

シラバスVer.6.0対応 追補レジュメ

IPAより公表されたシラバスの改訂に伴い、新たな用語例が追加されました。
ITパスポート試験シラバスVer.6.0で追加された主な用語例と、新しく追加された「擬似
言語」分野についてまとめました。

TAC

ストラテジ系 ◆企業と法務◆

テレワーク	テレワークとは、遠い(tele)と働く(work)を組み合わせた造語であり、IT を利用した、場所や時間の制約を受けない柔軟な勤務形態のことを指す。テレワークは、勤務場所により、 ・在宅勤務 … 自宅を就業場所とする ・モバイルワーク … モバイル機器を使用した、施設に依存せず仕事が可能状態のもの ・サテライトオフィス勤務 … サテライトオフィス、テレワークセンタ、スポットオフィスなど本来の勤務先以外のオフィスを就業場所とする に大きく分けられる。テレワークの導入によって、多様な働き方を実現できる、通勤が困難な優秀な人材を獲得できるなどのメリットがあるが、労務管理の困難さといった経営上の留意点もある。
官民データ活用推進基本法	国や地方公共団体、独立行政法人、民間事業者がもつ“官民データ”の流通・活用を推進するための法律。 「官民データ活用による、効果的かつ効率的な行政の推進」や「AIやIoT、クラウドコンピューティングなどの先端技術の活用促進」などを基本理念とし、「行政手続に係るオンライン利用の原則化」や「マイナンバーカードの普及及び活用の促進」などを基本的施策として挙げている。また、オープンデータ(国や地方公共団体が保有する公共データが、二次利用や機械判読に適した形態で、かつ原則無償で利用できる形で公開されること、及び公開されたデータ)についても定めている。
デジタル社会形成基本法	デジタル社会の形成に関する施策を迅速かつ重点的に推進し、経済の持続的かつ健全な発展と国民の幸福な生活の実現に寄与することを目的に制定された法律。「デジタル庁の設置」及び「デジタル社会の形成に関する重点計画の作成」などについて定めている。 なお、この法律における“デジタル社会”とは、AI や IoT、クラウドコンピューティングなどの先端技術をはじめとする情報通信技術を用いた情報の活用により、あらゆる分野における創造的かつ活力ある発展が可能となる社会を指す。
情報収集手法	代表的な情報収集の手法としては、 ・アンケート調査 … 定量的な情報の収集に向いている ・インタビュー … 定性的な情報の収集に向いている などがある。インタビューの手法は、さらに次のように分類することができる。 構造化インタビュー:あらかじめ決めておいた質問事項を、決められた順序で質問する 半構造化インタビュー:質問事項は決めておくが、回答の内容に対してインタビュアがさらに質問することで詳細な回答を引き出す 非構造化インタビュー:テーマのみを決めておき、回答者に自由に話してもらいながらインタビュアが必要に応じて質問していく その他にも、情報収集手法としては、実際に現地に行き様子を観察するフィールドワークがある。フィールドワークでは、必要に応じて現地でアンケート調査やインタビューを行うこともできる。
系統図	新 QC 七つ道具の一つ。問題解決や目的達成のための手段や方策を順次展開していき、最適な手段を追求していく図。

最小二乗法	回帰直線を求める際に用いられる手法。回帰直線と各データの差(距離)の二乗和をとり、その値が最小になるようなパラメタ設定を決定する。
マトリックス図	2次元の表を用いて、各要素の関連を表す図。これを用いたマトリックス図法は新QC七つ道具の一つ。
モザイク図	2次元の構成比を表す図。クロス集計表をグラフ化する際などに用いられる。帯グラフの内訳と横幅を使って、クロス集計表の行及び列の構成比を表す。
ロジックツリー	情報を整理し可視化する手法の一つ。事象を論理的に分析・検討する際に、樹形図によって論理展開を表現する図解技法。
コンセプトマップ	情報を整理し可視化する手法の一つ。アイディアや項目などをマップ上に並べ、線でつないで関連を示した図。概念地図などとも呼ばれる。
名義尺度／順序尺度／間隔尺度／比例尺度	質的データと量的データは、測定される尺度(ものさし)の違いでそれぞれ次のように分類できる。 ・質的データ 名義尺度: 性別や血液型など順番に意味がないもの 順序尺度: サービスの満足度(満足, 普通, 不満)など順番に意味があるもの ・量的データ 間隔尺度: 気温などデータの間隔に意味があるもの 比例尺度: 身長や速度など、原点(ゼロ)がありデータの比率に意味があるもの
クロスセクションデータ	時系列データが時間に従って収集したデータであるのに対して、時間のある時点に固定して複数の場所・空間の情報を横断的に集めたデータを、クロスセクションデータ(横断面データ)という。
仮説検定	統計学において、ある事柄についての仮説(効果がない, 差がないなど)が成り立つかどうかを、データから統計量を計算して調べる。単に“検定”と呼ぶこともある。
有意水準	統計学における検定において、仮説が成り立つか否か(ある結果が偶然に頼らず起こるかどう)かを判断する際の基準として用いられる値。
第1種の誤り／第2種の誤り	統計学における検定などで生じ得る誤りのこと。「仮説が正しいのに棄却してしまう」ことを“第1種の誤り”といい、「仮説が正しくないのに棄却されない」ことを“第2種の誤り”という。 製造業における抜き取り検査などでも用いられる考え方であり、「合格にすべきロットが誤って不合格になってしまう」ことを第1種の誤り(生産者危険)、「不合格にすべきロットが誤って合格になってしまう」ことを第2種の誤り(消費者危険)という。
シェープファイル	地理情報システム(GIS)で用いられる、図形情報や属性情報をもつ地理情報データが集まったファイル。“ <i>.shp</i> ”“ <i>.dbf</i> ”“ <i>.shx</i> ”などの拡張子をもつ複数のファイルから構成されている。
共起キーワード(共起語)	あるキーワードと共に使われることの多い言葉。テキストマイニングにおけるテキストデータの可視化や分析などで用いられる。
GIS(Geographical Information System: 地理情報システム)	地理情報をデジタルデータ化して管理・利用するシステムの総称。地理的位置を手がかりに、位置に関する情報を含むデータを管理・加工して視覚的に表示し、高度な分析や迅速な判断を可能にする。

モデル化	モデルとは、ものごとや現象の本質的な形状や法則性を抽象化して、より単純化したものであり、モデルを作ることをモデル化という。モデルは、対象の特性によって、次のように分類することができる。 静的モデル：プラスチックモデルや建築図面など、時間的要素を含まないモデル 動的モデル：レジの待ち行列や気象予測、生物の成長など、時間的要素を含んだモデル（動的モデルの中の分類として） 確定モデル：複利計算など、不規則な現象を含まず方程式などで表せるモデル 確率モデル：サイコロやくじ引きなど、不規則な現象を含んだモデル
ブレンライティング	意見やアイデアを集めるための発想法の一つ。議論をするのではなく、アイデアを書いた紙を回覧板のように順番に回していく。ブレンストーミングと同様に、批判厳禁で自由な発想からアイデアを出すことを奨励する。 具体的には、まず全員に一枚ずつ紙を配り、各人はその紙の1行目に自分のアイデアを書いて隣の人に渡す。次に、自分が渡された紙に書いてあるアイデアを読んで、連想するアイデアを次の行に書き、再び隣の人に渡す。これを、自分が最初に書いた紙が戻ってくるまで繰り返すことで、多数のアイデアを集めることができる。
限定提供データ	電磁的管理(ID・パスワード等)を施しつつ、相手方を限定して提供するデータ。不正競争防止法の平成30年改正によって、限定提供データを不正取得等する行為は不正競争行為として位置付けられることになった。 限定提供データの例としては、自動車メーカーが提供する車両の走行データ、POS で収集した消費動向データ、携帯電話の位置情報データを集計した人流データなどが挙げられる。
個人識別符号	特定の個人を識別できる文字や番号、記号の総称。認証用の指紋データや旅券番号(パスポートの番号)、個人番号(マイナンバー)などが該当する。
ISO 26000	社会的責任に関する国際規格。企業に限らずあらゆる組織を対象としている。
JIS Q 38500	IT ガバナンスに関する規格。国際規格である ISO/IEC 38500 に対応している。JIS Q 38500:2015 では、IT ガバナンスを「組織の IT の現在及び将来の利用を指示し、管理するシステム」と定義しており、良好な IT ガバナンスのための枠組みとして、経営者が“評価－指示－モニタ”という三つの職務によって IT を統制すべきであることなどが述べられている。
ストラテジ系 ◆経営戦略◆	
ITS(Intelligent Transport Systems: 高度道路交通システム)	人と道路と自動車の間で情報の受発信を行い、道路交通が抱える事故や渋滞、環境対策など、様々な課題を解決するためのシステム。
住民基本台帳ネットワークシステム(住基ネット)	各市町村の住民基本台帳データ(氏名、生年月日、性別、住所などが記載された住民票を編成したもの)を全国どこの市町村からでも照会できるようにしたシステム。
マイナンバーカード	本人確認の際の公的な本人確認書類として利用できる IC カード。カードの裏面に個人番号(マイナンバー)が記載されているため、税・社会保障・災害対策の法令で定められた手続きを行う際の番号確認や、様々な行政サービスを受けることができるようになる。

緊急速報	気象庁が配信する緊急地震速報や津波警報、特別警報、国・地方公共団体が配信する災害・避難情報などを、対象エリア内の対応端末(スマートフォンなど)に対して携帯電話会社から通知するサービス。エリアメールと呼ばれることもある。
J アラート(全国瞬時警報システム)	弾道ミサイル攻撃に関する情報や緊急地震速報、津波警報、気象警報などの緊急情報を、人工衛星及び地上回線を通じて全国の都道府県、市町村等に送信し、市町村防災行政無線等を自動起動することにより、人手を介さず瞬時に住民等に伝達するシステム。
AI 利活用ガイドライン	AI の利用者が AI を利活用するにあたり留意すべき事項を、10 の原則からなる「AI 利活用原則」としてとりまとめ、その実現のための具体的な措置についての解説を記載したガイドライン。特に、AI を利用して事業を行う者への解説書として、総務省の「AI ネットワーク社会推進会議」によってとりまとめられた。
スマート農業	IoT や AI といった先端技術を生かした農業。トラクターなどの農業機械の自動走行やアシストスーツによる重労働の軽労化、ドローンを使つての農薬散布などが期待されている。
テクノロジー系 ◆基礎理論◆	
グラフ理論	ネットワークや電気回路などを、グラフという数学モデルを用いて解析する情報科学の1分野。SNS などでの人のつながりや鉄道の路線などのような関係性も、グラフを用いて表現することがある。 グラフは「頂点(ノード、節)」と頂点同士のつながりを表す「辺(エッジ、枝)」から構成される。また、辺に一方通行の道路のような方向性が加えられたグラフを「有向グラフ」といい、方向性をもたないグラフを「無向グラフ」という。
P(ペタ)	補助単位(接頭語)の一つ。10 の 15 乗 = 1,000,000,000,000,000 倍を表す。
バックプロパゲーション(誤差逆伝播法)	ニューラルネットワーク全体の重みを調整する手法。調整作業は、出力層から入力層に向かって行われる。
Fortran	数式処理に適しており、主に科学技術計算に用いられるプログラム言語。
R(R 言語)	統計解析や機械学習の分野に適している、オープンソースのプログラム言語及び実行環境。データ分析、グラフ描画などのソフトウェアパッケージが多数提供されている。
コーディング標準	プログラミングにおいてプログラマが守るべきルールを、コーディング標準(コーディング規約)という。コーディング標準では、変数名などの命名規則やインデント(字下げ)の文字数、ネスト(入れ子)の深さの許容数などが決められている。 コーディング標準を守ることで、そのプログラムの可読性が高くなり、プログラム作成者以外の開発メンバが見てもプログラムを理解しやすくなるため、ソースコードの変更・修正といった保守における作業効率の向上やコストの削減が期待できる。
メインルーチン/サブルーチン	プログラムが開始された際に最初に呼び出され、プログラムの中核となる処理群(関数など)をメインルーチンと呼ぶ。これに対して、他の処理から呼び出される処理群を、サブルーチンと呼ぶ。一般的なプログラムは、メインルーチンで全体を制御しつつサブルーチンを呼び出すことで処理を進める構造になっていることが多い。

ライブラリ	汎用的な機能や関数を他のプログラムから利用できる形にまとめたファイル群。ライブラリには、OS やプログラミング言語の開発元から提供されるもの(システムライブラリ)や、一般のユーザが開発して使用するもの(共用ライブラリ)がある。
API(Application Program Interface)	ソフトウェアの機能を別のソフトウェアから呼び出して利用するためのインタフェース(関数や手続の集まり)。
WebAPI	HTTP など、Web の技術を用いて構築された API。WebAPI によって、ネットワーク上に分散したアプリケーションを連携させることができる。送受信されるデータの形式は様々だが、一般に XML、JSON などが用いられることが多い。
ノーコード／ローコード	ソースコードを全く書かずに開発する手法を、ノーコードという。ノーコードでは、GUI を利用した専用ツールのテンプレートや機能部品を使って開発する。これに対して、通常のプログラミングよりも圧倒的に少ないソースコードで開発する手法を、ローコードという。いずれも、開発期間短縮や開発コスト削減、業務効率化などの導入効果が期待される。 なお、ノーコードやローコードに対して、エンジニアがプログラム言語を用いて通常のプログラミングを行う開発(スクラッチ開発)を、プロコードと呼ぶこともある。
JSON(JavaScript Object Notation)	異なるプログラム言語で書かれたプログラム間でのデータのやり取りなどに用いられる、データ記述仕様の一つで、WebAPI におけるデータの送受信などで用いられている。JSON は、JavaScript の言語仕様として規定されたデータ形式であり、リスト構造などを簡潔に記述できる。

テクノロジー系 ◆コンピュータシステム◆

DDR3 SDRAM／DDR4 SDRAM	DRAM の一種。動作クロックに同期してデータを転送することで高速化を実現した DRAM を、SDRAM(Synchronous DRAM)といい、それをさらに1クロックで2回データ転送を行えるよう改良したメモリが DDR-SDRAM である。DDR-SDRAM は、世代を重ねるごとに転送速度などが改良されており、“DDR3”や“DDR4”などの数字は DDR-SDRAM の世代を表している。なお、2020 年には“DDR5”の最終仕様が発表された。
DIMM／SO-DIMM	DIMM(Dual Inline Memory Module、ディム)とは、DRAM チップを搭載したメモリ基板(メモリモジュール)、及びそのピン配置などを規定した規格のことである。マザーボードにある専用の差込口(DIMM スロット、メモリソケット)に差し込んで使用する。 一方、SO-DIMM(Small Outline DIMM)は、DIMM をノート PC などを使用するために小型化した規格である。いずれも、搭載される DDR-SDRAM チップの種類ごとに多数の規格が存在する。
煙センサ	煙を検知するセンサ。煙感知器などに利用されている。
Chrome OS	Google 社が開発した OS。Linux をベースにしている。

アクチュエータ(DC モータ, 油圧シリンダ, 空気圧シリンダ)	アクチュエータとは、入力されたエネルギーやコンピュータから出力された電気信号を、機械的な動作へ変換する仕組みや駆動装置の総称である。IoT システムにおいては、センサ類が収集した情報がクラウド上で分析され、その判断結果を受け取ったアクチュエータが物理的に作動する。 アクチュエータの例としては、次のようなものがある。 DC モーター: 直流によって車輪のような回転運動をする。ドローンなどで使用され、プロペラの回転数を変えることで姿勢を制御する。 油圧シリンダ: 油圧によりシリンダ内のピストンを動かす。 空気圧シリンダ: 空気を圧縮膨張させシリンダ内のピストンを動かし、ドアの開閉などを行う。
テクノロジー系 ◆技術要素◆	
デザインの原則(近接, 整列, 反復, 対比)	画面内に画像や文章を配置する際には、目の特性を考慮すると効果的である。目の特性を考慮したデザインの原則として、次の4つがよく知られている。 ・近接 … 人間の目は、距離が近いものをまとまりとして捉えるので、同じまとまりの情報は近づけ、違う要素は離す ・整列 … 人間の目は、要素が揃っていることで安定するので、右揃え、左揃え、中央揃えなど、要素を揃えることを意識する ・反復 … 人間の目は、同じ要素が繰り返されることで統一感を感じるので、見出しやフォーマットを揃える ・対比／強弱 … 人間の目は、要素ごとの差分を取って違いを認識するので、要素ごとの大小や強弱を明確にする。見出しは本文よりも太くする
シグニファイア	利用者を特定の行動に誘導する役割をもたせたデザイン。例えば、Web サイト上に配置したボタンのアイコンに影を付けて立体感を出すことによって“クリックする”という行動を誘導したり、検索用のテキストボックスの真横に虫眼鏡アイコンの検索ボタンをつけることによって“キーワードを入力して検索する”という行動を誘導したりする効果が期待できる。
構造化シナリオ法	デザインしたものが実際に利用される場面を具体的に想定して、段階的に要件を定義していくシナリオ手法。主に、UX デザインを実現するためのプロセスとして用いられる。 構造化シナリオ法では、「①バリューシナリオ(ユーザにもたらす価値) → ②アクティビティシナリオ(ユーザの行動) → ③インタラクションシナリオ(ユーザがする具体的な操作)」の3段階でシナリオを考えていく。
インフォグラフィックス	インフォメーションとグラフィックスを組み合わせた造語。情報やデータを、視覚的にわかりやすくなるようにグラフや図、イラストなどを利用して一つにまとめて表現したものである。例えば、電車の路線図などが該当する。
人間中心設計	ユーザビリティの向上を目的として、利用者の立場に立った(利用者のニーズを満たす)デザインをするという考え方。その設計プロセスは、ISO 13407 及び JIS Z 8530(人間工学—一人とシステムとのインタラクション—インタラクティブシステムの人間中心設計)として規格化されている。
WAV	音声データ用のコンテナフォーマット(複数種類のファイルをまとめて格納するファイル形式)。一般的には、非圧縮形式の音声データファイルに用いられることが多い。

ラスタデータ(ビットマップデータ)	画像を点(ピクセル)の集まりとして表現したデータ。色の種類や明るさを、ピクセルごとに調節することが可能。図形を拡大すると輪郭にジャギー(ギザギザ)が生じるといった欠点がある。
ベクタデータ	画像を、点とそれを結ぶ線や面の方程式のパラメータ、及び塗りつぶしなどの描画情報によって表現したデータ。ベクタデータで描かれた画像や字体は、画像を拡大・縮小したり変形したりしても輪郭の計算処理などがその都度行われ、解像度に見合った画質が維持される。
BMP	主にWindowsで用いられる静止画像フォーマット。無圧縮なため、データ容量が大きくなる。
TIFF	タグを用いることにより複数の画像データを一つのファイルに格納できる静止画像フォーマット。無圧縮、または可逆圧縮方式で保存できるため高解像度で再生や印刷できるが、Web ページで扱えない、データサイズが大きいといった欠点もある。
EPS	PostScript(ページ記述言語)で記述されたデータ(文字や図形、画像などをレイアウトしたページ)を、画像ファイルとして記録するファイルフォーマット。ラスタデータとベクタデータの両方を含めることができる。
フレーム/フレームレート	動画において、1枚の画像が記録される区画の単位をフレーム(コマ)という。また、1秒間の動画を構成するためのフレーム数(コマ数)をフレームレートといい、単位として fps(frames per second)を用いる。フレームレートが大きいほど動きがなめらかな映像となるが、その分データ量が増える。
H.264/MPEG-4 AVC	ディジタルハイビジョン放送やワンセグ放送などに用いられる動画データの圧縮符号化方式。ITU-T では、H.264 として勧告されている。また、ISO/IEC では MPEG4 の一部として規定されており、MPEG-4 AVC などと呼ばれる。 MPEG2 と比べて圧縮率が高く、低ビットレート(低速低画質)から高ビットレートまでの様々な用途に用いられる。
H.265	H.264/MPEG-4 AVC の後継となる動画データの圧縮符号化方式。H.264 よりさらに圧縮率が高い。
AVI	主にWindowsで用いられるコンテナフォーマット。様々な圧縮データを格納できるが、ストリーミング再生に対応できないという欠点をもつ。ファイルの拡張子には“.avi”が付けられる。動画データでは MPEG-4 や H.264 など、音声データでは MP3 や AAC などが格納できる。
MP4	コンテナフォーマットの一つ。映像や音声の他に静止画や字幕なども格納できる。ファイルの拡張子には“.mp4”が推奨されている。 動画データでは MPEG-4 や H.264、H.265 など、音声データでは MP3 や AAC など、静止画データでは JPEG などが格納できる。
ランレングス法	データ中の同一の情報が連続している部分に着目したデータ圧縮法。例えば“BBBBBB”のような連続する圧縮対象文字の箇所を、「特殊文字」「対象文字」「連続数」からなる3文字“\$B6”のように置き換えることで、データを圧縮する。白黒2値画像などの圧縮に用いる。
ハフマン法	各データの出現頻度に応じてビット数の異なる符号を割り当てるデータ圧縮法。文字列の圧縮では、出現確率が高い文字にビット数が少ない符号を、出現確率の低い文字にビット数が多い符号を割り当てることによって、データ全体を圧縮する。

加法混色	各色を混ぜ合わせるほど明るくなり白に近づく現象。ディスプレイ画面などで光の三原色(RGB)を用いて色を表現する方法は、加法混色である。
減法混色	各色を混ぜ合わせるほど暗くなり黒に近づく現象。カラープリンタなどで色の三原色(CMY)を用いて色を表現する方法は、減法混色である。
ppi(pixels per inch)	1インチ当たりのピクセル数(画素数)。ディスプレイなどの解像度を表す単位として用いられる。画素密度ともよばれる。
ペイント系(ラスタ形式)ソフトウェア	マウスやペンなどを用いて画面上にフリーハンドで絵を描くように図を作成し、点(ピクセル)の集まりとしてラスタ形式で保存するグラフィックスソフトウェア。Adobe Photoshop などが該当する。
ドロー系(ベクタ形式)ソフトウェア	イラストなどの図を、点や線、図形と塗りつぶしなどを組み合わせて作成し、座標や計算式に線の色などの情報も加えたベクタ形式で保存するグラフィックスソフトウェア。Adobe Illustrator などが該当する。
キーバリューストア(KVS)	NoSQL の一つ。データを一意に識別できるキー(Key)と、任意の保存したいデータ値(Value)の組合せを管理単位として保存するデータベース。構造が単純なため、高速、スケーラビリティ(拡張性)が高い、複数のサーバに分散して配置できるといった特徴をもつ。
ドキュメント指向データベース	NoSQL の一つ。データを“ドキュメント”として扱うデータベース。個々のドキュメントのデータ構造は自由で、JSON 形式や XML 形式のデータを保存することが可能。ドキュメントに対してインデックスを作成することもできる。
グラフ指向データベース	NoSQL の一つ。データをグラフ構造で関連させたデータベース。「ノード(節)、リレーション(エッジ)、プロパティ(属性)」の3要素でノード間の関係性を表現する。データ構造がネットワーク状となっている場合に、データ間の関係を検索する処理が速いという特徴をもつ。
バックワードリカバリ	ロールバックと同じ。
フォワードリカバリ	ロールフォワードと同じ。
VLAN(Virtual LAN)	物理的な接続形態に依存せず、論理的(仮想的)な LAN を構築する技術。例えば、VLAN を実現する方法の一つであるポートベース VLAN では、スイッチングハブのポート 1 とポート 2 で一つの LAN とし、ポート 3 とポート 4 で別の LAN とすることで、同じスイッチングハブ上に異なる LAN を構築することができる。
WPS(Wi-Fi Protected Setup)	Wi-Fi アライアンスが策定した、無線 LAN の接続やセキュリティの設定を簡単に行うための機能。親機と子機の双方で WPS ボタンを押すプッシュボタン方式や、親機の生成した数桁の数字(PIN)を子機側で入力する PIN コード方式などがある。

OSI 基本参照モデル	プロトコルの機能を階層化して考える方法(モデル)の一つに、ISO によって定められた OSI 基本参照モデルがある。このモデルでは各機能を7層に分けており、下位層(数値の小さい層)ほどハードウェアに近い立場での機能を実現するようになっている。 第7層:アプリケーション層: ファイル転送や電子メールなど、アプリケーションが実現すべき機能を提供する 第6層:プレゼンテーション層: 各アプリケーションが送受するデータを、実際に伝送するためのデータに変換する 第5層:セッション層: アプリケーション間で取り交わす会話を構成する 第4層:トランスポート層: ネットワークごとの品質の差を吸収した透過的な(下位層の違いを意識させない)伝送路を上位層に提供する 第3層:ネットワーク層: データの中継や経路選択を行い、送信元から宛先までのエンドシステム間(エンドツーエンド)のデータ伝送を実現する 第2層:データリンク層: 隣接するノード間(データリンク)における確実なデータ伝送を実現する 第1層:物理層: 電圧やコネクタの形状など、伝送を行う上での物理的・電氣的な特性を規定する
TCP/IP 階層モデル	OSI 基本参照モデルは機能を分割したモデルの一つであり、実際のプロトコルを規定したものではない。実際のプロトコル群としては、インターネットなどで標準的に用いられている、4層からなる TCP/IP 階層モデルが代表的である。単に TCP/IP とも呼ばれる。 第4層:アプリケーション層: HTTP, SMTP などの各種アプリケーションプロトコル。OSI 基本参照モデルの第5～7層に対応する 第3層:トランスポート層: TCP, UDP など。OSI 基本参照モデルのトランスポート層に対応する 第2層:インターネット層: IP など。OSI 基本参照モデルのネットワーク層に対応する 第1層:データリンク層: 各種 LAN など。OSI 基本参照モデルの第1, 2層に対応する
ファイルレスマルウェア	実行ファイルが PC の磁気ディスクなどに保存されることなく、メモリ上で攻撃を実行するマルウェア。ディスク上に実行ファイルがないため、ウイルス対策ソフトのパターンマッチングなどでは検出できない。また、PC の電源を切るとメモリ内のデータが消えるため、その痕跡が残らずステルス性が高いという特徴をもつ。

アルゴリズムとデータ構造

ITパスポート試験シラバスVer.6.0では、高等学校学習指導要領「情報Ⅰ」に基づいて、プログラミング的思考力を問う擬似言語を用いた出題が追加されます。

アルゴリズムとプログラム

アルゴリズムは「解法」を表す言葉であり、**特定の目的を果たすための方法や手順**のことをいいます。アルゴリズムをコンピュータが実行できる形式で具体的に表現したものが**プログラム**です。

プログラムは「数値10を変数aに代入する」などの具体的な命令(処理)の集合で成り立っており、表現手段には、**流れ図(フローチャート)**や**擬似言語**、JavaやPythonなどの**プログラム言語**があります。擬似言語とは、情報処理技術者試験などで用いられる簡易なプログラム言語です。

変数と代入

プログラムの中では、**変数**と呼ばれる「データの入れ物」を用意し、そこへ具体的な値を**代入(格納)**して使用します。例えば“a”という名前の変数を用意し、そこへ10という数値を代入する場合、

$a \leftarrow 10$

のように記述します。

変数には、他の変数の値を代入することもできます。例えば、変数aの内容を変数bに代入するときには、

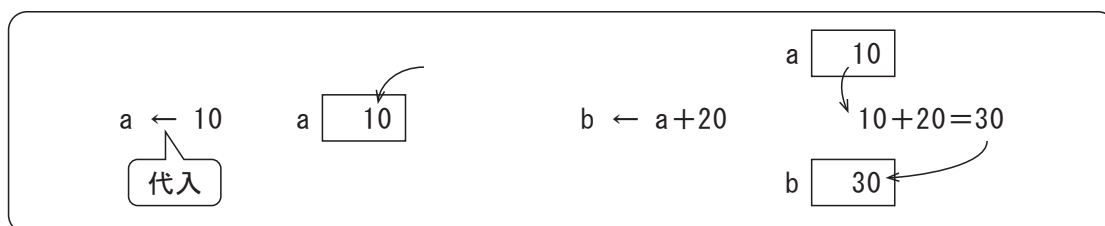
$b \leftarrow a$

のように記述します。このとき、変数aの内容が変数bに**コピーされる**と考えましょう。移動するわけではないので、変数aの内容は変わりません(元の値のまま)。

また、

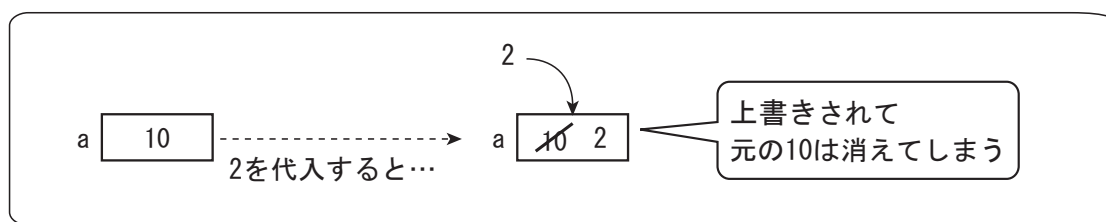
$b \leftarrow a + 20$ (変数aの値に20を加算して、変数bに代入)

のように変数を演算に用いることもできます。演算では、その時点で変数aに格納されている値が使用されます。



変数と代入

なお、変数には同時に一つの値しか格納できません。例えば、あらかじめ変数aに10が格納されている場合に、その変数aに2を代入すると、変数aの内容は10から2に**上書き(更新)**されます。



変数の上書き

データ型

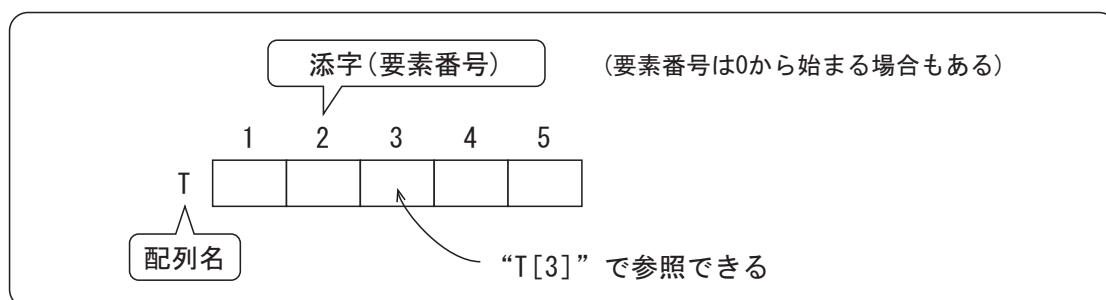
プログラムで取り扱うデータにはいくつかの種類(**データ型**)があります。主なものには、整数型、実数(小数)型、文字型、論理型などがあります。

主なデータ型

整数型	123や-987などの整数を値としてとる。
実数型	2.345や0.01などの実数を値としてとる。
文字型	'A'などの文字を値としてとる。"ABCDE"のような複数の文字の並びをまとめて扱う、 文字列型 を定義することもある。
論理型	true(真) か false(偽) の2値をとる。条件判定などに用いられる。

配列

配列は、**同じデータ型をもつ複数の要素の集まり**です。配列の名前と「その配列内で何番目か」を表す**添字(要素番号)**を指定することで、各要素を識別できます。



配列

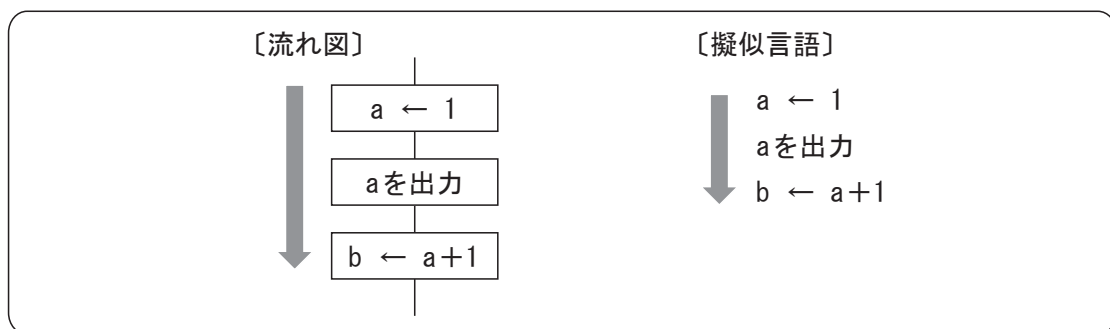
“T[i]”のように、添字として変数を使うこともできます。この場合、その時点で変数iに格納されている値が適用されます。

制御構造

アルゴリズムの基本構造は、「順次、選択、繰返し」という構造の組合せで構成されています。このような処理の流れを**制御構造**といいます。この基本的な制御構造を、流れ図と擬似言語で表した例も含めて紹介します。

①順次

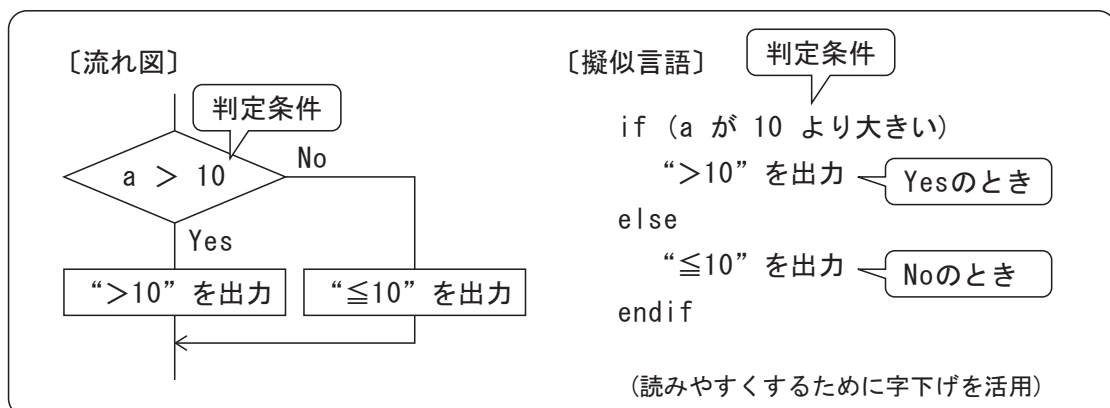
各命令を、**記述した順番に逐次実行していく**流れです。



順次構造

②選択(分岐)

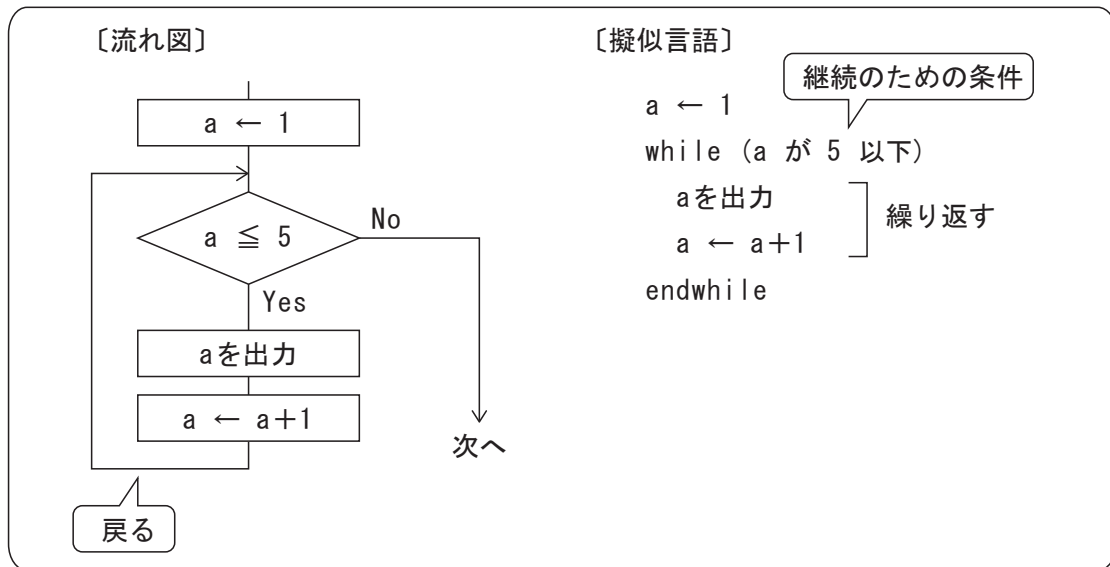
ある**条件**についてYes(真, true)かNo(偽, false)のどちらになるか評価し、それぞれ**個別に用意された処理へ分岐**します。



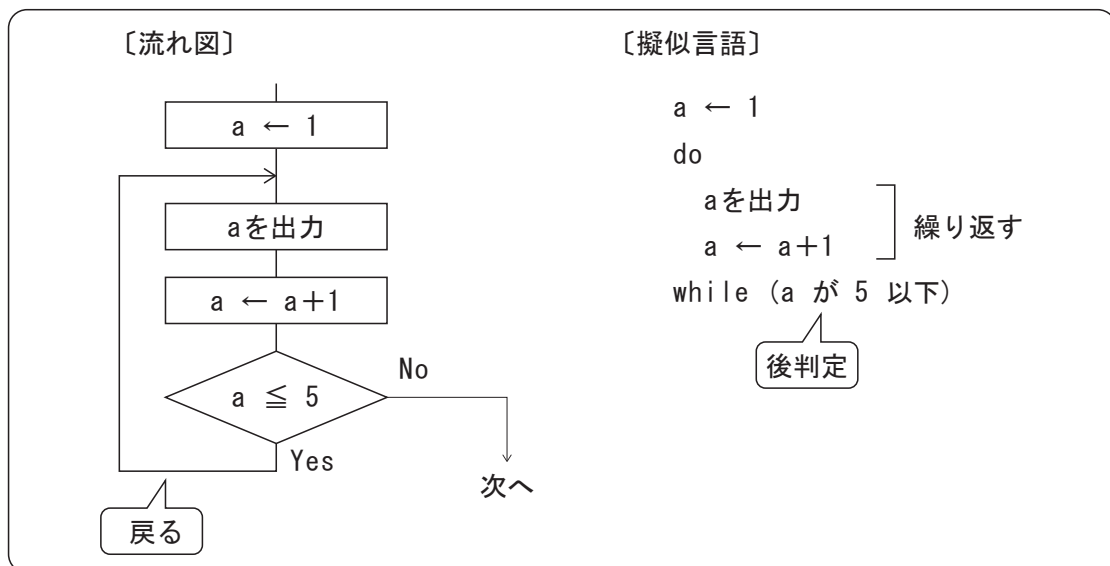
選択構造

③ 繰返し(ループ, 反復)

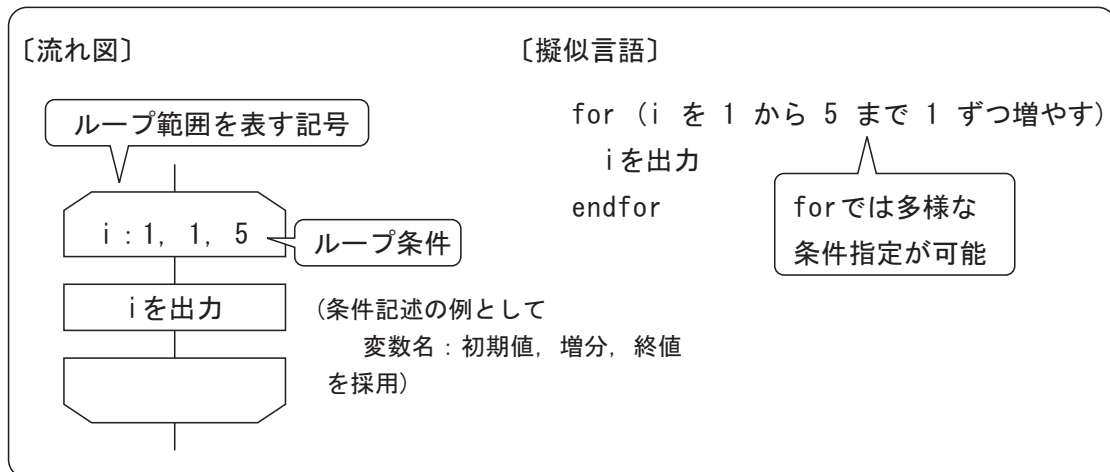
指定した条件に沿って, **特定の処理を繰返し実行**します。



繰返し構造1(whileによる前判定繰返し)



繰返し構造2(do ~ whileによる後判定繰返し)



繰返し構造3(forによる繰返し)

「前判定」繰返しと「後判定」繰返しの違いは、

前判定：条件式の評価をしてから繰返し処理を行うので、

繰返し処理を「**1回も行わないことがある**」

後判定：条件式の評価をするのは繰返し処理を行った後なので、

繰返し処理を「**少なくとも1回**は行う」

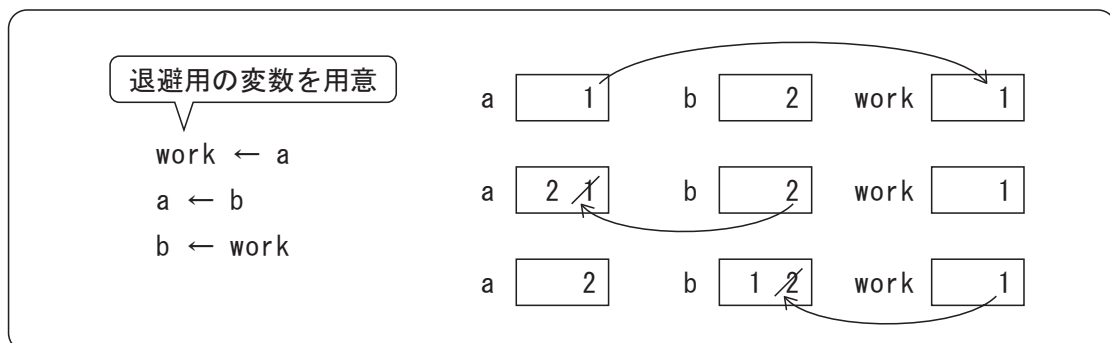
という点です。

よく用いられるアルゴリズム

汎用性の高い簡単なアルゴリズムをいくつか紹介します。

①変数間の交換

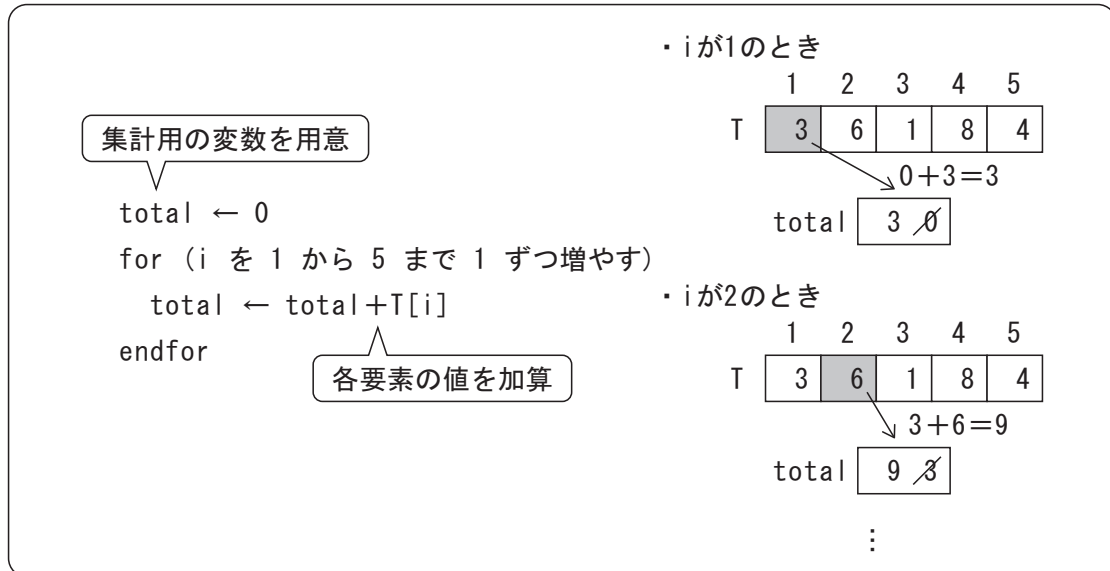
変数aとbの内容を**交換**したい場合は、**もう一つ別に変数を用意**して、三つの代入によって交換を実現します。



変数間の交換

②配列の集計

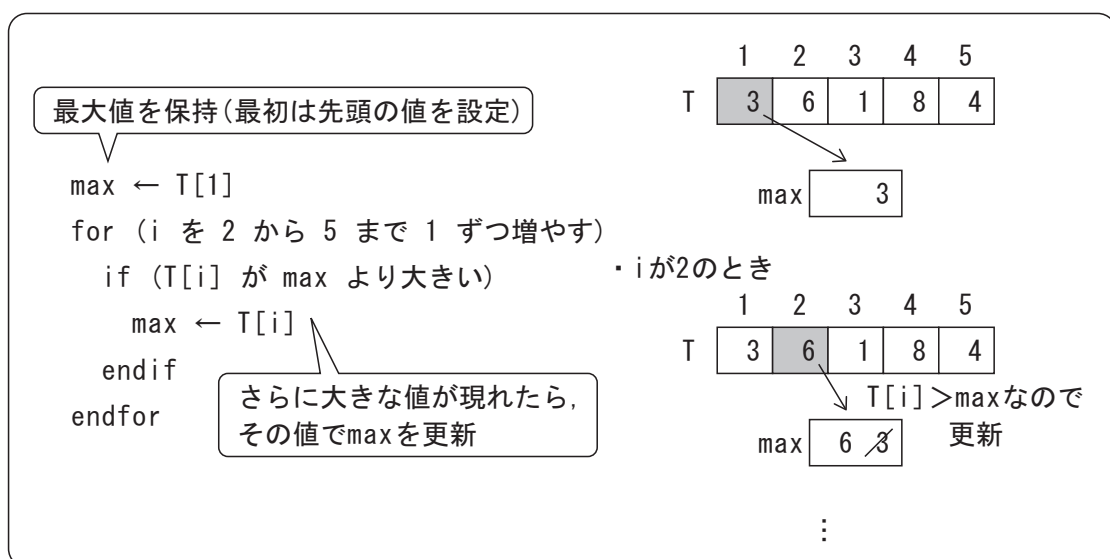
繰返し(ループ)を用いて変数を変化させ、それを配列の添字に用いることで各要素を順番に調べていくことができます。**集計処理**もその応用の一つです。



配列の集計

③配列中の最大値を求める

②と同様に配列の内容を順番に見ていきながら、**仮の最大値**(その時点での、最大の値)を**常に変数に保存**しておきます。最後まで見終わったときに残った値が、全体の中での最大値です。



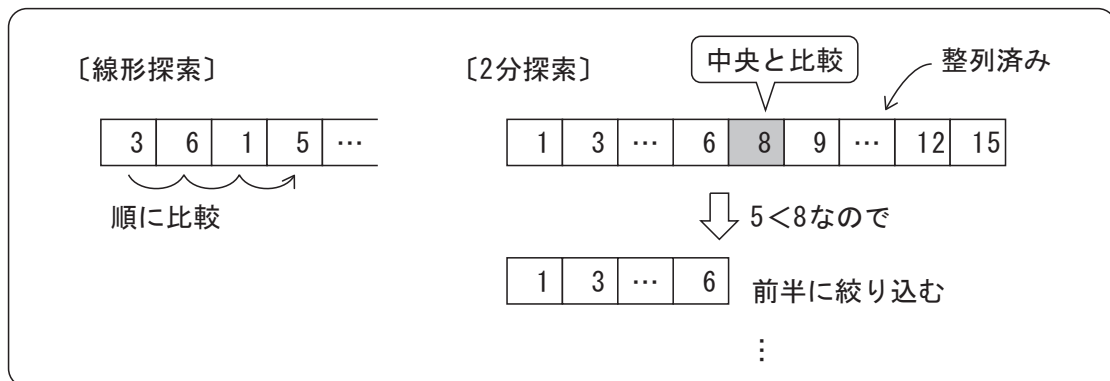
最大値を求める

探索アルゴリズム

多数のデータ群の中から特定の値を探し出す探索アルゴリズムには、次のようなものがあります。一般的には、2分探索の方が比較する回数が少なく済みます。

線形探索：先頭から順番に一つずつ探索値と比較していく

2分探索：中央の値と比較し、次の探索範囲は前半か後半のどちらかに絞り込む
(データ群が整列されていることが前提条件)



探索アルゴリズム(5を探索する例)

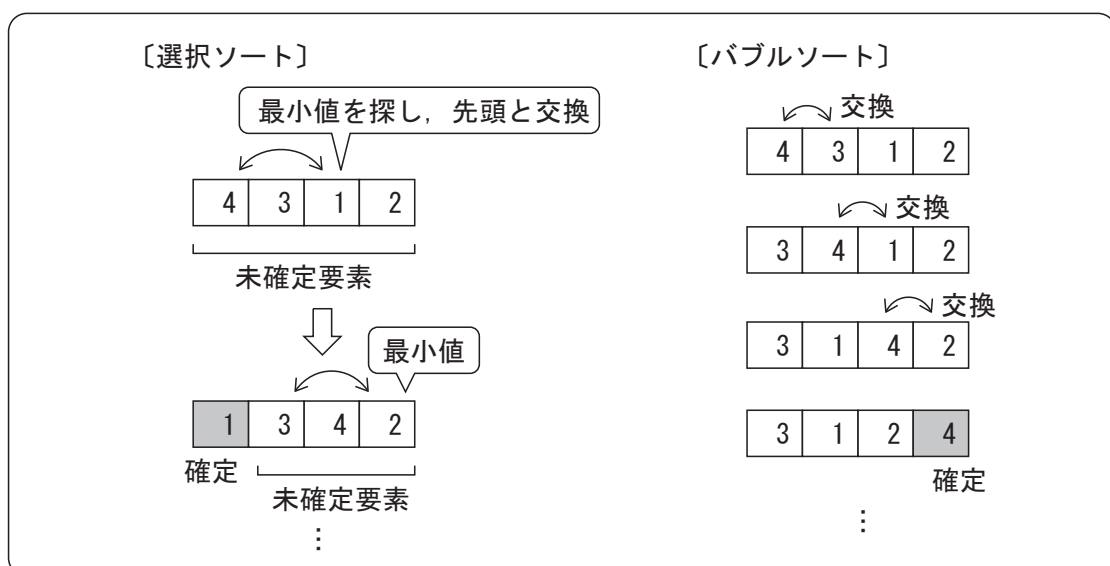
整列アルゴリズム

多数のデータ群を、昇順(小さい順)や降順(大きい順)に並べ替えるアルゴリズムです。沢山の**整列アルゴリズム**が存在しますが、ここでは代表的なものを紹介します。

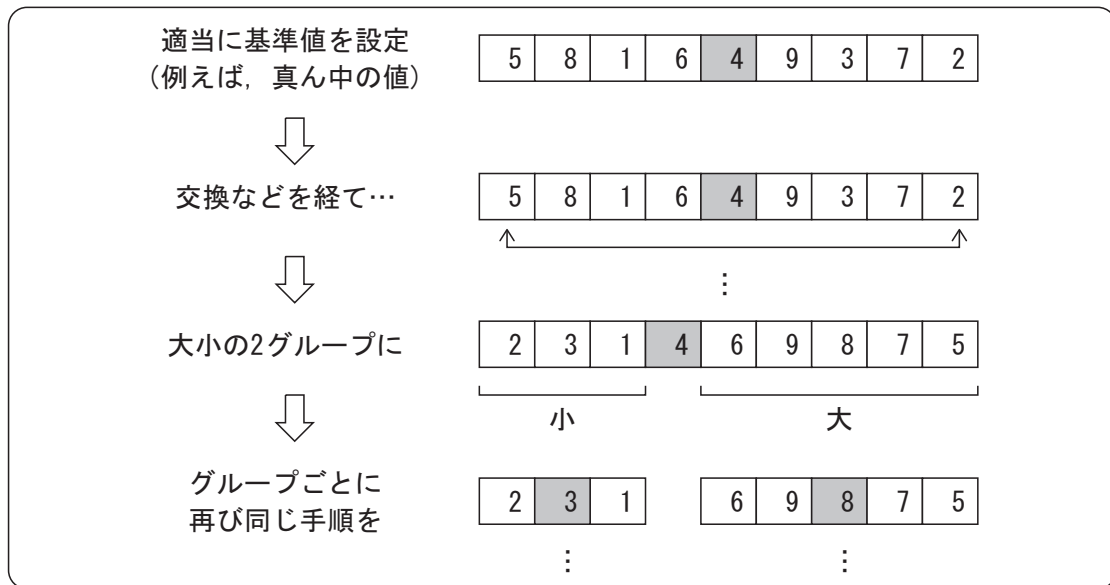
選択ソート：「最小値(最大値)を選んで並べていく」処理を繰り返す

バブルソート(交換ソート)：「隣同士を比べ、逆順なら入れ替える」処理を繰り返す

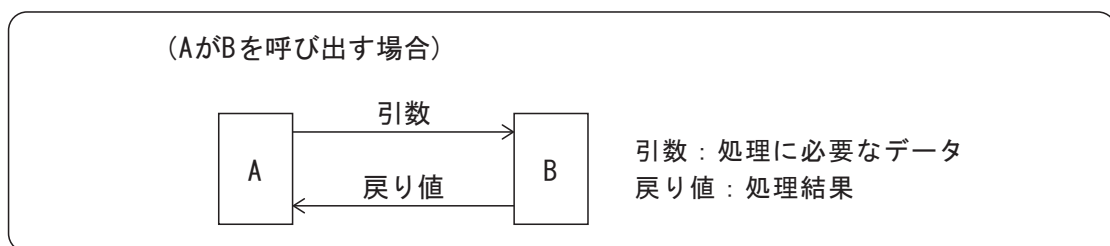
クイックソート：「基準値を選び、それを基に大小2グループに分ける」処理を繰り返す



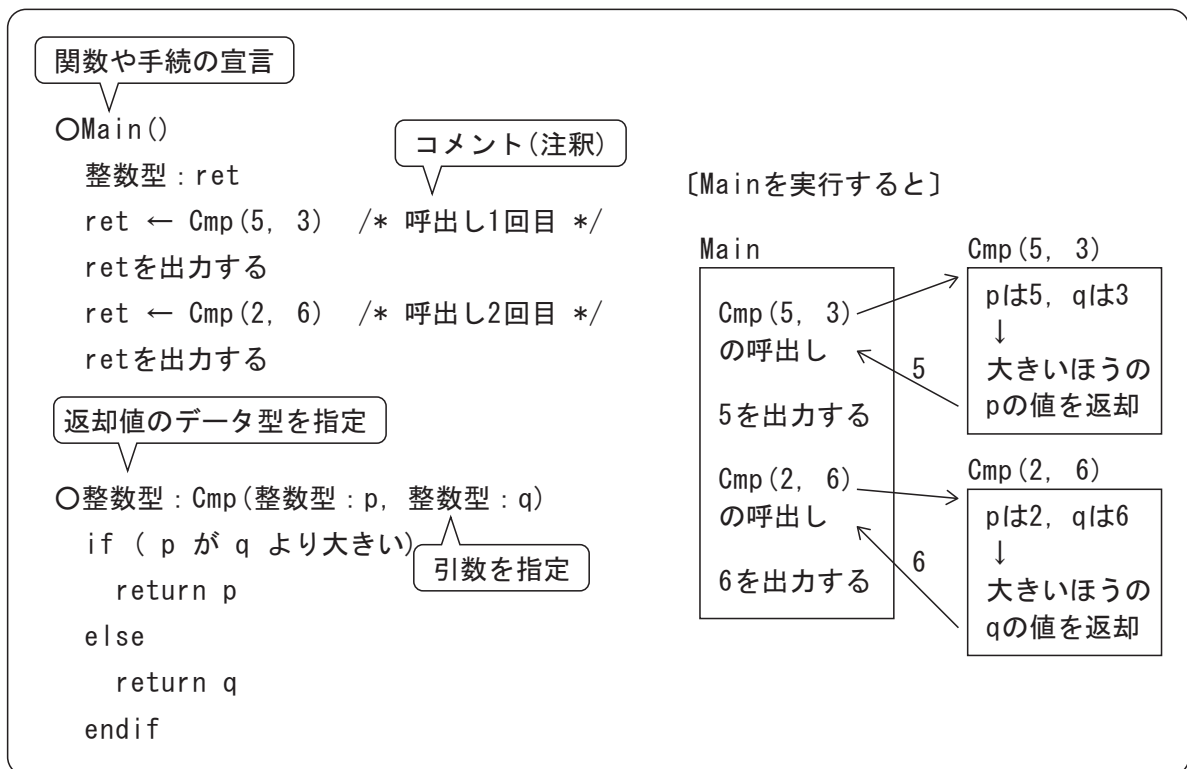
選択ソートとバブルソートの例



関数



二つの整数の比較を何回か行うプログラム例を示します。比較自体は関数Cmpが行い、大きな方を返却します。手続Mainは関数Cmpを呼び出して、返却された値を出力(表示)します。



関数利用の例

● 関連用語

メインルーチン／サブルーチン

プログラムが開始された際に最初に呼び出され、プログラムの中核となる処理群(関数など)をメインルーチンと呼ぶ。これに対して、他の処理から呼び出される処理群を、サブルーチンと呼ぶ。一般的なプログラムは、メインルーチンで全体を制御しつつサブルーチンを呼び出すことで処理を進める構造になっていることが多い。

以上のプログラムで用いた疑似言語の仕様は、IPAのITパスポート試験に関するWebページ“試験内容・出題範囲”の中で

「試験で使用する情報技術に関する用語・プログラム言語など」
として紹介されています。2022年3月現在では、2021年10月8日に掲載されたVer.4.3

https://www.jitec.ipa.go.jp/1_13download/ip_programming_sample.pdf
が最新のバージョンになっています。

サンプル問題

IPA が「IT パスポート試験 疑似言語のサンプル問題」として公開している問題を紹介します。

【問 1】

関数 calcMean は、要素数が 1 以上の配列 dataArray を引数として受け取り、要素の値の平均を戻り値として返す。プログラム中の a, b に入れる字句の適切な組合せはどれか。ここで、配列の要素番号は 1 から始まる。

〔プログラム〕

```
○実数型: calcMean(実数型の配列: dataArray) /* 関数の宣言 */  
  実数型: sum, mean  
  整数型: i  
  sum ← 0  
  for (i を 1 から dataArray の要素数 まで 1 ずつ増やす)  
    sum ← a  
  endfor  
  mean ← sum ÷ b /* 実数として計算する */  
  return mean
```

	a	b
ア	sum + dataArray[i]	dataArray の要素数
イ	sum + dataArray[i]	(dataArray の要素数 + 1)
ウ	sum × dataArray[i]	dataArray の要素数
エ	sum × dataArray[i]	(dataArray の要素数 + 1)

【問2】

手続 printStars は、“☆”と“★”を交互に、引数 num で指定された数だけ出力する。プログラム中の a, b に入れる字句の適切な組合せはどれか。ここで、引数 num の値が 0 以下のときは、何も出力しない。

〔プログラム〕

```
○printStars(整数型: num)      /* 手続の宣言 */  
  整数型: cnt ← 0              /* 出力した数を初期化する */  
  文字列型: starColor ← "SC1" /* 最初は“☆”を出力させる */
```

a

```
  if (starColor が "SC1" と等しい)
```

```
    "☆"を出力する
```

```
    starColor ← "SC2"
```

```
  else
```

```
    "★"を出力する
```

```
    starColor ← "SC1"
```

```
  endif
```

```
  cnt ← cnt + 1
```

b

	a	b
ア	do	while (cnt が num 以下)
イ	do	while (cnt が num より小さい)
ウ	while (cnt が num 以下)	endwhile
エ	while (cnt が num より小さい)	endwhile

【解答解説】

問1 ア

引数として受け取った配列dataArrayについて、要素の値の平均を求めるには、

(1) dataArrayの各要素の値の合計を計算する。

(2) (1)で求めた合計を、「dataArrayの要素数」で割って平均を計算する。

という手順で処理を行えばよい。本問のプログラムを見ると、空欄bを含む処理で、

```
mean ← sum ÷ b
```

のように、「変数sumの値を空欄bの式で割った値」を変数meanに代入していることから、この処理は「meanに、前述(2)の平均を求める」ための処理であり、その前のfor文などの処理が「sumに、前述(1)の合計を求める」ための処理であると推測できる。

aについて

配列dataArrayの各要素の合計を求めるための処理中の空欄である。このような場合、

「何らかの箱(変数)を用意して、その変数に

配列dataArrayの要素を一つずつ足し込んでいく」

というアルゴリズムがよく使われる。本問のプログラムでは、そのための箱が変数sumであり、forを用いた繰返しが足し込んでいく処理に該当する。

繰返し処理では、整数型の変数iを、

1, 2, 3, ..., dataArrayの要素数

のように1ずつ増やしながら、

```
sum ← a
```

を行っている。この処理は配列dataArrayの各要素の「合計」を求めるための処理なので、空欄aには、

```
sum + dataArray[i]
```

を入れればよい。

こうすることによって、forを用いた繰返し処理で、

```
sum ← sum + dataArray[1]    … 繰返し1回目
```

```
sum ← sum + dataArray[2]    … 繰返し2回目
```

…

```
sum ← sum + dataArray[dataArrayの要素数]
```

という処理が行われる。このように、「今までのsum + dataArrayの要素の一つ」の結果を「新たなsum」とする処理を、dataArrayの全要素に対して1回ずつ行うことによって、変数sumに配列dataArrayの要素の合計を求めることができる。

bについて

前述(2)で述べた平均を求めるための処理中の空欄である。dataArrayの各要素の合計はsumに求められているので、平均を求めるにはsumの値を、

dataArrayの要素数

で割ればよい。

問2 エ

二つの空欄は、いずれも繰返し処理の制御に関する部分である。解答群を見ると、アとイは“do ~ while”を用いた後判定繰返しであり、ウとエは“while ~ endwhile”を用いた前判定繰返しである。前判定と後判定の大きな違いは、前判定が繰返し処理を「1回も行わないことがある」のに対し、後判定は繰返し処理を「少なくとも1回は行う」という点である。

この点を念頭に置きつつ、まず繰返しの中(空欄aとbで囲まれた部分)でどのような処理が行われているかに注目すると、“if ~ else”を用いて、

“☆”または“★”のどちらか一つを出力する
処理が行われている。具体的には、

①変数 starColor が“SC1”に等しい場合は、

“☆”を出力して、

starColor に“SC2”を代入する

②変数 starColor が“SC1”に等しくない(“SC2”に等しい)場合は、

“★”を出力して、

starColor に“SC1”を代入する

のどちらか一方の処理を行うことになる。

ここで、問題文に“引数 num の値が0以下のときは、何も出力しない”とあることに注意する。つまり、num の値によっては、繰返し処理は行わずに終了しなければならない。そのためには、前述の「前判定」の繰返しを採用することになる(解答はウかエに絞られる)。

さらに解答を絞るために、「num が0であった」場合を例に、空欄aに入る繰返しの条件式を検討する。条件式の候補は、

ウ: cnt が num 以下

エ: cnt が num より小さい

である。変数 cnt は変数を宣言する際に、

整数型: cnt ← 0

のように値が0で初期化されている。よって、num が0であった場合、cnt と num の値は等しいことになり、ウとエはそれぞれ、

ウ: 条件式が「真」となり、繰返し処理が実行される

エ: 条件式が「偽」となり、繰返し処理は実行されない

となる。よって、“引数 num の値が0以下のときは、何も出力しない”のは、エの、

空欄a: while (cnt が num より小さい)

空欄b: endwhile

の組合せである。

なお、前述の①及び②で最後に starColor に“SC2”または“SC1”を代入するのは、“☆”と“★”を「交互に」出力するためである。すなわち、

“☆”を出力した後は、次の繰返し処理で“★”を出力するための

準備として starColor の値を“SC2”にしておく

というように制御している。