
```

と入力し実行します。プログラムにエラーがあるとコンパイラはエラーメッセージを出力します。主に文法上の間違いが行番号付きで報告されますので、これを参考にプログラムを修正します。

エラーがなければ、メッセージは出力されず、a.out という実行ファイルが生成されます。この実行ファイルを

```
./a.out
```

と入力し実行します。このプログラムは、標準入力（キーボード）から2つの整数が入力されるのを待ちます。2つの整数が入力されるとそれらを加算した結果を出力し、プログラムが終了します。

プログラムを開発する作業の流れは次のフローチャート⁶によって表すことができます：

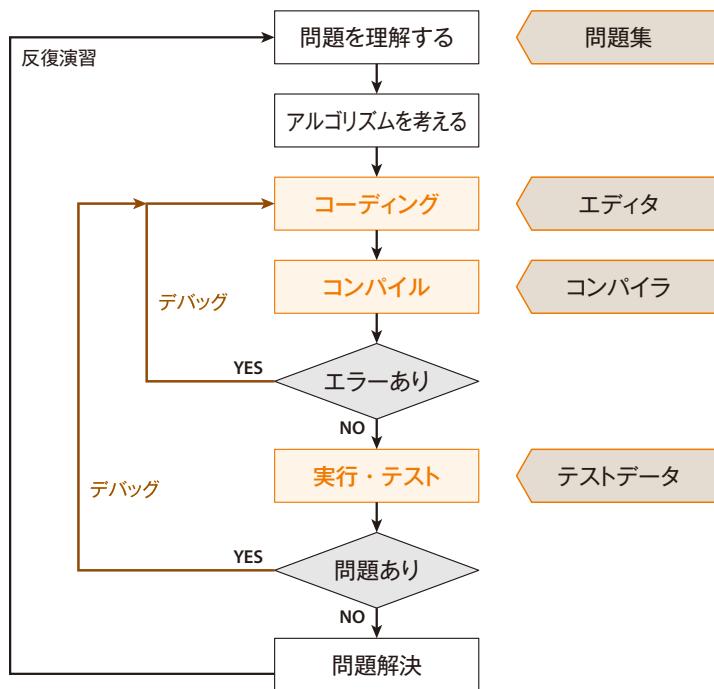


図1.4：プログラムを作成する流れ

まずははじめに、テキストエディタなどでコーディングを行います。ひと通りコーディングが終了したら、コンパイルを行います⁷。コンパイルの結果エラーが報告されたら、そのメッセージに従いプログラムの修正を行わなければならないので、再びコーディングに戻ります。プログラムを修正する作業をデバッグといいます。

コンパイルが成功したら実行ファイルを実行し、プログラムが正しく動作しているかを確認します。基本的に、様々な入力に対して意図した出力が得られているかをチェックします。

正しく動作していないことが発覚したら、デバッグを行うために再びコーディングへ戻ります。

⁶ 通常はプログラムの処理を可視化するために用いられる流れ作業図です。主に作業の流れを矢印、処理を長方形、分岐をひし形で表わします。

⁷ プログラムが大きい場合は、大量のエラーメッセージを回避するために、機能ごとにこまめにコンパイルを行いましょう。

同じ入力に対してプログラムを何度も実行することになるので、プログラムの入力はデータとしてファイルに作成しておきます。例えばProgram 1.1をテストするためには、2つの整数が書かれたファイル（例えばin.txt）を作成して、以下のようにリダイレクト⁸を使用してプログラムを実行します。

```
./a.out < in.txt
```

これはファイルに記録されているデータを標準入力としてプログラムを実行するコマンドで、毎回キーボードで入力することなく実行結果（出力）をすぐに確認することができます。

このように、プログラミングとは、コーディングから始まり、コンパイル・テスト・デバッグの繰り返し作業となります。

ポイント

ソフトウェア開発の現場では、フレームワークや開発手法に従った規則やツールが用いられます。本書では、個々の問題を解くための小規模なプログラムを多数作成することを想定しています。



1.3 プログラミングの学習方法

多くの問題をステップ by ステップで解いていくことにより、言語の仕様を覚えられるだけでなく、アルゴリズムを組み立てられるようになります。適切なトピックの順番で反復演習を行うことが、効率的な学習に繋がります。

プログラムは問題の仕様に基づいたあらゆる入力に対して十分効率良く出力を行う必要があります。仕様を満たすテストデータの作成は、プログラミングの自主学習においては手間がかかるものです。問題によっては、テキストに掲載しきれないほど大きくなり、考えられる入力のパターンを漏れなく作成することは容易ではありません。より信頼性の高いプログラムにするためには、厳格な入力データによりテストを行う必要があります。

オンラインジャッジにはたくさんの演習問題があり、各問題には、このような入力データとそれに対応する正しい出力データが準備されています。提出されたプログラムの正誤とその効率の判定を即座に行ってくれるので、プログラミングの自主学習に活用することができます。第2章でオンラインジャッジを紹介しその使い方を確認します。

⁸ プログラムの入力元や出力先を変更する機能です。



1.4 学習環境の設定

本書では、次章で解説するプログラムの自動採点システムであるオンラインジャッジシステムを用いて演習問題を解いていただきます。一方、提出するプログラムを完成させるための、エディタ、コンパイラ、実行環境はお使いのコンピュータに準備して頂く必要があります。オンラインジャッジの提出フォームに直接プログラムを書くこともできますが、より高機能なエディタと開発環境をお使いのコンピュータに準備していただくことをお勧めします。オンラインジャッジは主に演習問題の観覧、解答プログラムの自動採点ツールとして活用します。また、進捗の確認や各種ステータスの観覧などにご活用いただけます。

普段Linuxをお使いのユーザは特に環境設定の必要はありません。ここでは、Windowsでプログラミングを行うための環境の例としてCygwinといいくつかのオンラインコンパイラをご紹介します。

1.4.1 Cygwin

CygwinはWindows環境でLinuxのターミナルのようにコマンドを打つことで作業が進められる開発ツールです。インストール時に、エディタやコンパイラを同時にインストールし、簡単に環境を構築することができます。フリーソフトで以下のURLから入手することができます。

<http://www.cygwin.com/>

Cygwinのインストール方法や使い方の解説はWEB上に豊富にあります。本書ではその詳細は解説しませんが、検索エンジンで調べてみてください。

1.4.2 オンラインコンパイラ

オンラインコンパイラとは、Webブラウザのみを使ってソースコードを編集、コンパイル、実行することができるオンラインサービスです。より高機能になるとオンライン統合開発環境もあります。様々なオンラインコンパイラが開発されていますが、ここでは以下にあげるいくつかのサービスを紹介します。こちらもWeb上に解説が豊富にありますので本書では詳細を省きます。

- ideone <https://ideone.com/>
- compilr <https://compilr.com/>
- compileonline <http://www.compileonline.com/>



オンラインジャッジ

Online Judge

2.1 オンラインジャッジとは?

2.2 ユーザ登録する

2.3 問題を観覧する

2.4 問題を解く

2.5 マイページ

2.6 その他の機能

2.7 本書での活用方法



試し読みはお楽しみ
いただけましたか？

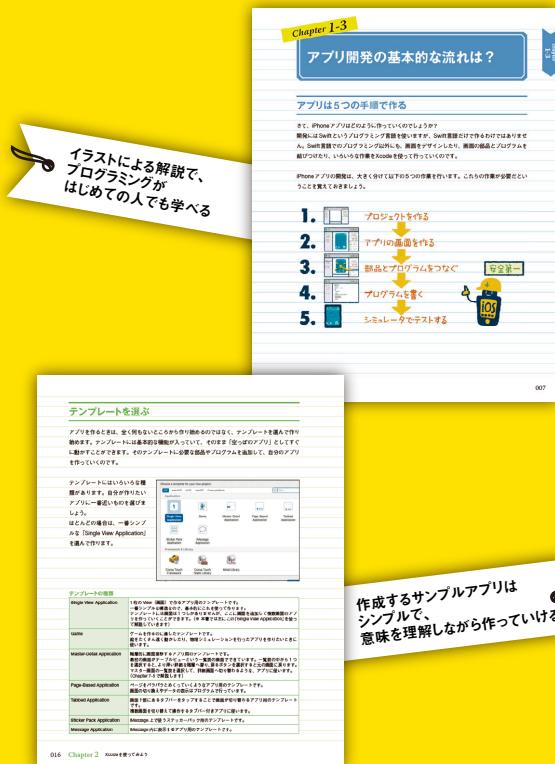
ここからはManatee
おすすめの商品を
ご紹介します。

Tech Book Zone
Manatee 

1

体験型の本書でプログラミングの 第1歩を踏みだそう!

『やさしくはじめるiPhoneアプリ作りの教科書【Swift 3&Xcode 8.2対応】』は、iPhoneアプリを作成する初心者のための入門書です。プログラミングが初めての人、苦手意識がある人でも楽しく学んでいけるよう、なるべくやさしく、イラストや図をたくさん使って解説しています。本書では実際にサンプルアプリを作りながら学んでいきますが、イラストによる解説で、一歩ずつ丁寧に、iPhoneアプリ作りの基本と楽しさを学べます。



やさしくはじめる iPhone アプリ作りの 教科書 [Swift 3 & Xcode 8.2 対応]



マイナビ出版
森巧尚(著者)、
まつむらまきお(イラスト)
312ページ 價格: 3,002円(PDF)

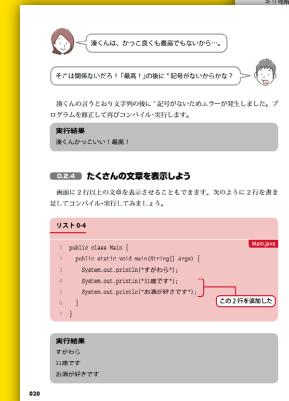
2

豊富なイラストで「なぜ?」を解消! Javaの第一歩を踏み出そう

『スッキリわかるJava入門 第2版』は、Javaの基礎から初心者には難しいとされるオブジェクト指向まで、膨らむ疑問にしっかり対応しました。Javaプログラミングの「なぜ?」がわかる解説と約300点の豊富なイラストで、楽しく・詳しく・スッキリとマスターできる構成となっています。「なんとなくJavaを使っているけれど、オブジェクト指向の理解には自信がない」「学習の途中で挫折してしまった」という方にもおススメです。

「プログラミング」

300ものイラストで
楽しく・詳しく・スッキリ!
マスター!



会話のやりとりの中にも、開発現場でのヒントが詰め込まれている

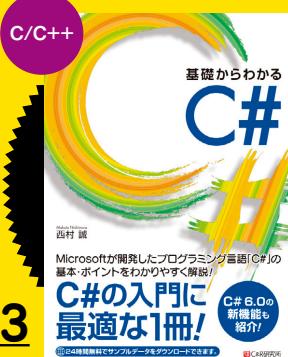
スッキリわかる
Java 入門 第2版

インプレス
中山清喬・国本大悟(著者)
658 ページ
価格: 2,376 円(PDF)





幅広いジャンルで活躍!
C# がキホンから学べる本



3

基礎からわかる C#

本書はプログラミングの経験がある人を対象とした、プログラミング言語「C#」の入門書です。C# の概要から基本的な文法、特徴的な機能まで、わかりやすく解説しています。C# 6.0 の新機能についても解説しています。

シーアンドアール研究所
西村誠(著者)

168 ページ 値格: 1,944 円(PDF)

プログラミング未経験でも
Android アプリを開発!



4

イラストでよくわかる
Android アプリのつくり方
Android Studio 対応版

親しみやすいイラストや、ステップバイステップでの丁寧な解説が基本コンセプト。開発環境「Android Studio」に対応し、Android のプログラムを作りながら、自然に Java というプログラミング言語の知識が身につきます。

インプレス
羽山博・めじろまち(著者)
価格: 2,138 円(PDF)

JavaScript を網羅的に
取り上げた骨太の 1 冊



5

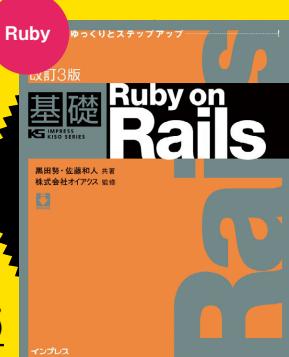
JavaScript 逆引きハンドブック

JavaScript の逆引きリファレンスの決定版。JavaScript の機能を網羅的に取り上げていて、骨太の 1 冊になっています。JavaScript の基本的な処理や便利な Tips はもちろん、HTML5 の API についても数多く掲載しています。

シーアンドアール研究所
古旗一浩(著者)

993 ページ 値格: 3,694 円(PDF)

初めてのウェブ開発も安心
Ruby の文法を基礎から解説



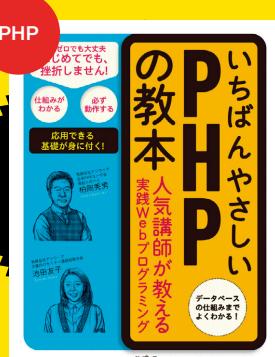
6

改訂 3 版
基礎 Ruby on Rails

Ruby の文法やオブジェクト指向の考え方を初歩から解説。アプリケーションのモックアップ作り、データベースを導入し、ログイン・ログアウト機能を加え、最終的にはメンバーや記事の管理ページまでできあがります。

インプレス
黒田努・佐藤和人(著者)
536 ページ 値格: 3,240 円(PDF)

初歩から順に理解できる
PHP とデータベース



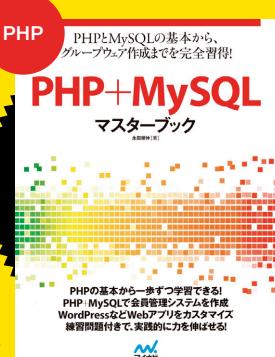
7

いちばんやさしい PHP の教本

PHP とデータベースの基本を順番に学んで、実践的なプログラムを完成させていく PHP の入門書です。大事なポイントや勘違いしやすいポイントは講師がフォロー。セミナーを受けている感覚で読み進められます。

インプレス
柏岡秀男・池田友子(著者)
240 ページ 値格: 1,836 円(PDF)

PHP5.4 の基本から
MySQL との連携まで!



8

PHP+MySQL マスター ブック

この一冊で PHP と MySQL の基本と Web アプリケーションの構築法について学習できる実践的なプログラミング入門です。現場必須のプログラム構築法、API の活用法から、セキュリティ技術まで詳しく解説します。

マイナビ出版
永田順伸(著者)
384 ページ 値格: 2,916 円(PDF)

はじめてプログラミングに触れる前に読んでおこう



9

最初に読みたい入門書！

プログラミングの世界へようこそ

全くの初心者がプログラミングを勉強したいとき、さまざまな疑問が湧いてきます。「どの言語を覚えればいい?」「文系でも大丈夫?」本書はプログラミングに触れる前に知っておきたい基本をイラスト付きで解説します。

マイナビ出版
尾川一行・中川聰(著者)
192 ページ 價格: 1,933 円(PDF)

プログラミングの初心者が Python 3 を学ぶのに最適



12

基礎 Python 基礎シリーズ

プログラミングの初心者を対象にした Python 3 の入門書です。変数の取り扱いから、リスト、タブルといった Python 固有のデータの操作、制御構造や関数などについて、初心者でも基礎から学習できるように説明しました。

インプレス
大津真(著者)
312 ページ 價格: 2,894 円(EPUB)

プログラミングの基本から手取り足取りじっくり解説



10

目標せプログラマー！
プログラミング超入門

本書は Windows 開発の標準ツールとも言える「Visual Studio」を使い、C# というプログラミング言語でプログラミングの基本を学びます。最終的には、ちょっとしたアクションゲームが作れるくらいになるのが目標です。

マイナビ出版
掌田津耶乃(著者)
312 ページ 價格: 2,074 円(PDF)

Go 言語の基礎から応用まで
ポイントがよくわかる



13

改訂2版
基礎からわかる Go 言語

Google が開発したプログラミング言語「Go」の基礎から応用までをわかりやすく解説した 1 冊です。最新の Go 1.4 のバージョンに対応して改訂しました。Linux、Mac OS X、Windows の各環境に対応しています。

シーアンドアール研究所 古川界(著者)
240 ページ 價格: 2,138 円(EPUB)

必ずアルゴリズムの意味がわかるようになる入門書！



11

楽しく学ぶ
アルゴリズムとプログラミングの図鑑

図解とイラストを豊富に使ったアルゴリズムの入門書。アルゴリズムとは「問題を解決するための考え方」です。それが分かってきたら、8 種類のプログラミング言語を使ったサンプルプログラムを実際に試しましょう。

マイナビ出版 森巧尚(著者)、まつむらまさお(イラスト)
300 ページ 價格: 2,689 円(PDF)

R 言語の機能を目的から見つけ出せる！



14

改訂3版
R 言語逆引きハンドブック

本書では、最新バージョンの R 3.3.0 に対応し、R 言語の機能を目的から探すことができます。統計が注目を集めています。初心者でも使えるように、導入から丁寧に解説しています。

シーアンドアール研究所
石田基広(著者)
800 ページ 價格: 4,860 円(PDF)