テクノロジ系

第 10 章　アルゴリズムとプログラミング

この章では，データ構造とアルゴリズムとプログラミングに関して基本的な考え方やさまざまな種類について学びます。基本的な考え方や種類について理解しましょう。

1　データ構造

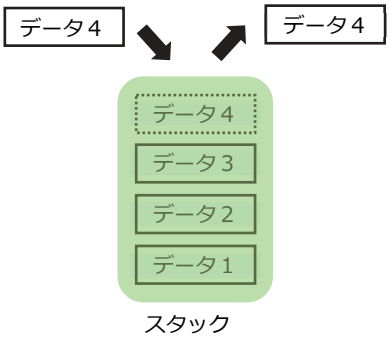
データ構造とは，データを効率的に扱うためのデータの格納や整理の方法です。プログラミングにおいて，データの検索，挿入，削除などの操作を高速に行うために用いられます。

⑴　リスト

順序を持ったデータの集合で，各データが前後のデータと関連付けられている構造です。データの追加や削除が柔軟に行えます。

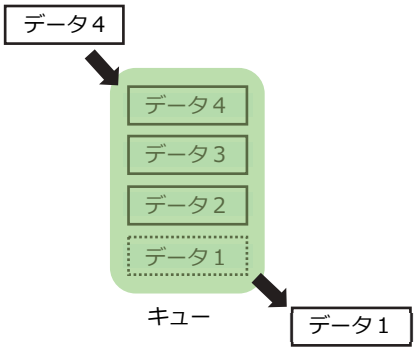
⑵　スタック

後入れ先出し（LIFO：Last In First Out）の原則に基づいたデータ構造で，追加したデータは逆の順番で取り出されます。イメージとしては本の積み上げに似ています。本を積み上げるときは，上へ上へと本を重ねていきます。一方で，積み上げられた本の中から本を取り出すときは，最上部にある本から順番に取り出します。



⑶　キュー

先入れ先出し（FIFO：First In First Out）の原則に基づいたデータ構造で，一方の端からデータを追加し，もう一方の端からデータを取り出します。列に並んだ人の中から最初に並んでいる人を先にサービスするイメージです。



⑷　木構造

木構造（ツリー構造）とは，ノードと呼ばれるデータの要素と，ノード間を繋ぐ枝で構成される階層型のデータ構造です。データの検索や管理に効率的です。

⑸　２分木

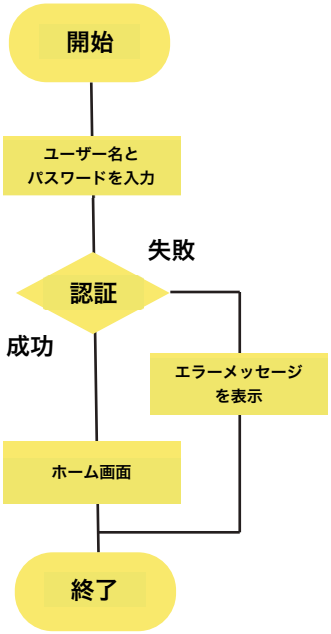
各ノードが最大２つの子ノードを持つことができる木構造で，データの整理や検索に用いられます。特にバランスの取れた２分木は検索効率が高いです。

2　アルゴリズム

アルゴリズムとは，問題を解決するための手順や処理の流れを明確に定義したものです。コンピュータにおけるタスクを効率的に実行するための基礎となります。

⑴　フローチャート

フローチャート（流れ図）とは，アルゴリズムやプログラムの処理手順，プロセスの流れを視覚的に表現するための図です。さまざまな形の記号と矢印を用いて，処理の順序や分岐点を示します。



⑵　アルゴリズムの基本構造

アルゴリズムの基本構造には「順次（シーケンシャル）」，「選択（セレクション）」，「繰返し（イテレーション）」があり，プログラムの流れを制御するための基本的な手法です。

順次　：命令を順番に実行していく構造
選択　：条件によって処理の流れを変える構造
繰返し：ある条件が満たされるまで，または満たされている間，同じ処理を繰り返し実行する構造

⑶　アルゴリズムの表現方法

①　式

値，変数，演算子を組み合わせたもので，計算のための規則を表します。例えば，x + y = z のように，具体的な計算を行います。

②　条件式

特定の条件を満たすかどうかを評価するための式です。例として，if（x > y）のように，条件が真か偽かを判断します。

③　演算子

数値や文字，論理値の計算を行うための記号です。例えば，+（加算）や = =（等価）などがあり，計算や論理操作を行います。

④　代入

変数に値を割り当てる操作のことです。x = 5 のように，特定の変数に値を設定します。この例では変数 x に 5 という数値を代入しています。

⑤　注釈

プログラムのコード内で説明を加えるための文です。実行時には無視されます。実行時には無視され，“// これは注釈です”のように書かれます。コメントともいいます。

⑥　入出力

データの入力と結果の出力を指す基本的な操作です。例えば，ユーザーからの入力を受け取るフォームや，計算結果を画面に表示することが含まれます。

⑦　手続

特定のタスクを実行するための一連のステップを定義したものです。例えば，料理のレシピのように，材料を準備し，手順に従って調理する順序立てた作業が含まれます。

⑧　関数

一定の処理を行い，結果を返す独立したコードのブロックです。例えば，ｆ（ｘ）＝ｘ＾２のように，特定の入力に対して出力します。

⑨　引数

関数に渡される値や変数のことです。関数ｆ（ｘ，ｙ）のｘやｙが該当し，処理に必要なデータを提供します。

⑩　戻り値

関数が処理の結果として返す値のことです。例えば，関数ｆ（ｘ）が計算結果として数値を返す場合です。

⑪　データ型

整数型，実数型，論理型，文字型など，プログラム内で扱うデータの種類を指します。これにより，データの格納方法や処理方法が決まります。

⑷　代表的なアルゴリズム

代表的なアルゴリズムには，探索，マージ，ソートなどがあります。これらのアルゴリズムは，データの効率的な処理や管理に不可欠です。以下の表にまとめました。

アルゴリズム	種類	説明
探索	線形探索法	・リストの先頭から順に要素を調べる最も基本的な探索方法
	２分探索法	・事前にソートされたリストを半分に分けながら目的の値を探す効率的な方法
整列	選択ソート	・リストから最小（または最大）の要素を選び，リストの先頭に移動させていく方法
	バブルソート	・隣接する要素を比較し，必要に応じて交換していく方法。大量のデータには適していない
	クイックソート	・ピボットを用いてリストを分割し，それぞれを再帰的にソートしていく方法 ・平均的な実行時間は非常に効率的である

3　プログラミング

プログラミングとは，プログラム言語を用いてアルゴリズムを記述し，コンピュータに特定の計算やタスクを実行させるための指示を出す行為です。これにより，意図した処理を自動で，かつ効率的にコンピュータが実行できるようになります。

⑴　プログラム言語の種類

プログラム言語にはさまざまな種類があり，それぞれ特定の目的や特徴を持っています。高級言語から低級言語，手続き型からオブジェクト指向型，特定のアプリケーション開発に特化した言語まで，使用する文脈や目的に応じて選択されます。

①　C

手続き型プログラミングをサポートする汎用のプログラム言語で，オペレーティングシステムや組み込みシステム開発に広く使用されています。

②　Fortran

数値計算や科学技術計算に強みを持つ高級プログラム言語で，長い歴史を持ちます。

③　Java

オブジェクト指向プログラミングをサポートし，書いたプログラムがさまざまなプラットフォームで実行できる「Write Once，Run Anywhere」を特徴とします。

④　C++

C 言語をベースにオブジェクト指向機能を追加したプログラム言語で，システム開発からアプリケーション開発まで幅広く使用されています。

⑤　Python

シンプルで読みやすい文法を持ち，初心者からプロフェッショナルまで幅広い分野で利用されるプログラム言語です。

⑥　JavaScript

Web ページの動的な動作を実現するために使用されるスクリプト言語で，フロントエンド開発に不可欠です。

⑦　R

統計計算やグラフィックスのためのプログラム言語で，データ分析や統計解析に広く利用されています。

⑵　コーディング基準やプログラム構造

コーディング基準を遵守することは，プログラムの品質向上と開発効率の向上に不可欠です。以下では，プログラムの構造とその可読性に関連する要素，および開発効率を高めるための外部リソースの利用に焦点を当てます。

①　プログラムの構造と可読性を向上させる要素

プログラムの構造と可読性を向上させる要素として，「字下げ」，「ネストの深さ」，「命名規則」があります。**字下げでは，プログラムの階層構造を視覚的に示すために空白やタブを用います。ネストの深さでは，プログラム内の条件分岐やループ構造がどれだけ深くなっているかを表します。**深すぎると可読性が低下するため，適切な管理が求められます。命名規則は，変数や関数にわかりやすい名前を付けるルールで，コードの理解と保守を容易にします。

②　開発効率を高めるための要素

開発効率を高めるためには，「モジュール分割」，「メインルーチン」，「サブルーチン」の概念が重要です。モジュール分割は，プログラムを機能ごとに独立した部分にすることで再利用性と保守性を向上させます。メインルーチンはプログラムの主要な処理を行い，サブルーチンは特定の機能を実現するために用いられ，ともにコードの構造を明確にします。

■　ライブラリと API の利用

「ライブラリ」，「API」，「Web API」の利用は，既存の機能を再利用し，新たな機能を容易に追加できるため，開発時間の短縮に貢献します。**ライブラリとは，特定の機能を提供するプログラムの集まりで，開発者が簡単に機能を組み込めるようにするものです。**API とは，異なるプログラム間で機能を共有するためのインターフェースで，ソフトウェアの連携を容易にします。Web API とは，Web サービスへのアクセスを提供する API で，インターネットを通じたデータのやり取りを可能にします。

■　ローコードとノーコードプラットフォームの利用

「ローコード」および「ノーコード」プラットフォームは，少ないコーディング知識でも，あるいはコーディング知識がなくてもアプリケーションを開発できます。ローコードは，少ないコーディングで迅速な開発を実現し，ノーコードは，コーディング知識がなくても使用できるため，ビジネスユーザーでも使えます。

⑶　擬似言語

擬似言語とは，実際のプログラム言語の文法にとらわれずに，アルゴリズムのロジックや処理の流れを人間が理解しやすい形で記述するための言語です。プログラムの設計やアルゴリズムの学習に有効です。

⑷　その他の言語

プログラミングの世界では，さまざまな言語が特定の目的で使用されます。ここでは，データを記述し，プログラム間で情報をやり取りするために用いられるマークアップ言語とその他の言語に焦点を当てます。

これらの言語を理解し，適切に利用することで，プログラムの機能を拡張し，異なるシステム間でのデータの共有や交換を効率的に行うことができます。

①　マークアップ言語

マークアップ言語とは，データの構造や意味を定義するためにタグや記号を用いる言語です。マークアップ言語には，HTML や XML などがあります。

■　HTML と XML

HTML と XML は，SGML という規格に基づいています。

HTML は Web ページの構造を定義し，XML はさまざまな種類のデータを表現するために使用されます。どちらもタグを使用して要素の開始と終了を示します。

②　JSON

JSON は，異なるプログラム言語間でデータをやり取りする際によく使用される軽量なデータ記述言語です。そのシンプルさと可読性の高さから，Web 開発を中心に幅広く採用されています。