

2022年(予想)の学習指導要領改訂に向けた

試作教科書2012「情報I」

2012.10.1版

情報処理学会初等中等教育委員会

2012.10.1

## はじめに

情報処理学会初等中等教育委員会は、初等中等段階における情報教育・情報技術教育がよりよいものとなるよう、さまざまな活動をおこなっています。その活動の一つに、高等学校共通教科「情報」の内容・範囲に関する検討と提言があります。これまで私たちは、1999年告示/2003年実施指導要領(「情報A」「情報B」「情報C」の3科目構成)、2008年告示/2013年実施指導要領(「社会と情報」「情報の科学」の2科目構成)それぞれについて、その公表に先立ち、試作教科書という形でその内容・範囲に関する提案をおこなって来ました。

そして、今後も10年間隔で指導要領が改訂されるとすれば、次の指導要領に対する提案を行う時期となっています。本「試作教科書2012」は、この「次の」(2018年前後に告示)指導要領において、どのような内容・範囲を扱うのが適切かを検討した結果に基づいています。

私たちは、共通教科「情報」は、必修の科目「情報I」(2単位)およびその履修後に興味・関心を持つ生徒を対象とする「情報IIx」(1科目以上、各2~4単位)から成るのが適切である、と考えます。本文書は、そのうちの「情報I」に相当する内容をとして必要と思われる範囲を、「教科書」の体裁で解説したものです。

ただし、実際の教科書のように丁寧に説明すると、分量が多くなり、読むのに時間が掛かりますし、要点が分かりにくくなります。そこで、この文書では記述をできるかぎり簡潔にし、少ない分量で済むように工夫しました。言い替えれば本文書は、私たちが適切と考える「情報I」の知識体系(BOK, Body Of Knowledge)を文章化したもの、と捉えて頂いても構いません。

本文書の中で、概念の網羅性のために説明に含めてはありますが、実際に高校生に教えなくてもよいと考える範囲は薄い色で記してあります。また、キーワードのうちで、そのような概念や考え方は扱いたいけれど、その単語そのものは取り上げなくてもよいと思えるものも同様に、その語だけ薄い色\*で、さらに「\*」を付して記してあります。

本文書の内容や記述については、あくまでも叩き台であり、私たちはこれに対してさまざまなご意見を頂きたいと考えています。ご意見がある方は積極的にお知らせ頂けるよう、お願い致します。

情報処理学会初等中等教育委員会 高校「情報」WG

### 謝辞

本文書の作成に当たっては、初等中等教育委員会・情報処理学会情報処理教育委員会のメンバー、井筒勝信氏(北海道教育大学)、中山代志子氏(弁護士)をはじめ、多くの方のご意見を参考にしました。ここに感謝致します。

# 目次

<b>第 1 章</b>	<b>情報とその表現</b>	<b>5</b>
1.1	情報と情報の役割 . . . . .	5
1.1.1	情報の定義 . . . . .	5
1.1.2	情報が持つ性質 . . . . .	6
1.2	メディアと情報の表現 . . . . .	6
1.2.1	情報とメディア . . . . .	6
1.2.2	コンピュータとデジタル情報 . . . . .	7
1.2.3	デジタル情報のさまざまな扱い . . . . .	9
1.2.4	数値のデジタル表現 . . . . .	10
1.2.5	その他の情報のデジタル表現 . . . . .	12
1.3	情報の伝達と表現 . . . . .	15
1.3.1	情報伝達とデザイン . . . . .	15
1.3.2	例: プレゼンテーションのデザインと実施 . . . . .	16
<b>第 2 章</b>	<b>コンピュータとネットワーク</b>	<b>19</b>
2.1	コンピュータとソフトウェア . . . . .	19
2.1.1	コンピュータはどこにあるか? . . . . .	19
2.1.2	コンピュータの構成と動作原理 . . . . .	20
2.1.3	さまざまなソフトウェア . . . . .	21
2.1.4	ソフトウェアとその作られ方 . . . . .	22
2.2	アルゴリズムとプログラム . . . . .	24
2.2.1	アルゴリズムとその基本要素 . . . . .	24
2.2.2	選択 . . . . .	24
2.2.3	繰り返し . . . . .	26
2.2.4	プログラミング言語による記述 . . . . .	26
2.3	ネットワークの構造と原理 . . . . .	28
2.3.1	インターネット . . . . .	28
2.3.2	インターネットの接続技術 . . . . .	29
2.3.3	セキュリティとその対策 . . . . .	30
2.3.4	TCP/IP の主要なサービス . . . . .	33

<b>第3章</b>	<b>コミュニケーションと情報モラル</b>	<b>37</b>
3.1	コミュニケーションとメディア	37
3.1.1	コミュニケーションとは	37
3.1.2	メディアリテラシーと情報操作	38
3.1.3	有効なコミュニケーションのために	39
3.2	情報倫理と情報モラル	40
3.2.1	モラルと倫理	40
3.2.2	トレードオフとジレンマ	41
3.3	情報社会における法と個人の責任	42
3.3.1	法と個人の責任の関係	42
3.3.2	知的財産権	42
3.3.3	個人情報とその保護	44
<b>第4章</b>	<b>情報社会と問題解決</b>	<b>47</b>
4.1	情報社会と情報システム	47
4.1.1	情報社会とその特徴	47
4.1.2	情報システム	48
4.2	情報社会と情報・情報技術	51
4.2.1	情報と職業	51
4.2.2	情報に関わる権利とビジネス	52
4.3	問題解決と情報	53
4.3.1	問題とその明確化	53
4.3.2	問題解決のプロセス	53
4.3.3	情報収集と分析	54
4.3.4	解の導出と選択	55
4.3.5	実行と評価	56

# 第1章 情報とその表現

この章では、情報とは何でありどのような性質を持っているか、情報を伝達する媒体としてどのようなものがあり、さまざまな情報がどのようにデジタル表現されるか、情報を効果的に伝達するためには表現にどのような工夫がなされ、伝達の際にどのような配慮がなされるべきか、などについて扱います。

## 1.1 情報と情報の役割

### 1.1.1 情報の定義

現代は情報社会であり、情報に大きな価値が置かれています。では情報 (information) とは何でしょう。情報とは「ものごとの知らせ」であり、受け手に取って価値を持つようなものです。単なる「ものごとの知らせ」はデータ (data) であり、それ自体に価値があるかどうかは分かりませんが、受け手によってはそのデータが情報になるのです。

たとえば、壁に温度計が掛かっていて温度が表示されているとき、それを気に止めないならその温度は単なるデータです。しかし、その温度を見てどれくらい厚着をして行こうか判断するのに用いるとすれば、そのデータは価値があるものとなり、情報だと言えます。

なぜ情報に価値があることになるのでしょうか。適切な上着を選べれば、その日が快適に過ごせ、それは自分にとって喜ばしいことです。喜ばしいことのためには、人は労力や手間を掛けます。このように、自分の行動を選択する際の判断に使うようなデータは、自分にとって価値のある「知らせ」すなわち情報だと言えます。

「自分に取っての価値」のように、「自分自身の考えや判断について」認識したり考えることをメタ認知 (metacognition) と呼びます。情報について考えるときは、さまざまな場面で、このメタ認知が重要となります。

演習 情報の価値を測る (定量化する) としたら、どのような基準があり得るか、考えてみなさい。

### 1.1.2 情報が持つ性質

では、情報にはどのような性質があるのでしょうか。まず、情報は単なる「知らせ」ですから、それ自体は形がありません。その「知らせ」が、たとえば紙に書いて伝えられたとしたら、その紙という形はありますが、同じ「知らせ」が喋って伝えられても手話で伝えられてもいいのですから、情報自体には形が無いといえます。

次に、情報は複製が容易です。これは情報はものではなく、相手に伝えても自分のところから無くならないということによっています。たとえば、紙に書いたメモを沢山のの人に見せるだけで、それらの人に同じ情報を簡単に伝えることができますし、伝えた後でも自分のところから無くなったりはしません。

情報の「量」はどのように測るのがよいのでしょうか。手紙だったら手紙の長さ(文字数)などは関係がありそうです。しかし、このような単に媒体が運んでいるデータ量 (data volume) だけで情報の量は測れません。なぜなら、同じ手紙を2回読んでも新しい情報は得られないからです(1回目によく読まなかったことが2回目に分かった、などのことは除いて考えています)。つまり、「知らなかったこと」の知らせのみに、情報が含まれています。

知っている/いないという2者択一からさらに進めて、「予測できそうな知らせ」と「珍しい(意外な)知らせ」では後者の方が知らせが伝えている情報の量は多いと考えることができます。たとえば、雨が降るか降らないか半々の地方での「雨が降った」という知らせと、1年に数日しか降水のない地方での「雨が降った」という知らせでは、運んでいる情報の量は後者の方が多いとするわけです。

そして、同じ量の情報を伝えるのに、「簡潔な手紙」と「長々しい手紙」のようにデータ量が違うこともあります。つまり、データがどれくらいの「密度」で情報を含んでいるか、ということも考えられるわけです。

**演習** データの量や情報の量を測る「単位」ないし「やりかた」を自分なりに考えてみなさい。

## 1.2 メディアと情報の表現

### 1.2.1 情報とメディア

メディア (media) ないし媒体とは、情報を伝える手段やもののことを言います。たとえば誰かに喋って情報を伝える時は、メディアは「会話」であり、かつ、使っている言葉つまり「日本語」であり、かつ、その日本語の伝達に使われる「音声」であり、かつ、音を伝える「空気」

である、ということになります。つまり、ひと口にメディアといっても、さまざまなレベルで考えることができます。

新聞やテレビのように、1箇所から多くの人に向かって情報を伝えることのできるメディアをマスメディア (**mass media**) と呼びます。マスメディアによる情報伝達がマスコミュニケーション (**mass communication**) すなわち「マスコミ」です。

マスメディアと対比して、会話や手紙のような個人対個人で用いるメディアを**個人メディア\*** (**personal media**)、複数の個人が集まったグループ内で相互にやりとりするものを**グループメディア\*** (**group media**) と呼ぶこともあります。

メディアと情報伝達の目的には関係があります。たとえば絵画や音楽などは、発信者が自分のアイデアや感情を表現することを目的として用いる、**表現メディア (expressive media)** だと言えます。一方、電話やテレビは発信者から受信者に情報を伝えることを主に目的として用いる、**伝達メディア (transmission media)** だと言えます。メモリカードやDVD-Rのように、情報を保管するために用いるものは**記録メディア (recording media)** です。紙の上にかかれた文字のように、表現メディア (文学作品など)、伝達メディア (手紙など)、記録メディア (覚え書きや会議記録など) のいずれにも使われるようなものもあります。

情報伝達の形態もまた、メディアによって決まって来ることが多く見られます。たとえば伝達者どうしの関係でいえば、個人の会話のように1対1のもの、マスメディアのように1対多のもの、そしてグループでの会話や会議のように多対多のものがあります。また、伝達時の時間関係として、電話やテレビのように伝えているその同じ時点で受け取られる**実時間メディア\*** (**realtime media**) と、新聞や手紙のように後で受け取られる**蓄積型メディア\*** (**stored media**) があります。

従来においては、多くの人に情報を伝達することが可能なのはマスメディアにほぼ限られていましたが、今日ではネットワークの発達により、いち個人が容易に多数の人に向けて情報を伝達できるようになっています。またネットワークは情報伝達の形態についても、さまざまなものを柔軟に使い分けられるようにしてくれています。

**演習** さまざまなメディアとその特性について分類表を作ってみなさい。

### 1.2.2 コンピュータとデジタル情報

**アナログ (analog)** 量とは、温度、圧力、重さなど切れ目なく連続的に変化する量を言います。**デジタル (digital)** 量とは、ものの個数のように「とびとびに一定間隔で」変化する量を言います。

情報を表現するときにも、アナログ表現とデジタル表現があります。たとえば、アルコール式温度計はガラス管の中の液の上下で温度を表す

ので、アナログ表現です。デジタル温度計は、数字で温度を表すものです。数字による表現は、とびとびに変化するので、必ずデジタル表現となります。

デジタル表現は「とびとびのどれか」なので、多少読み取りづらくても(たとえば数字が多少かすれていても)、正しい値が読み取れるという特徴があります。その代わり、一番細かい値よりも小さな違いは表現できません。たとえば3桁表示の体重計であれば、「45.32Kg」も「45.26Kg」も同じ「45.3」としか表示されず、区別が付きません。一方、アナログ表現は連続的なのでいくらでも細かい表現が可能ですが、実際に読み取れる細かさには限界があります。たとえば、目盛り式の体重計で0.01Kg(100g)の細かさまで読み取るのはなかなか難しいでしょう。

コンピュータの内部では、個々の電子回路に電流が流れて「いる」「いない」、電圧が「高い」「低い」のように2つの値のどちらかだけを区別する**デジタル回路\*** (**digital circuit**) が使われています。このことから、コンピュータが扱う情報は、すべてデジタル表現された情報(デジタル情報)になります。

上で述べたように、コンピュータ内部での情報はすべて「2つのどれか」が基本になっているので、これを数字の「0」と「1」で代表させることがよくあります。「0と1のどちらか」という情報は、コンピュータが扱える情報の最小単位となります。これを**ビット (bit)** と呼びます。

もっと沢山の情報を表す場合は、ビットを並べたビット列を用います。たとえば、2ビットだと「00」「01」「10」「11」の4通りのうちのどれか、という情報が表せます。一般に  $N$  ビットとだと  $2^N$  通りのどれか、という情報が表せます。

コンピュータとは、このようにして情報を表したビット列を記憶したり加工したりすることを、非常に大量かつ高速におこなえる装置です。さらに、どのような加工をおこなうかを、**プログラム (program)** を通じて指示することで、柔軟に制御できます。つまりコンピュータとは、「ビット列をプログラムの指示に従って加工する装置」だといえます。

なお、今日の多くのコンピュータは、ビット列を**バイト byte**(1バイトは8ビット)単位で扱います。これは、1ビット単位で格納場所などを指定するよりも指定がしやすかったなどの歴史的事情によります。さらに、1,024バイトを1**キロバイト\*** (**Kilobyte, KB**)、1**メガバイト\*** (**MegaByte, MB**)、1,024MBを1**ギガバイト\*** (**GigaByte, GB**)、1,024GBを1**テラバイト\*** (**TeraByte, TB**) と呼びます。これらはデータ量の単位ということになります。1,024倍ずつになっているのは、 $1024 = 2^{10}$  なのでコンピュータでの扱いに便利なためです。

### 1.2.3 デジタル情報のさまざまな扱い

#### DA 変換と AD 変換

コンピュータはデジタル情報だけを扱うので、アナログ量をコンピュータで扱うときには、デジタル表現に直す必要があります。これを **AD 変換 (Analog-to-Digital conversion)** と呼びます。逆に、私たち人間が知覚できるものの多くは (明るさなどの) アナログ表現なので、コンピュータ内部のデジタル表現をアナログ量に直すことも必要になります。これを **DA 変換 (Digital-to-Analog conversion)** と呼びます。

アナログ量を AD 変換するには、元のアナログ量を細かい段階にあてはめて、「その段階の何番目か」を決めることによってデジタル表現にします。この操作を、「とびとびうちのどれかにする」ことから **量子化 (quantumization)** と呼びます。このとき、ぴったり当てはまらない部分については、正しく表現できません。これを **量子化誤差\* (quantization error)** と呼びます。量子化の細かさは通常、読み取った結果のビット数で表します。たとえば、8 ビットであれば 256 の段階、16 ビットであれば 65536 の段階にあてはめます。ビット数が多い方が量子化誤差は少なくできますが、データ量は多くなります。

アナログ情報のうちには、音や動画などのように、時間とともに変化するものもあります。時間とともに変化する情報をデジタル情報にするには、一定の時間間隔で信号を観測します。このことを **サンプリング (sampling)**、それぞれの観測値を **サンプル (sample)** と呼びます。そしてその各サンプルを量子化することで、デジタル情報にします。

**1 秒間あたりのサンプリングの回数をサンプリング周波数\* (sampling frequency)** と呼びます。サンプリング周波数が低いと (つまりサンプリングの時間間隔が長いと)、信号の細かい変化が記録できません。これを **サンプリング誤差\* (sampling error)** と呼びます。一般に、ある周波数の信号を記録するためには、その周波数の 2 倍のサンプリング周波数を用いればよいことが知られています。

DA 変換については通常、それぞれの出力装置の中で行われます。たとえば、ディスプレイはデジタル表現された明るさや色にもとづいて、画面のそれぞれの箇所がその表現された明るさや色で光るようにします。

#### 圧縮と伸長

さまざまな情報をビット列でデジタル表現したとき、元の情報の性質にもよりますが、ある程度の規則性が見られることがよくあります。この規則性を利用して、デジタル情報の量を減らすことができます。これを **圧縮 (compression)** と呼び、逆に圧縮したのから元のビット

列を復元することを伸長 (**expansion**) と呼びます。このとき、完全に元と同じビット列に戻るものを損失のない (**lossless**) 圧縮、おおよそ戻すが完全に同じにはならないものを損失のある (**lossy**) 圧縮と呼びます。損失のある圧縮は、音や画像など、多少の違いがあっても人間にとっては分からないような情報の圧縮に使われます。

## エラー検出とエラー訂正

デジタル情報を伝送したり記録・再生するとき、雑音などの影響でビット列の一部が正しく取り出せないことが起こり得ます。このように、情報の一部が元と異なってしまうことを誤り (**error**) と呼びます。誤りが含まれたままの情報をそのまま扱ってしまうとまずいので、次の2つの方法が取られます。

- 誤り検出 (**error detection**) — 誤りがあったということを検出する。
- 誤り訂正 (**error correction**) — 上に加えて、誤りの場所を元の状態に復元する。

誤り検出のみを使うのはおもに、ネットワーク上の伝送のように、間違いがあつたらデータを再送してもらうなど、入手し直せる場合です。DVD-R への記録などのように、入手し直すのが難しい場面では、誤り訂正が使われます。誤り検出も誤り訂正も、元のデータに冗長\* (**redundant**) なビット群を追加しておき、その追加のしかたの工夫をもちいて、検出や訂正を行うものです。冗長な情報とは、本来は無くても済む(他の部分から得られる) 情報を言います。

### 1.2.4 数値のデジタル表現

#### 二進法

コンピュータは文字通り「計算する (compute)」装置として生み出されたものであり、その初期においては数値を処理することが目的の大半を占めていました。現在のコンピュータの用途においても、数値に基づく処理は重要な位置を占めています。

コンピュータで数値を扱うときには、二進法 (**binary system**) を使います。私たちが普段使っている十進法 (**decimal system**) では0~9までの十個の数字を用い、1桁左に行くごとに表している数値が十倍になります。これに対し二進法では、0~1の二個の数字だけを用い、1桁左に行くごとに表している数値が二倍になります。たとえば、2進法で「1011」と表現された値を考えてみます。それぞれの数字は右から「1の

位」「2の位」「4の位」「8の位」を表します。従って、「1011」は「8が1個」「4が0個」「2が1個」「1が1個」ということになり、十進法で「 $8+2+1=11$ 」を表すこととなります。

$$\begin{array}{r} 1\ 0\ 1\ 1 \rightarrow 8 + 2 + 1 = 11 \\ 8\ 4\ 2\ 1 \end{array}$$

二進法の表現を十進法に直すには、上で例示したように、「1」に対応している各桁の位の値を足していけばできます。逆に十進法から二進法に直すのは、次々に二で割り算して行き、余りを逆順に並べることでできます。

$$\begin{array}{l} 11 \div 2 = 5 \cdots 1 \\ 5 \div 2 = 2 \cdots 1 \\ 2 \div 2 = 1 \cdots 0 \\ 1 \div 2 = 0 \cdots 1 \rightarrow \text{「1011」} \end{array}$$

実際にコンピュータなどで扱う場合は、ビット数を決めてそのビット数で数値を表し、 $N$ ビットの符号無し表現を用いた場合、表せる整数の範囲は  $0 \sim 2^N - 1$  となります。つまり整数をコンピュータで扱う場合、表せる範囲に限界があります。

**演習** 自分の学籍番号を二進法で表現してみなさい。隣の人の学籍番号と、二進法の形のままで、足し算してみなさい。足した結果を十進法に直してみ、正しいかどうか確認してみなさい。

### 負の数の表現

ここまででは、0以上の数値だけを扱っていましたが、これを**符号なし整数\*** (**unsigned integer**) と言います。負の数を扱いたい場合は、一番上のビットを符号ビット (**sign bit**) とし、これが「0」のときを正の数、「1」のときを負の数として、次のよう符号ビットが1のものをそっくり負の側にずらして対応させます (例は3ビットの場合)。これを**二の補数\*** (**two's complement**) 表現と呼びます。 $N$ ビットの二の補数表現を用いた場合、表せる整数の範囲は  $-2^{N-1} \sim 2^{N-1} - 1$  となります。

	符号なし	二の補数
7	111	
6	110	
5	101	
4	100	
3	011	011
2	010	010

1	001	001
0	000	000
-1		111
-2		110
-3		101
-4		100

二の補数表現を使っている場合、ある数の符号を反転するには、(1) 各桁の0と1を反転し、(2) 符号なし二進数だと考えて1足す、という操作でできます。たとえば、「 $2_{10} = 010$ 」→(1)「101」→(2)「110 =  $-2_{10}$ 」、 $-3_{10} = 101$ 」→(1)「010」→(2)「011 =  $3_{10}$ 」のような具合です。

二の補数表現を使うことの利点は、負の数の計算も符号なし二進法と同じやり方で計算でき、従って同じ計算回路がそのまま使えることです。たとえば「 $-2 + 3 = 1$ 」は「 $110 + 011 = 1001$ 」で、桁あふれを無視すれば確かに符号なし二進法の計算と同じになります。

### 小数点つき数の表現

小数点つき数 (fractional number) の場合もコンピュータの内部では二進法を使いますが、さらに広い範囲の数値を扱えるようにするため、内部では仮数 (mantissa) と指数 (exponent) に分けて表現します。十進法で例をあげると、4桁の仮数、2桁の指数を用いると、すべての数値を「 $\pm 0.000 \times 10^{\pm 00}$ 」という形で表すことになります。この場合、表せるもっとも絶対値の大きな数値は  $\pm 9.999 \times 10^{99}$  となります。また、0でない最も絶対値の小さな数値は  $\pm 0.001 \times 10^{-99}$  となります。

つまり、コンピュータで小数点つき数を扱う場合、表せる範囲に限界があるだけでなく、その範囲内の値でも誤差があります。一般に、コンピュータで小数点つきの計算をする場合、近似値 (approximation) による計算になっていると考えるべきです。

## 1.2.5 その他の情報のデジタル表現

### 文字のデジタル表現

コンピュータで文字を扱う場合は、まず文字に番号を割り当て、その番号をビット列で表現することでデジタル表現します。この「文字と番号の対応」のことを文字コード (character code)、文字コードと具体的なビット列の対応方法のことをエンコーディング (encoding) ないし符号化と呼びます。また、文字コードに含まれている文字の集まりのことを文字セット\* (character set) と呼びます。

コンピュータの初期から使われている文字コードは数字と英語のアルファベットおよび少数の記号だけを含むものでした。**ASCII** コードがそのようなものの代表です。ASCII コードは1文字を7ビットで表せるので、各文字をバイト単位でそのまま格納します。

1バイトでは最大256文字までしか表現できないので、日本語など文字種類の多い言語を扱う場合には、2バイト以上で文字を表すやり方が必要です。日本では**日本工業規格\* (JIS)**が日本語のための文字セットと文字コードを定め、これに基づいて日本語をコンピュータで扱ってきました。ただし、実際にビット列にする際には、**iso-2022-jp**、**Shift-JIS**、**EUC-JP**の3通りの符合化が使われています。いずれも、数字や英字の文字や記号についてはASCIIコードと併用することが前提となっています。

その後、世界各国の文字をすべて文字セットとして含めた**UNICODE**と呼ばれる規格が作られて広まりました。UNICODEの文字を符合化するやり方としても、**UTF-8**、**UTF-16\***など複数のものがあります。

**演習** 「あいうえお」など簡単な日本語を打ち込んだファイルに対して、そのビット表現を表示するツールを適用してみて、どの文字がどのように符合化されているか検討してみなさい。

## 音のデジタル表現

音とは空気の圧力の変化(振動)が波として空間を伝わって行く現象です。1秒間あたりの振動の回数(周波数)が高いほど、調子(音程)の高い音になります。周波数の単位はHzおよびkHz(1kHz = 1000Hz)です。また、振動の振幅(振幅)が大きいほど、大きな音になります。

私たちの耳がこの振動を感知すると「音が聴こえる」のです。音を電気信号に変換する機器であるマイクロフォン(マイク)は、圧力の変化を電圧の変化に変換します。逆に電気信号を音に戻す機器であるスピーカは、電圧の変化を空気の振動に戻します。

音をデジタル表現で記録するときには、マイクによる電気信号に対してAD変換(サンプリングと量子化)を行います。このとき、サンプリング周波数が高いほど、また量子化のビット数が多いほど、元の信号に近いデジタル情報となりますが、データ量も多くなります。

たとえば音楽CDでは、音質を考慮した結果、サンプリング周波数44.1kHz、16ビット、2チャンネル(ステレオ)で音をデジタル化しています。これは1秒間あたり $44100 \times 16 \times 2 = 1,411,200$ ビット = 176,400バイトのデータ量になります。音声データはデータ量が多くなるため、最近の音楽プレーヤなどでは損失のある圧縮を使ってデータ量を小さくして扱います。

サンプリングによる音声データとは別に、楽曲などを表現するのに MIDI\*形式のデータが使われます。MIDI 形式のデータは、楽器の種類、音程、音の出る/止まるタイミング、強さなどを数値的に表して並べたもので、サンプリングによる音声データよりもデータ量が小さくなります。MIDI 形式のデータは、MIDI 出力を持った楽器 (電子キーボードなど) から取り込んだり、MIDI データを入力・編集するプログラムを使って打ち込んだりして作成します。

## 画像のデジタル表現

画像のデジタル表現には、大きくわけて 2 つの形態があります。1 つはピクセル画像 (pixel image) ないしラスタ画像 (raster image) で、画像を縦横に並んだ多数のピクセル (pixel) の集まりとして表現します。個々のピクセルは明るさと色の情報を持ちます。もう 1 つの形式はベクタ画像 (vector image) で、こちらは図形などの輪郭線の情報を数式的に表します (それぞれの図形の色はピクセルと同様に表しますが、模様や連続的な変化を含めることもあります)。

デジタルカメラなどで取り込んだ画像はピクセル画像になります。これに対しベクタ画像は、ベクタ画像用の作画ツールで取り込んだり、ピクセル画像からプログラムを用いて変換するなどの方法で作られます。

ピクセル画像はピクセルの集まりなので、大きく拡大すると個々のピクセルも大きくなり、最後はぎざぎざに見えるようになります。これに対し、ベクタ画像は拡大してもなめらかさは変わりません。なお、ベクタ画像も画面やプリンタに出力するときはピクセル画像に変換されて表示・出力されます。

では、色とは何でしょう。もともと光は電磁波であり、私たちの目の中の視覚神経に電磁波の中の 3 つの波長に感応するものがあるため、この 3 つの波長をそれぞれ赤 (Red)、緑 (Green)、青 (Blue) の色として知覚するのです (そして電磁波の強さが明るさになります)。RGB の 3 つの色その混ざり具合でさまざまな色が知覚されます (この 3 色を光の三原色と呼びます)。

ピクセル画像の表現では、各ピクセルごとに RGB それぞれの強さを 8 ビットの二進表現の数値で表すものが多く使われます。これを (3 色で 24 ビットになることから) 24 ビットカラー (24bit color)、ないしフルカラー (full color) と呼びます。たとえば、縦 600 ピクセル、横 800 ピクセルの 24 ビットカラーのピクセル画像は、データ量が  $600 \times 800 \times 24 = 11,520,000$  ビット = 1,440,000 バイト になります。

画像もデータ量が多くなるので、圧縮を多く用います。Web で使われている画像ファイル形式では GIF\* と PNG\* で損失のない圧縮、JPEG\* で損失のある圧縮を使っています。

**演習** 好きなピクセル画像を用意し、それを適切なソフトで拡大して、ぎざぎざになることを確認しなさい。また、そのどれか1つの点を選び、その点と同じ色を RGB 値を指定して作ってみなさい。

### 動画のデジタル表現

人間の目には、短い時間間隔で少しずつ違う絵を見ると、なめらかに動いて見えるという性質があります。動画はこれを利用して、多数の画像を連続して記録/再生することで表現しています(アナログ映画も同様に、ロールフィルムに多数の写真を連続して焼き付けることで表現していました)。

動画の中の個々の画像(コマ)のことをフレーム (**frame**)、1秒間あたりに表示するフレーム数のことをフレームレート (**frame rate**) と呼びます(単位は fps ないしフレーム/毎秒)。なめらかに動いて見えるためには、およそ 10 フレーム/毎秒以上のフレームレートが必要とされています。

動画は画像よりもさらに一層データ量が多くなるため、圧縮の工夫がなされます。たとえば、フレーム全体をそっくり記録するのは一定時間毎に現れるキーフレーム\* (**key frame**) と呼ばれる特別なフレームだけにしておき、他のフレームは前のフレームとの違い(差分)だけを記録する方法などがあります。

動画には普通、音声と一緒に記録されます。このため、動画ファイル形式はその中にさまざまな方式で圧縮された音声と、さまざまな方式で圧縮された動画とを格納する容れものないしコンテナ\* (**container**) として働きます。中に入る音声や動画の圧縮方式を **CODEC**\* (コーデック) と呼び、それぞれ特徴をもった複数のものが使われています。

## 1.3 情報の伝達と表現

### 1.3.1 情報伝達とデザイン

「バベルの塔」の逸話にもあるように、情報が適切に共有できてはじめて、人々は協力して一つのことを成し遂げることができます。とりわけ情報社会である現代において、情報を相手に効果的に伝えることはとても重要です。

このためには、絵や図、文章、動画、Web サイトなど、どのような形態を取るにせよ、伝えたいことそのものが情報の中に確実に含まれていることに加えて、情報の表現が適切にデザイン (**design**) されていることが重要です。

デザインの具体的な目標は、情報の表現形態によってさまざまです。たとえばポスターであれば、使われている絵や図がテーマを適切に表していること、重要なことや伝えたいことが目立つようにはっきりと表現されていること、配色や絵柄が目立ったり目に心地よいこと、などを目標としてデザインがおこなわれますし、文書であれば構成が伝えたい内容をうまく表現できていること、文章が読みやすいこと、過不足ない内容であること、整形や配置が見やすく美しいことなどが目標になります。

このように、適切なデザインがなされることで、同じ情報を、同じ形態を用いて伝えようとする場合でも、より効果的に、受け手にとってより受け入れやすい形で、伝えることができるようになるのです。

### 1.3.2 例：プレゼンテーションのデザインと実施

ここではプレゼンテーション (presentation) を題材として、情報の伝達とそのためのデザインについて説明します。プレゼンテーションをデザインするという場合、どのような内容をどのような順番で説明するかをデザインするという面と、(スライドを用いる場合) プレゼンテーションに用いるスライドの構成や内容をデザインする、という2つの側面があります。しかし、スライドを提示しながら説明するわけですから、この2つは本来並行して行うのがよいでしょう。

まず最初に、プレゼンテーションの内容として取り入れたいことがらをそれぞれ別個のメモ用紙などに書き、次にこれらを見渡して、関連があつて一緒に説明するものどうしをまとめます。そして、説明する筋道やつながりを考えながら順番を検討します。このような、紙などに内容構成の断片を書いて並べながら検討する方法を**ストーリーボード\* (storyboard)**と呼びます。ストーリーボードはプレゼンテーション以外にも、文章(レポート)、Web サイト、動画(映像)などさまざまな種類の情報のデザインに有効です。

ストーリーボードができてきたら、各スライドがどのようなものになるかがおおよそ分かってきます。プレゼンテーションには決められた時間があるので、平均してスライド1枚に2分程度の割り当てになるように、なおかつ各スライドに掛ける時間が均等になるように調整します。このとき、枚数が超過したり、1つのスライドの内容が多くなりすぎるようなら、内容を削って整理すべきです。

スライド構成のデザインができたら、各スライドの内容をデザインします。カンニングペーパーなどにたよらないために、スライドを見れば説明する内容が自分で分かるようにデザインしてください。これも、まずデザインは紙の上でおこない、そのデザインに基づいて素材(絵や写真など)を収集した後、プレゼンテーションソフトウェアで製作するのがよいでしょう。ソフトウェアを動かしながらデザインすると、操作に

時間が掛かるばかりでなく、全体として統一のとれないスライド群になるおそれがあります。スライド作成時には、次のことを守るべきです。

- 各スライドには明確なタイトルをつける。
- 字を詰め込まない。最大でも7行程度で、会場のどこからでもはっきり見える文字サイズを使う。
- 文章ではなく、説明する内容を表すキーワードを中心とする。
- 字体や文字飾りは「おふざけでない」見え方になるように選び、色は投影時に見やすいことを優先して選ぶ(背景色と文字色の対比がはっきりしているものが読みやすい)。
- 赤などの目立つ色は濫用しない。
- 図・表・写真などを活用し、文字ばかりになることを避ける。

スライドが完成したら、実際にスライドを提示しながら喋って練習をします。このときに時間も計測します。実際に練習しなければ、どれくらいの時間が掛かるかを正確に見積もることができません。プレゼンテーションでは各自の時間をきっちり守ることが求められるので、もし時間がオーバーするようならスライドの構成を見直して内容を減らす必要があります。

本番では、画面やスライドばかりを見るのではなく、できるだけ聞き手の人たちとアイコンタクトを取りながら喋るようにします。また、見ぶりや表情などもうまく活用できることが望ましいスタイルです。

発表が終わったら、できるだけ聞いてくれた人たちからフィードバックをもらい、直すべきところは記録しておいて次からは直せるようにすることで、上手なプレゼンテーションができるようになるでしょう。

**演習** 大まかなテーマを決めて、その中で各自が自分固有のテーマを決めた上で、プレゼンテーションを準備し、実施しなさい。その際、問題がないかチェックするためのチェックシートをデザインし、相互評価に使用してみなさい。



## 第2章 コンピュータとネットワーク

この章では、情報を取り扱うための装置であり、現代社会において重要な役割を果たしている機器であるコンピュータについて、その構造と、その上で動くソフトウェアの原理を説明します。さらに、情報を流通させる重要な手段となっているネットワークについても、その構造と原理について説明します。

### 2.1 コンピュータとソフトウェア

#### 2.1.1 コンピュータはどこにあるか？

前章で、コンピュータの内部では「すべての情報をデジタル表現している (ビット列で表している)」ことと、コンピュータが「ビット列をプログラムの指示に従って加工する」装置であることを説明しました。では、コンピュータはどんなところにあるのでしょうか？ まず、私たちの目につくところには、パーソナルコンピュータ (**personal computer**)、いわゆるパソコン (PC) があります。

しかし身近にあるコンピュータはこれだけではありません。たとえば、タブレット (**tablet**) やスマートフォン (**smart phone**) つまりスマホも、中身はコンピュータであり、一般の PC と同様にさまざまな作業がこなせます。電子ブックリーダー\* (**book reader**) や、以前からある携帯電話 (**cellular phone**) も、できることは限られていますが、中身はコンピュータと同様です。

さらに、エアコン、洗濯機、炊飯器などの家電製品も、今日ではコンピュータが入っているものが増えています。これらの場合、できることはそれぞれ限られていますが、その制御を「うまく」やるのにはコンピュータを使うのが便利なのです。このような、各種の機器にコンピュータが組み込まれているものを組み込みシステム\* (**embedded system**) と呼びます。今日では 1 台の自動車の中に多数のコンピュータが搭載されていたりします。

### 2.1.2 コンピュータの構成と動作原理

「目に見える」コンピュータであるデスクトップPCやノートPCでも、目立つ部分は画面やキーボードですが、コンピュータの「本体」は小さな箱に入っていたり、画面やキーボードの裏側に隠されていたりしてよく分かりません。まして組み込みシステムの場合は、どこに本体があるのかまったく分からないのが普通です。コンピュータは生まれたばかりのころは巨大な装置でしたが、技術の進歩とともに小型化したため、現在ではこのようにほとんど目につかないのが本当のところ です。

コンピュータの内部で、一番重要な部分は **CPU (central processing unit)** ないし中央処理装置と呼ばれる部分で、ここでプログラムの実行がすべて制御され、また演算などが行われます。二番目に重要な部分は **メインメモリ (main memory)** ないし主記憶 (と) と呼ばれる部分で、実行中のプログラムやプログラムが扱うデータはこの中に入っています。では、キーボードや画面は — これらはすべて **入出力装置 (I/O device)** と呼ばれ、キーボードやマウスのように人間から情報を受け取ったり (入力装置)、画面やスピーカのように人間に情報を伝えたり (出力装置) します。また、メモリに入らない大量の情報を保管するための **2次記憶装置 (secondary storage)** や、ネットワークとの間で情報をやりとりする装置もあります (これらは入力と出力の両方を行います)。まとめると次のようになります。

- CPU(中央処理装置) — 制御、演算
- メモリ (主記憶装置) — 記憶
- 入出力装置 — 外部との間のデータのやりとり

CPU もメモリも、今日ではごく小さいシリコン (硅素) のチップに微細な回路を焼き付けて作る、**VLSI(大規模集積回路)** と呼ばれるものでできています。VLSI の製造技術が進歩したおかげで、昔のコンピュータの何万倍もの能力を持つコンピュータを、携帯電話などの小さな機器の中に搭載でき、小さなバッテリーで動かすことができるようになったのです。

CPU の中には、上で説明したように、プログラムを実行するための制御回路と、演算回路、および演算で使う少量のデータを入れておく **レジスタ (register)** と呼ばれる記憶場所があります。一方、プログラムそのものやプログラムが扱う大量のデータは、メモリに格納されています。メモリには **番地 (address)** が割り振ってあり、どのデータでも、そのありかの番地を指定することですぐに取り出せます。

CPU がプログラムを実行するときは、メモリから1つずつ **命令 (instruction)** を取り出します。通常は、1つ命令を取り出したら、取り出す番地を1命令ぶん先に進めて覚えておくので、命令は並んでいる順番

に実行されていきます(このほか、次に実行する命令の番地を指定する命令もあります)。

それぞれの命令は、データをメモリとCPUの間で移動させたり、CPU上データに演算をしたり、条件に応じて次の命令の番地を変更するなどの、ごく単純な動作をします。CPUの動作は、(1) 命令を取り出す、(2) その命令の内容を解読する、(3) 解読内容に応じた動作を実行する、というステップをひたすら実行しています。コンピュータは、このような単純な動作を非常に高速に実行することで、大量のデータを処理するのです。一般的なデスクトップPCでは、1秒間に10億~100億個くらいの命令が実行できます。

では、コンピュータはなぜこんなに様々な仕事をこなすことができるのでしょうか。それは、コンピュータが実際におこなう動作は、すべて命令の並び、つまりプログラムによって定まってくるから、ということです。そして、プログラムはメモリに格納されていますから、このメモリ上のプログラムを別のものに入れ替えることで、コンピュータの動作をまったく別のものに切り替えられます。このような、プログラムをメモリに格納しておく方式のことを**プログラム内蔵\* (stored program)**方式と呼んでいます。

**演習** 身近なもので、コンピュータが搭載されているような機器を1つ選び、どこにコンピュータが搭載されているか検討してみなさい。また、その機器における入力装置、出力装置は何であるか、コンピュータがおこなう処理はどのようなものかも、考えてみなさい。

### 2.1.3 さまざまなソフトウェア

コンピュータが動作するためには、プログラムに加えてそのプログラムが扱うデータが必要です。このプログラムやデータを合わせたものを**ソフトウェア (software)**と呼びます。これと対比して、コンピュータの装置のことを**ハードウェア (hardware)**と呼びます。コンピュータが動作するためには、ハードウェアとソフトウェアの両方が必要なのです。

ソフトウェアにも、いくつかの種類があります。1つは、実際に私たちがやりたいこと — 文書を作ったり、音楽を聞いたり、絵を描いたり — をさせてくれるソフトウェアです。これを**アプリケーションソフト (application software)**と呼びます。私たちが普段使うアプリケーションとしては、**ワードプロセッサ (word processor)**、**表計算 (spreadsheet)**、**音楽プレーヤ\* (music player)**、**描画ソフト (drawing software)**などがあります。

コンピュータの初期のころは、メモリが少量しかなく複数のソフトを入れておくことができなかつたので、メモリにアプリケーションだけを

書き込んで動かしていました。しかし今では、コンピュータが強力で複雑になったので、コンピュータの上でさまざまなものを動かしたりするのも、ソフトウェアの助けを借りるようになりました。このような、「コンピュータを使うことを手助けするためのソフトウェア」のことを**基本ソフトウェア\* (primitive software)** ないし**システムソフトウェア\* (system software)** と呼びます。

その中でも、常にコンピュータの上で動いていて、私たちが必要なアプリケーションを使いたいと思ったらそれを動かしてくれるように、コンピュータ全体を管理しているソフトウェアがあります。これを、オペレーティングシステム (**operating system**) ないし OS といいます。私たちが普段目にする多くの PC 用の OS としては、**Windows\***、**Mac OS\***、**Linux** などがあります。また、スマートフォンやタブレットの上で動いている OS としては、iPhone/iPad 向けの **iOS\***、アンドロイド携帯/タブレットで使われている **Android\*** が代表的です。

OS の役割はコンピュータの管理だと書きました。コンピュータに備わっている CPU、メモリ、2次記憶装置、入出力装置などはすべて「必要なもの」という意味で**資源\* (resource)** と呼びます。OS の役割はこれらの資源を管理し、ユーザの指示に従ってコンピュータを動作させることです。たとえばアプリケーションを動かすときは、OS はユーザの指示にしたがってメモリを割り当て、そこにアプリケーションのプログラムを入れ、実行開始させます。また実行中は、アプリケーションが資源を使いすぎないように監視しながら、アプリケーションがネットワークやファイルなどを使う時にその手助けをします。

基本ソフトウェアは OS だけではありません。OS 以外の基本ソフトウェアの例としては、データベースサーバ、Web サーバなど、OS に直接は含まれていないけれど、さまざまなサービスを提供するのに必要な機能を持つものもあります。このような基本ソフトウェアは、OS とアプリケーションの「間に」あることから**ミドルウェア\* (middleware)** と呼ばれます。

#### 2.1.4 ソフトウェアとその作られ方

ソフトウェアはどうやって作るのでしょうか？ プログラムは命令の並んだものですから、CPU の命令を直接指定して並べて行くことで、プログラムを作ることも可能ではあります。これはコンピュータが生まれたたのころに行われました。しかし、CPU の命令を直接指定するのはとても大変です。

そこで現在では、まず人間に分かりやすいような書き方でプログラムを記述することが普通です。この書き方のことを**プログラミング言語 (programming language)** と呼びます。Java、C、C++、Perl、Ruby

などは代表的なプログラミング言語の名前です。これらの書き方で人間が直接書いたもののことを、ソースコード (**source code**) と呼びます。

プログラミング言語を実行するためのソフトウェア (これも基本ソフトウェアです) を、言語処理系\* (**language processor**) と呼びます。言語処理系の中には、ソースコードを読み込んで直接動作する形のもの、CPU が実行する命令の形に変換して、その後 CPU によって実行させる形のものがあります。前者は一般にインタプリタ\* (**interpreter**)、後者は一般にコンパイラ\* (**compiler**) と呼ばれます。

簡単なプログラムは、後で皆さんにも書いてみて頂きます。でも、世の中の大規模なソフトウェアは、非常に複雑で、多数の人が長い時間働いて作ることが普通です。ソフトウェア開発\* (**software development**) は、非常に手間とコストの掛かる作業なのです。

PC や携帯電話を購入すると、コンピュータを動かす上で必要な OS は最初から搭載されてきます (自作 PC などの場合は、ソフトウェアは何もなくて後から入れることもあります)。また、OS と一緒に、ブラウザ、テキストエディタ、お絵描き、ファイル表示などのよく使うアプリケーションも付属してくることが普通です。

しかしそれ以外のアプリケーションは、購入したりネットから無料で取り寄せたりして自分でコンピュータに入れます。スマートフォンの場合にはアプリケーションは「アプリ」と呼ばれて、専用の場所から取り込むようになっていますが、PC の場合はさまざまなアプリケーションはそれぞれの作成者が公開していたり、さまざまなソフトウェアを集めて配布しているサイトがあったりと、色々です。

有料のアプリケーションがある理由は、分かります。ソフトウェアを作成して、それを販売することで成り立っている会社があるからです。無償のものがあるのはなぜでしょう? 1 つは、OS などを作っている企業が、その OS の価値を高めるために OS の上で動くアプリケーションを無償で配布していることがあります。また、有料のソフトウェアを購入してもらうために、それよりも機能が限定されものを無料で配布することもあります。これら、企業が商売のために提供しているソフトウェアは商用ソフトウェア (**proprietary software**) と呼びます。

一方で、開発者の趣味や主義に基づいて、作成したソフトウェアやそのソースコードを無料で公開しているものもあります。これをフリーソフトウェア (**free software**) とかオープンソース (**open source**) と呼びます。ソースコードを公開している場合は、それを手直しして機能を拡張したりできます。ただし、そのように取り寄せて来たソフトを手直ししたり自分から配布することについては、ソフトウェアごとにライセンス (**licence**) 条項があつて決まっていることが普通です。商用ソフトウェアも、それぞれがライセンスで使用条件を定めています。

## 2.2 アルゴリズムとプログラム

### 2.2.1 アルゴリズムとその基本要素

アルゴリズム (**algorithm**) とは、プログラムの元となる、計算の手順そのもののことを言います。アルゴリズムを、コンピュータで実行できるようにプログラミング言語で書き表したものが、プログラムです。1つのアルゴリズムを、さまざまな言語で書き表すことができます。このとき、プログラムは使用する言語によってそれぞれまったく違って見えますが、アルゴリズムとしては同じものです。

アルゴリズムを検討するときには、いきなりプログラミング言語で書く代わりに、もっと検討しやすい書き方を使う方が間違いが少なくてみます。書き方としては、たとえば、フローチャート (**flowchart**) という、処理や選択を表す箱を矢線でつないだ図なども使われます。ここでは、日本語を使って「プログラムふうに」コードを記述します。次の例は、「2つの数値を読み込み、その合計を表示する」アルゴリズムです。

```
変数 x に数値を入力。  
変数 y に数値を入力。  
z ← x + y。  
数値 z を表示。
```

この処理は、1行目から順番に実行されて行きます。アルゴリズムはこのように、まず処理に必要なデータを入力し、処理 (計算) をおこない、最後に結果を出力する、という形になるのが普通です。

アルゴリズムの中に出て来る  $x$ 、 $y$  などの名前は変数 (**variable**) といい、計算途中で扱う値を入れておく「いれもの」です。「変数 ← 計算式」のようなものは代入 (**assignment**) といい、右側に書かれた計算式で計算した結果を、左側に書かれた変数に入れる (格納する) ことを表します。値を代入すると、その変数に前に入っていた値は上書きされて消えます。一方、計算式の中などに変数を書いた場合は、その変数に入っている値を取り出して使います。この場合は (別の値を代入するまで) 何回でも同じ値が取り出せます。

### 2.2.2 選択

アルゴリズムには、次のような基本的な原理があります。最初の2つについては、既にここまでで取り扱いました。

- 変数への代入によって計算が進んで行く。
- 複数の処理を並べて書くと、並べた順番に処理が行われる。

- 条件に基づいて、複数の処理のどれかを選択して実行できる。
- 条件に基づいて、一連の処理を繰り返し実行できる。

**選択 (selection)** の例を見てみましょう。次の擬似コードは、2つの数値のうちの大きい方を表示します。

```
変数 x に数値を入力。  
変数 y に数値を入力。  
もし  $x > y$  ならば、  
    数値 x を表示。  
そうでなければ、  
    数値 y を表示。  
枝分かれ終わり。
```

条件「 $x > y$ 」によって2つの処理「数値 x を表示」と「数値 y を表示」からいずれかを選択しているので、xの方が大きければx、そうでなければyが表示されます。では、両方が等しければ？ その場合はyが表示されますが、等しいのですから、それはxとも同じ値です。このように、アルゴリズムはプログラムではありませんが、動作は厳密に定まっていることが分かります。

ところで、同じ動作をするアルゴリズムで、次のものもあります。こちらは、「zにとりあえずxを入れておくが、もしyの方が大きければyを入れ直す」「そうすれば、最後にはzに大きい方の値が入っているはず」という考え方で作られています(この例のように、枝分かれには「そうでなければ」の側が無いものもあります)。

```
変数 x に数値を入力。  
変数 y に数値を入力。  
z ← x。  
もし  $y > z$  ならば、  
    z ← y。  
枝分かれ終わり。  
数値 z を表示。
```

どちらが分かりやすいでしょうか。アルゴリズムも(そしてプログラムも)、まったく同じ動作をするものを、いく通りも書けるのが普通です。では、その中でどれが「正解」なのでしょう？ それは、1つだけが正解ということではなく、どれも正解です。ただし、場面に応じて読みやすいもの、間違いにくいもの、などの違いがありますから、自分でよいと思うものを考えて使うようにするのがよいでしょう。

**演習** 3つの数を読み込み、最大のものを表示するアルゴリズムを作ってみなさい。複数の方針で作れるとなおよいでしょう。

### 2.2.3 繰り返し

こんどは、繰り返しの例を見てみましょう。次のアルゴリズムは、正の整数を入力すると、1からその数までの整数すべての合計を表示します。

```
変数  $x$  に数値を入力。  
 $sum \leftarrow 0$ 。  
変数  $i$  を 1 から  $x$  まで変えながら繰り返し、  
     $sum \leftarrow sum + i$ 。  
繰り返し終わり。  
数値  $sum$  を表示。
```

ここで出て来る繰り返しは、変数(ここでは  $i$ )を指定し、その変数を1つずつ変化させながら本体部分を繰り返すものです。たとえば、 $x$  に5を入れた場合は、本体部分は次のように書いたのと同事です。

```
 $sum \leftarrow sum + 1$ 。  
 $sum \leftarrow sum + 2$ 。  
 $sum \leftarrow sum + 3$ 。  
 $sum \leftarrow sum + 4$ 。  
 $sum \leftarrow sum + 5$ 。
```

だったら最初からそのように書けばよいと思ったかも知れませんが、「いくつまでの」合計を計算したいかは  $x$  を読み込んで見るまで分からないので、最初から用意しておくことはできません。また、1000までの合計を取るのに1000行書くのは大変ですが、このアルゴリズムなら元のまま大丈夫です。

ところで、「 $sum \leftarrow sum + 1$ 」とは何でしょう。これは、「現在  $sum$  に入っている値と、1とを足し、その結果を  $sum$  に代入する」ことになりますから、 $sum$  の内容は1増えます。次の周回では2増え、次の周回では3増え、というふうに繰り返されるので、(最初  $sum$  は0にしてありましたから)最終的に  $sum$  には1~ $N$  までの数の合計が入り、それが最後の行で表示されるわけです。

このように、繰り返しを使うことで、短いアルゴリズムでも大量の計算を指定できます。また、これと選択、代入を組み合わせることで、とても複雑な計算でも指定できます。

### 2.2.4 プログラミング言語による記述

先に説明したように、今日ではプログラムは人間に分かりやすいプログラミング言語でプログラムを記述します。ここでは **JavaScript** という言語をもちいます。この言語は Web ページの中に埋め込むことができ、一般的なブラウザ上で実行することができます。

ここまででアルゴリズムについて説明しましたが、最後のアルゴリズムを実際にプログラムとして動かしてみましょう。次の例では、`<script>… </script>`は Web ページを記述する HTML の記法の 1 つで、この内側に JavaScript プログラム (この例では 4 行あります) が書かれていることを表します。

```
<script>
var x = parseFloat(prompt('数 x を入力'));
var sum = 0;
for(var i = 1; i <= x; i++) {
    sum = sum + i;
}
document.writeln('合計は: ' + sum);
</script>
```

プログラムの説明は次の通りです。

- 「`var 名前`」は、指定した名前の変数を使うことを表します。
- 「`=`」は、JavaScript 言語で代入を表す記号です。(「等しい」を表すときには区別のため「`==`」と書きます)。
- 「`prompt('メッセージ')`」は、メッセージを表示して、1 行ぶんの文字列 (**string**) をユーザに入力してもらい、その文字列を返します。文字列とは文字の並びのことで、キーボードからは文字を打ち込むので、ユーザが打ち込んだものは文字列です。また、プログラムの中で「`'...'`」または「`"..."`」で囲んだものも、文字列です。
- 「`parseFloat(...)`」は、「`...`」の部分で用意した文字列 (この場合はユーザに入力してもらった文字列) を数値に変換して返します。
- 「`for(var i = 0; i <= x; i++) { ... }`」は、「変数 `i` を用意し、まず `0` を代入し、`i` が `x` 以下である間、`i` を 1 つずつ増やしながら繰り返す」と読みます。これで上のアルゴリズムにあった繰り返しが実行できます。
- 「`sum = sum + i;`」は上に説明した通り、計算と代入を行います。
- 「`document.writeln(...);`」は、「`...`」の部分で指定した文字列をブラウザの窓に表示します。「`+`」は左右どちらかが文字列のときは文字列をくっつける処理になり、その時、数値は文字列に変換されます。

たった数行のプログラムであっても、それぞれの書き方はすべて構文 (**syntax**) に従っていて、それぞれに対応して意味 (**semantic**) が決まっています。このように、プログラミング言語は非常に緻密で厳密に作られています。このため、大きく複雑なプログラムを作る作業は、とても手間が掛かるのです。

**演習** 例題プログラムを打ち込み、ブラウザで実行させてください。動いたら、さまざまな値までの和を計算してみてください。1億とか10億だとどれくらい計算に時間が掛かるでしょうか？ また、階乗 ( $1 \times 2 \times 3 \times \dots \times N$ ) の計算に直して見て、いくつくらいまで正確に計算されるか調べてください。

## 2.3 ネットワークの構造と原理

### 2.3.1 インターネット

ネットワーク (**network**) とはもともと、網目状の構造をしたもののことを言います (交通網とか連絡網などもネットワークです)。今日では、コンピュータやその他の情報機器を相互に接続したコンピュータネットワーク (**computer network**) が、私たちの生活に大きな影響を及ぼすようになってきました。

コンピュータネットワークに含まれるの個々の機器や中継点のことをノード (**node**) と呼びます。ノードにはコンピュータが搭載されていて、通信を適切な方向に振り向けることができます。この機能により、複数のネットワークどうしを結んだ場合、その中のどの2つのノードも相互に通信ができるようになり、全体としてより大きなネットワークが構成されます。このようにして、世界中の多数のネットワークが相互に結びあっているのが、インターネット (**the Internet**) なのです。

私たちの家庭や学校などからどのようにしてインターネットへの通信が行われているのかを見てみましょう。まず、家庭や学校には、そこにある複数の機器 (PC など) を相互に接続する LAN (**local area network**) が構成されていることが普通です。多数の機器を定常的に接続する場合は、ケーブルとハブ (**hub**) を用いて LAN を構成しますが、最近では、線をつながなくても通信できる無線 LAN (**wireless LAN**) が普及したため、LAN の構成はより手軽になっています。

この LAN をインターネットに接続するには、ISP\* (**internet service provider**) と呼ばれる業者と契約します。ISP は自分が保有しているネットワークと加入者の LAN との間を、無線、光ファイバー、電話線などを通して接続します。そして、ISP のネットワークが他の ISP や、地域ネットワーク、大学間ネットワークなどのプロジェクトが維持しているネットワークなどと相互に接続することで、インターネットが構成されています。

**演習** 自分の家や自分の学校の情報機器の「接続地図」を調べて描いてみましょう。その上で、どのようにしてインターネットまで接続されているのかも検討してみましょう。

### 2.3.2 インターネットの接続技術

インターネットの各ノードは、**TCP/IP\***と呼ばれる通信規約ないしプロトコル (**protocol**) に従って互いに通信をしています。さまざまな国の人たちでも、共通の言葉である英語で喋ることで互いに意思疎通ができるように、さまざまな国のさまざまな製品であっても、プロトコルという共通の約束ごとに従って動作するように作られていることで、互いに通信ができるのです。

TCP/IP では、**パケット (packet)** 方式による通信を行います。これは、ある決まった形式と上限サイズを持つデータの「かたまり」を定め、すべてのデータをこの「かたまり」単位で送受信するというやり方です。これにより、複数の通信を1つの線に「相乗り」させたり、混雑やエラーが発生したときにパケット単位での廃棄や再送を行ったりして、柔軟な通信を行うことができます。

TCP/IP で使用するパケットのことを **IP パケット\* (IP packet)** と言います。IP パケットの先頭部分には、**IP ヘッダ\* (IP header)** という部分があり、ここに「送り主」「宛先」が **IP アドレス (IP address)** と呼ばれる形式で格納されています。中継ノードでは、これらの情報に基づき、個々のパケットを正しい宛先に向けて送り出して行きます。この機能を**経路制御 (routing)** と呼びます。**IP (internet protocol)** とは、TCP/IP の一部で、IP パケットの形式やアドレスの形式、中継点の動作や中継点どうしのやりとりの方法を定めた約束ごとの集まりです。

IP には、古くから使われて来た **IPv4**(バージョン 4) と、IPv4 では容量不足になることを見越して新たに作られた **IPv6**(バージョン 6) とがあります。IPv4 のアドレスは 32 ビットで、8 ビットずつ「.」で区切って 10 進法で表記し、たとえば「192.168.0.1」などのように表します。一方、IPv6 のアドレスは 128 ビットで、16 ビットずつ「:」で区切って 16 進法で表記し、たとえば「2012:db8::1」などのように表します(::**と**なっている部分は全部 0 が詰まっているものとして、全体で 128 ビットの値として解釈します)。

IP アドレスには、LAN など限られた範囲の中だけで使われる**ローカルアドレス\* (local address)** と、世界中で重複なく割り当てられた**グローバルアドレス\* (global address)** があります。ローカルアドレスを持つノードからインターネットへの通信を行う場合には、途中の中継点でパケット内のローカルアドレスを中継点のアドレスに書き換えることで、グローバルアドレスの(世界のどこへでも送れる)パケットに直します。このグローバルアドレスは通常、ISP などが保有しているアドレスということになります。

パケットは上で述べたように、廃棄やエラーなどにより正しく届かない場合があります、その場合は再送が必要となります。この機能をエラー制

御 (**error control**) と呼び、TCP/IP のうちで **TCP\* (transmission control protocol)** と呼ばれるプロトコルが主に扱います。TCP を使うことで、あるノードが世界中にあるインターネット上の任意のノードとの間で、信頼性のある通信を行うことができます。たとえば世界中にある Web サーバの情報を見られるのも、この機能が前提となっています。

TCP によるエラー制御のやり方の原理は、相手先ごとにパケットに一連番号を割り当て、到着側ではこの番号を見て欠落や重複などをチェックし、正しく受け取ったものには OK の返事を出すというものです。送り側では、送ったパケットの控えは OK が来るまで残しておき、一定時間内に OK が来なければ再送します。

そして TCP の上にさらに、使用するサービスごとに異なるさまざまなプロトコルが定義され、使われています。TCP/IP という言葉は、このようなプロトコル群\* (**protocol suite**) 全体をさしているわけです。

**演習** 紙に 1 文字ずつメッセージを書き、中継役の人に渡すと、中継役の人がそれを相手に届ける、というゲームをしてみなさい。中継役の人が時々紙を捨ててしまったり、順番を入れ替えてしまっても大丈夫にするには、どうすればよいか考えなさい。

### 2.3.3 セキュリティとその対策

#### セキュリティとその 3 要素

インターネットは強力かつ便利な技術ですが、それは悪意を持つ人たちにとっても同様に便利、ということになりがちです。また、インターネットが広く使われるようになった結果、人為的なものでなくても事故や過失があった時の影響が大きくなっている、という面があります。

セキュリティ (**security**) とは「さまざまな危険から守られている」ことを表す言葉で、安全性 (**safety**) とほぼ同じ意味で使われています (安全性の方が「身体的な危険から守られる」というニュアンスが強くなります)。ここでは、コンピュータネットワークや情報システムに関わるセキュリティ (情報セキュリティ) を取り上げます。

情報セキュリティは、おもに次の 3 つの要素から成っています。

- **機密性\* (confidentiality)** — アクセスする権限を持つ者だけが情報にアクセスできること。
- **完全性\* (integrity)** — 情報が改ざんされたり消去されたりしないこと。
- **可用性\* (availability)** — 情報が必要なときに利用可能であること。

ここではセキュリティ確保のための重要な技術である暗号技術について説明した後、代表的なセキュリティ侵害とその対策について説明します。

### 暗号技術

情報の盗み見を防ぐためには、暗号 (**cryptgraphy**) が用いられます。暗号化 (**encrypt**) された情報は、傍受しても内容が分からないので、盗み見に対して安全になります。正しい受け手は、解読するための鍵 (**key**) を予め用意しておくことで、元のメッセージを復号 (**decrypt**) して内容を読むことができます。

ただしこの時、鍵をどのようにして安全に受け渡すかという問題が発生します。鍵をネット経由で渡すと、それが盗まれ、傍受された内容を解読されてしまいます。このような問題を克服するために、公開鍵暗号 (**public key cryptography**) と呼ばれる方式が考案されました。この方法では、鍵は公開鍵 (**public key**) と秘密鍵 (**secret key**) の対で生成され、そのうち公開鍵だけを「自分にメッセージを送りたい人」に渡します。送り手は公開鍵を用いてメッセージを暗号化して送ってくるので、それを秘密鍵で復号します。公開鍵は暗号化にしか使えないので、傍受者がそれを入手しても危険はありません。

公開鍵暗号と対比して、従来の暗号化と復号に同一の鍵を使う方式を**共通鍵暗号\*** (**common key cryptography**) と呼びます。共通鍵暗号も公開鍵暗号も、歴史的経緯や技術の発展の結果、複数の方式が存在し使われています。

暗号技術には、たとえ鍵が漏洩しなかったとしても、どのような暗号でも十分な時間を掛ければ解読が可能である、という側面があります。実用に使われている暗号方式の場合、現在のコンピュータの速度では解読まで何十年も掛かるから実用上問題はない、というふうに考えて使われています。ただ、CPUの高速化や解読技術の進歩により、これまで安全と思われていた暗号方式が安全ではないとされるようになった例もあります。

暗号と関連の深い技術に、**電子署名\*** (**digital signature**) があります。これは、秘密鍵を持っている人が文章全体と秘密鍵に基づいて計算した「署名」を付加し、他の人たちは公開鍵と「署名」を用いて、文章が確かにその公開鍵の持ち主によるもので、改ざんされていないことを確認するものです。電子署名には公開鍵暗号と同じ方式によるもの、そうでないものの両方があります。

## 侵入による脅威

ネットワークに接続されたシステムでは、外部からシステムに侵入 (**break in**) することで、情報を盗み出したり、システムを操って自分の悪事 (他のサイトの攻撃など) の手伝いをさせたりできてしまいます。

TCP/IP による通信は、特別な手段をこうしなければ、どの中継点でも覗き見できてしまいます。これによって、たとえばパスワードなどが漏洩すると、侵入される原因となります。言い替えれば、暗号化されていない通信を使用しているときにはパスワードを打つことを避けなければなりません。また、推測しやすいパスワードや、辞書に載っているような (英単語そのままなどの) パスワードも、推測やしらみつぶしによる解読がされやすいので、避けるべきです。

広く使われているソフトウェアにひそんでいる欠陥である **セキュリティホール\*** (**security hole**) についてシステムに侵入することも広く起こっています。これを避けるには、OS やブラウザなどのソフトウェアに対する **更新 (update)** をこまめに適用することで、欠陥はすみやかに除去することが重要です。

また、とくに必要がなければ外部から直接利用できないようにシステムを設定したり、システムまでのネットワークの経路が侵入に使われるようなプロトコルを通さないように設定することも有効です。一般に、ネットワーク機器に対して、不要なプロトコルを通さないように設定することを **スクリーニング\*** (**screening**)、侵入行為をしにくくしたり検出するような設定や機能全般のことを **防火壁 (firewall)** と呼びます。

## コンピュータウィルス

多くの問題を引き起こしているものの中に、**マルウェア\*** (**malware**) ないし広義の **コンピュータウィルス (computer virus)** があります。これは、悪意をもって作られたソフトウェアであり、さまざまなソフトウェアに自己感染することで広まったり、有用なソフトの一部として最初から仕込まれていて、騙されてそれを使った人が感染したりします。後者のタイプは **トロイの木馬 (trojan horse)** と呼ばれます。また、電子メールに自らを添付して手当たり次第に送信し、間違っただけでそれを開いた受信者が感染することで広まるものもあります。

ウィルスのおこなう悪さには、単にイタズラメッセージが出るだけのものから、システムの動作をおかしくしたりデータを破壊する悪質なまでの、さまざまなものがあります。利用者のさまざまな情報 (たとえばネットの閲覧履歴など) をこっそり送信する **スパイウェア (spyware)** などは、とくに目だつたふるまいをしないので、気付きにくく厄介です。

ウィルスは人間の不注意によって感染することが多いので、その方面での配慮が重要です。まず、ネットから取って来た素性の分からないソ

フトは起動してはいけませんし、知らない人からメールなどで受け取ったファイル(および知っている人からでも普段と様子の違うもの)は開いてはいけません。これらのものは、いかにも有用そうで無害そうな装いを務めてしますから、用心深く騙されないようにすることが大切です。

さらに、このようなソフトを受信したり間違っただけで開きそうになった時に検出し差し止めてくれる、ウイルス対策ソフト (**antivirus software**) を PC 上で動かしておくことも重要です。ウイルス対策ソフトは常時通信内容やファイル内容をチェックするために、PC が遅くなるという嫌う人もいますが、ウイルスに感染してしまったらもっと不愉快な目に合うことを考えましょう。

ウイルス対策ソフトの原理は、ウイルスに含まれているビット列のパターンを集めた**パターンファイル\*** (**pattern file**) を参照し、そのパターンにあてはまるものが送られて来たりファイルに格納されないか見張るといふものです。ウイルスは絶えず新しいものが現れるので、ウイルス対策ソフトの製造元はそれに対応してパターンファイルを更新しています。ですから、ウイルス対策ソフトの契約を切らさず、パターンファイルを常に最新に保つことが重要です(契約期間内で、かつ正しく設定してあれば、ウイルス対策ソフトは自動的にパターンファイルを更新するようになっています)。

### 2.3.4 TCP/IP の主要なサービス

#### ドメイン名と DNS

サービス (**service**) という言葉のもともとの語源は、他人に奉仕したり他人を助けたりすることですが、ネットワークの場合は他のコンピュータから呼び出せて利用できるような機能を提供しているものを言います。インターネット上で利用されているサービスにはさまざまなものがありますが、ここでは代表的なものを取り上げます。

既に説明したように、TCP/IP では通信の相手先となるコンピュータ(ホスト)を IP アドレスで指定しますが、IP アドレスは単なる番号であり、人間にとっては覚えにくく扱いにくいものです。そこでインターネットでは、ドメイン名 (**domain name**) と呼ばれるた名前を用意し、利用者はこの名前ですまざまなホストを指定します。典型的なドメイン名の例を示します。

`www.ecc.u-tokyo.ac.jp`

`www.amazon.co.jp`

`www.apple.com`

`www.london2012.com`

`www.nasa.gov`

ドメイン名は右から左に徐々に細かい分類になるように(米国などで住所を記すときのように)構造ができています。ドメイン名の一番右の部分は**トップレベルドメイン\*** (**Top-Level Domain, TLD**) と呼ばれ、2文字の場合はその名前を管轄している国(日本なら jp)、それ以外はそれぞれの種別を表します(com は企業、gov は政府機関)。jp の場合はその右側が種別で、co.jp は企業、ac.jp は大学等となっています。これらの下に、個別の企業名、(あれば) 下部組織の名称などが続き、一番左側が個別のホスト等の名前になります。

実際の通信はこの名前を IP アドレスに変換して通信をおこないます。**DNS (domain name system)** は、このドメイン名から IP アドレスを検索したり、その逆の検索をできるようにしてくれるサービスです。DNS もドメイン名の階層構造に合わせて、それぞれのレベルごとに分担して検索用のデータを用意するように作られています。

## 電子メール

電子メール (**electronic mail**) はユーザが相互にテキストのメッセージをやりとりできるようにしてくれます。電子メールアドレスは例えば次のような形をしています。

```
Myouji.Namae@example.ed.jp  
myname@example.com
```

ここで、「@」より左側はそれぞれのメールサーバにおけるどの利用者かを識別する名前であり、右側はメールを受信するメールサーバ (**mail server**) のドメイン名です。

誰かにあててメールを送信するときは、自分の使っているメールソフト\* (**mail software**) は自分が使っている手元のメールサーバに **SMTP\*** (**Simple Mail Transfer Protocol**) と呼ばれるプロトコルで接続し、メッセージを投入します。次に自分の手元のメールサーバは、受信先のメールサーバ (@ の右側で分かる) に、やはり SMTP でメッセージを送ります。メールの受信者は、**POP\*** (**Post Office Protocol**) または **IMAP\*** (**Internet Message Access Protocol**) と呼ばれるプロトコルで、自分のメールサーバからメッセージを取得して読みます。これにより、受信者のパソコンが稼働していなくても、メールを送ることができるわけです。

## World Wide Web

**WWW (World Wide Web)** ないし **Web** は、ネットワークのどこからでも **Web ページ (Web page)** という単位で情報を取り寄せられ

るサービスです。WWW では、**Web サーバ (Web server)** が公開しているページのありかを **URL (Uniform Resource Locator)** と呼ばれる形式で指定して **Web ブラウザ (Web browser)** で読み込むことで、世界中どこで公開されている情報でも瞬時に取り寄せて読むことができます。URL は次のような形をしています。

```
http://www.example.co.jp/example.html  
https://www.example.com/
```

ここで先頭部分 (:より前) の部分は、情報の取り寄せ方を指定しています。普段 Web で使われているのは **http:** で、これは **HTTP (HyperText Transfer Protocol)** というプロトコルによる取り寄せを表します。クレジットカード番号やパスワードなど覗き見られては困るような情報を、本当に正しい相手と分かっている (なりすましの心配のない) サーバとやりとりしたい場合には **https:** で指定される **SSL (Secure Socket Layer)** と呼ばれるプロトコルを使います。

「//」のすぐ次は、情報を取り寄せる Web サーバを指定し、その後の部分は、サーバごとに個別に解釈され、サーバ上でのどの情報を取り寄せるかを指定します。

Web サーバはさまざまな種類の情報をブラウザに返して来ますが、最も多いのは **HTML (HyperText Markup Language)** と呼ばれる書き方で記述された情報で、ブラウザ上では 1 つの **Web ページ (Web page)** として表示されます。

HTML では、ページ内容について、それぞれの部分が (見出し、本文、箇条書きなど) どのような内容なのかを **タグ (tag)** と呼ばれる書き方で印をつけて示します。これを **マークアップ (markup)** と呼びます。また、それぞれのタグの範囲について、色や文字サイズや配置などの **スタイル (style)** は、**CSS\* (Cascading StyleSheet)** と呼ばれる書き方で指定します。

Web ページの中には **マークアップ** を用いて、画像を埋め込んだり、他の URL を参照した **リンク (link)** を埋め込んだりできます。リンクの箇所を選択すると、そのリンクに対応する URL から内容が取り寄せられ、ブラウザに表示されるので、別のページに「ジャンプ」するように思えるのです。

Web サーバの側には、単にファイルが置いてあってその内容を返すようなページもありますが、プログラムが動いていて、そのプログラムが生成した結果を返すようなページもあります。後者の場合は、ブラウザの側でユーザが入力した情報をプログラムで受け取ることもでき、これによってブラウザから Web サーバ上のプログラムとやりとりできます。この機能を活用して、さまざまな Web サーバがさまざまな「WWW 上の」サービス (掲示板、ブログ、ネットショップ、オークション、決済な

ど)を提供しています。

一方、JavaScriptのようにブラウザの上でプログラムを動かすこともできるので、それとサーバ側の処理を組み合わせることで、細かい動作にたいおうする高度なプログラムも実現できます。つまり、WWWはそれ自身がサービスでありながら、他のサービスを構築する**基盤 (platform)**として使われているのです。

**演習** ブラウザ経由で使ったり動かすことのできるプログラムとして、どのようなものがあるか、色々集めてみなさい。それらを分類して整理してみなさい。

## 第3章 コミュニケーションと情報モラル

この章では、今日の情報技術で重要な地位を占めているネットワークと、その上でのコミュニケーションについて取り上げます。ネットワークは非常に幅広く緻密な技術の集まりであることや、ネットワーク上のコミュニケーションが私たちの社会に多くの新たな可能性を提供してくれていることが、分かると思います。

### 3.1 コミュニケーションとメディア

#### 3.1.1 コミュニケーションとは

コミュニケーション (**communication**) とは、人から人へ情報を伝達することを言います。情報とは「価値を持つデータ」でしたから、コミュニケーションにおいては送信側は自分にとって価値を持つものを送り、受信側はそれを自分の価値に照らして受け取ります。両者とも、価値があると思うからこそ、コミュニケーションという行為に関与するわけです。つまり、コミュニケーションでは双方の価値判断が問題になるのです。

既に学んだように、コミュニケーションはさまざまなメディア (媒体) を通して行われます。このとき、それぞれのメディアが持つ制約が、コミュニケーションの内容に影響を与えます。たとえば、電話で会話する場合、互いの表情や身振りは見えませんから、これらが必須であるような内容は伝えられません。しかし、伝えられるのが言葉だけというわけではなく、声の抑揚や調子や間などは活用することができます。

そもそも、直接目の前で会話をしている場合ですら、それぞれの人の考えていること全てが、言葉や表情や身振りなどで表現できるわけではありませんし、表現されたものを相手がどのように受け取るかも人によって大きな揺らぎがあります。つまり、コミュニケーションは本質的に、多くの制約のある「細い通路」を通して情報をやり取りすることなのです。

### 3.1.2 メディアリテラシーと情報操作

個人はもちろん、新聞、テレビ放送などのマスメディアもまた、それぞれの価値判断や意図を持って情報を発信します。従って、これらから情報を受け取る側には、その意図を考慮した上で情報を受け取る能力、すなわちメディアリテラシー (**media literacy**) が必要となります。

さらにメディアリテラシーには、自分が情報を発信する側になった場合に、受け手に適切に受け取ってもらえるように配慮する能力も含まれます。たとえば、自分がWeb ページを作成する場合、そのページが世の中の読みたいと思う人ならだれでも利用可能であって欲しいと思うはずで、このような考え方をユニバーサルアクセス (**universal access**) と呼びます。そのために、一部の人に見づらい配色を使わない、読む側で文字を大きくすること妨げる指定を使わない、などの配慮事項を知っておくことなども、メディアリテラシーなのです。

情報を受け取る側について言えば、個人が情報発信している Web ページは、それぞれの個人の考え方に基づいた内容を持っています。新聞なども、それぞれの立場に基づいて編集されていて、報道紙面においても、何を大きく扱い何を強調するかは、各紙の立場によって違って来ます。これらのことは、同一の題材を扱っている複数の Web ページを比較したり、同一のできごとを報道している複数紙の紙面を比較してみるとよく分かります。

また、これらと別の問題として、テレビ・新聞・雑誌などによる報道は、最終的にその内容に興味を持たれ、視聴や購入につながることを目的となっています。このため、必要以上にセンセーショナルな表現がなされることもあり、また「やらせ事件」(実際には起きていない事件を捏造して報道すること)なども起きたことがあります。

テレビや雑誌の商品広告などは当然、対象となる商品をよく見せて販売しようとする意図のもとに作られています。そのことを理解した上で有用な情報を得るために利用することもできます。

場合によっては、マスメディアや為政者が世論を自分の都合のよい方向に導くため、意図的に偏った情報を流すこともあります。これを情報操作 (**information manipulation**) と呼びます。その極端な場合は虚偽報道 (**false reort**) ですが、それほどでなくても、情報発信者にとって都合のよい情報だけを流し、都合の悪い情報は報じないという例は多く見られます。

センセーショナルな報道や情報操作などに惑わされて判断・行動することは結果的に自身や社会にとっての不利益につながりがちです。これらとは多少性質が違いますが、根拠のないうわさなども同様です。無自覚にこのような情報を他人に中継することは、有害な行動に加担していることを意識するべきでしょう。そしてこのような情報から身を守るた

めには、一面的な情報を鵜呑みにせず、さまざまな情報源を参照し、また自分できちんと考えて判断することが必要です。

**演習** 最近よく報道されているテーマを1つ選び、それらについて掲載されている新聞、雑誌、Web ページなどを(それぞれの種類ごとにできれば複数) 集めて比べてみなさい。どのような違いがあるかを表などに整理してみなさい。

### 3.1.3 有効なコミュニケーションのために

ここまで述べてきたことを逆の側から見ることで、有効なコミュニケーション、すなわち送信者・受信者の双方にとって価値のある情報伝達を実現するための指針が得られます。

まず、伝えられる情報が誤りを含まず(嘘がなく)、送信者の意図を正しく反映したものであることが必要です。このためには、送信者は送ろうとする情報の正しさを確認した上で送信するべきですし、自分が意図することは何であるかを自ら振り返り、また送信しようとする内容を見直して確かにその意図にかなっていることを確認してから送るべきです。

正しくない情報を送らない、というのは窮屈そうですが、必ずしもそうとは言えません。自分が聞き及んだ(確認の取れていない)情報について触れたければ、「たまたま耳にした、確認の取れていない情報であるが」と前置すればよいですし、同様に、自分の予想や考えを述べるときは「あくまでも自分の予想/考えだが」と前置すればよいのです(この逆、つまり確認が取れていない情報や単なる予想をあたかも事実であるかのように伝えることは、情報操作の基本的な手口です)。

次に、受信者の側も、まず前節で述べたように、情報は発信者の意図も考慮した上で受け取る必要があります。さらに、送られた情報を中立な立場で正確に読み取るように務める必要があります。よく「人は見たいものを見る」と言われるように、予断や先入観を持って情報に接すると、ありのままの情報でなく、予断や先入観と一致した情報を受け取ったものと勘違いしがちなものです。

最後に、感情的な対立や行き違いがあると、的確な情報のやりとりは難しくなります。これは、冷静な状態で発信する情報を用意するのが難しくなり、また中立的な立場で受け取ることが難しくなることによります。このような問題を避けるために、言葉づかいなどの表現は送る側は自分に厳しく(丁寧に)、受け取る側は相手に優しく(寛容に)なることが望まれます。

「自分が言おうとしていることは正しいか」「表現方法は適切か」「自分の受け取る態度は中立的か」などのことからは、自分自身に対する認識や判断、すなわちメタ認知を含んでいます。適切なメタ認知を行う能

力は、有効なコミュニケーションのために必須の能力だと言えます。また、次節で述べる倫理やモラルについて考える時にも、メタ認知が本質的に必要となります。

## 3.2 情報倫理と情報モラル

### 3.2.1 モラルと倫理

徳ないしモラル (**moral**) とは、正直や約束を守るなど、人間が内面的に持っている善悪や美意識に基づいた規範を指します。モラルは(他者から与えられるものではなく)内面的なものであり、社会的な慣習や法律などには左右されず、すべての人に共通のものであるという特徴があります。ただし、モラルは内面的なものであるため、なぜそう考えるのか、という問いに答えることは簡単ではありません。ここに、**哲学\*** (**philosophy**) の一分野である**倫理学 (ethics)** が果たす役割があります。

倫理学の中でも、正しく生きるとはどういうことか、善とは何か、などの根源的な問いを扱う部分を**規範倫理\***と呼びます。規範倫理における重要な真理の1つに**黄金律 (the golden rule)**があります。これは「自分に対するのと同じように、他者に対せよ」という言葉で表され、私たちの社会の基本的な考え方となっています。黄金律によることで、「なぜ人を殺してはいけないか」「なぜ嘘をついてはいけないか」などの問いに対して、「自分も殺されたくはないから」「自分も嘘をつかれたくないから」のように答えを得ることができます。

一方、具体的な個々のさまざまな状況・場面における「正しさ」を扱うのが**応用倫理\*** (**applied ethics**) です。応用倫理は、マナーやエチケットや法律など、個々の社会の慣習や約束(規則)も含めた「正しさ」を考えるため、「すべての人に内在」しているモラルよりも広い範囲を扱っています。たとえばこの中には、医者としてどう行動するか、技術者としてどう行動するか、などを扱う**職業倫理**も含まれています。

応用倫理の中でも、私たちが関心を持つのは**情報倫理 (information ethics)** ですが、これは情報技術に関わる場面での正しさや行動指針を扱います。これに対し、**情報モラル (information moral)** とは、情報社会で適正な活動を行うための基になる考え方や態度のことであり、つまり情報社会という状況を考えた上でのモラルを意味しています。なお、広い意味では、情報モラルという言葉は情報倫理まで含めた範囲の概念として使うこともあります。

なぜとりわけ、情報倫理や情報モラルを取り上げるのでしょうか。それは、情報技術が提供する状況はこれまでの私たちにとって経験のない

新しいものであることが多く、それに即した「正しさ」を新たに考える必要があるためです。

### 3.2.2 トレードオフとジレンマ

情報モラル (情報倫理) について、私たちが具体的に知っておかなければならないことは何でしょうか。実際にはそのような「知識」は、このテキストの各章に出て来るさまざまなことから全体だと言えます。たとえば、これまでに学んで来た (そしてこの後学ぶ)、コンピュータには何ができるか、ネットワークコミュニケーションはどのような特性を持つか、情報社会にはどのような出来事があるか、などのことからすべて、それぞれが関係する場面で「正しい」判断をおこなうために必要な「知識」です。さらに、そのような個別の知識以外に、情報モラルに関して知っておくべきこととして、トレードオフとジレンマがあります。

トレードオフ (trade-off) とは、ある望ましい事柄 A を増やそうとすると、他の望ましい事柄 B が減少する、という状況を表します。たとえば、相手に急ぎの電子メールで必要なことがらを説明するとします。詳しく丁寧に説明すれば、相手はそれを読んでよく分かりますが、書くのに時間が掛かるので相手に情報が伝わるのが遅くなります。逆に簡潔に説明すれば、すぐメールが送れますが、それで相手に十分なことが伝わるかが怪しくなります。トレードオフに直面したときは、「どの程度」がよいバランスかを考えて選択します。

ジレンマ (dilemma) とは、トレードオフよりも強い対立で、「ある基準からすれば A をしなければならないが、別の基準からすれば A をしてはならない」状況を言います。たとえば、画像を書き込んではいけない約束になっている掲示板で、誰かが怪我をしたので至急病院までの地図が欲しい、というメッセージが流れて来たとします。あなたは地図の画像を用意できるとします。通信手段がその掲示板しか無いとすれば、あなたは掲示板に地図の画像を書き込むでしょうか？

ジレンマに直面した時にどのように行動するか決めるよりどころは、「正しい」行動とは何かについての、あなた自身の考えしかありません (トレードオフに際しての選択も同様です)。そのことを意識し、自分としての「正しい」判断ができるように備えておくことは、情報モラルの大切な学習内容だと言えます。

**演習** ネットワークなどの情報技術が関係しているジレンマの場面を考えてみなさい。その場面で、あなたならどう行動するか、その理由はなぜかを文章で説明しなさい。また、あなたの友人にも同じことをしてもらい、行動や理由に違いがあるかどうか検討しなさい。

## 3.3 情報社会における法と個人の責任

### 3.3.1 法と個人の責任の関係

先に「正しさの依りどころ」について取り上げ、その一例とし黄金律について説明した。だとすると、なぜ法律 (**law**) やその他の規則・約束があるのでしょうか。たとえば、あることをしてよい/よくないの基準を黄金律のみに置いた場合、個人の主観によることになり、食い違いが生じます。また、たとえ食い違いが無いとしても、細かい個々の場合についてどうであるかを、そのつど原則に立ち返って議論するのはとても時間や手間が掛かります。

そこで、あるグループについて、「このグループのメンバーはこれらの約束に従う」という規則 (**rule**) を合意し、新たに加わる人はそれに同意した上で加わることにします。そうすれば、メンバーは (最初に合意したにせよ、同意して参加したにせよ) 「規則を守る」という「約束」をしたことになり、それに従うことが「正しい」ことになるわけです。学校の校則などの規則はこのようなものの例です。また居住地の法律など「生まれつき」であって明確な参加の同意が無いものは、まだ判断能力がない子供に代わって親が (子供の居住地を選択するという形で) 約束したものと考えることができます。

では、責任 (**responsibility**) とは何でしょう。個人がある行為を行う/行わないことを本人の自由意思により選択できる場合に、その行為や行為の結果に対して「責任がある」と言います。

### 3.3.2 知的財産権

さまざまな知的財産

知的財産 (**intellectual property**) とは、人間の知的活動によって作り出された、価値を持つもののことを言います。知的財産に関わる権利 (知的財産権) として、法律で認められている主なものとして、著作権 (**copyrights**) と産業財産権 (**industrial property right**) があります (そのほかにも細かいものがいくつかあります)。

産業財産権には、特許権 (**patent rights**)、実用新案権、意匠権、商標権などがあります。特許権や実用新案権は、技術などに関わるアイデアを保護するものです。意匠権は、商業的価値のあるデザインを保護するものです。商標権は、商標 (商品やサービスを表す名称やデザイン) を保護します。商号 (企業の名称) が商標であることもあります。これらはいずれも、著作権とは異なり、権利を申し出て認められた場合にだけ、その権利が保護されます。

このほかのものとしては、種や苗の品種を育成した物の権利、半導体回路の配置などが法律で保護されていますが、それ以外にも知的財産として権利を主張する人が現れたり、その結果さまざまな法律を根拠に権利が認められたことがあるものは多数あります。

## 著作権

著作権は、小説や絵画など人間の創作活動によって生み出された表現に対する権利です。日本の著作権法ではその対象(著作物)を「思想又は感情を創作的に表現したものであって、文芸、学術、美術又は音楽の範囲に属するもの」と定めています。ですから、単なる事実の列挙などは除かれますが、文章、絵、音楽などのほとんどのものは、作者が誰であっても、著作物ということになり、生み出された瞬間から保護の対象となります。また、日本では、コンピュータのプログラムも、著作権法によって保護されています。著作権の保護期間は、日本では著作者の死去から50年後までとなっていますが、国によってはより長い期間とするものもあります。

著作権は大きく分けて、著作者人格権といわゆる経済的著作権に分けられます。著作者人格権は、著作物を生み出した人に付随する、著作者の尊厳や名誉に関わる権利であり、他の人に譲渡することはできません。この中には、氏名表示権(どのように名前を表示するか決められる)、公表権(いつどのように公表するか決められる)、同一性保持権(勝手に改変されない権利)が含まれています。

一方、経済的著作権は、複製権(本やCDなどの形で複製できる)、上演権・演奏権(上演できる)、公衆送信権(ネットワークから送信できる)などがあります。このほかに、著作物の伝達者(レコード会社・放送事業者や歌手・俳優など)が持つ権利である、著作隣接権があります。

具体的に今日いちばん多く著作権の侵害があるのは、他人の著作物(音楽の曲、テレビ番組の動画、他人のWebサイトの内容など)をネットに流すことです。自分で演奏しても曲や歌詞はそれ自体が著作物であること、たとえ商売に使われているものでなくても、誰かが創作的に生み出した表現はすべて基本的に著作物だということに注意してください。また、自分が流さなくてもWebサーバや動画サイトにアップロードし、他人が取り寄せられる状態にすることは、公衆送信権を侵害したことになります。

他人の著作物を利用したい場合、2つの方法があります。その1つは、どのように使いたいかを説明した上で著作権者に連絡し、許諾を得ることです。著作権者の意に沿わないことでなければ、許諾が得られることも多くあります。

もう1つは、引用 (**quotation**) として扱うことです。引用とは、文章などの一部を「必要最小限」「明確に他の部分から区別できる形で」「出典を明記した上で」自分の著作物内に使用することです。引用は著作権法により認められている行為であり、許可を得る必要はありません。ただし、あくまでも自分の著作物が「主」である必要があります。また、絵や写真などは一部を取り出すこと(トリミング)は「改変」になってしまうので、引用は難しいと考えるべきです。

このほか、公共の利益のために、著作権者の権利が制限される場合があります。たとえば、(学校教育法で定められる意味での)学校において、(正規の)授業の一貫として必要とする場合、教員自らが行うという条件下で、著作物の複製や上映などが認められます。学校の授業で断り無しにテレビ番組の録画を見ることがあるのは、そのような根拠によっています。

また、個人が自分の家庭内などで自分や家族などのために複製を行うことも、私的複製 (**private copy**) として認められています。テレビ番組をレコーダで録画してもよいのは、これが根拠となっています。録画のコピーを友人などに渡すことは、この範囲を超えることになり、認められていません。

### 3.3.3 個人情報とその保護

私たちは、自分の個人的な行動などの事柄を、みだりに他人に知られたくないと考えます。プライバシー権 (**privacy rights**) とは、もともとはこのような、個人の私事を他人から隠すことの権利を意味していましたが、今日ではそれに加えて、他者が管理している自己の情報を制御する(内容をチェックしたり削除・訂正を要求する)権利までを含むものと考えられています(後者の部分を**積極的プライバシー権\***と呼びます)。

積極的プライバシー権をより確実に保護するため、日本では2003年に個人情報保護法が制定されました。この法律は個人情報を取り扱う事業者が従うべきことがらを定めたものであり、「5000権以上の個人情報をデータベース等として所持する」事業者が、本人からの開示請求や訂正・利用停止の請求に応じるべきであると定めています。

このような事業者でない組織の場合は、この法律によらず、その必要に応じて配慮しつつ個人の情報を扱えば済むはずですが、法律の名称が非常に一般的だったことから、「すべての場合において個人情報を受け渡してはいけない」かのような過剰反応の問題が一部で起きています。

なお、個人情報保護法では、個人情報を「生存する個人の情報であつて、特定の個人を識別できる情報(氏名、生年月日等)」と定めています。この定義も一般に広く用いられるようになっていきます。

また、個人の顔写真などについても肖像権\*があるとされ、むやみに公開されるべきでないとする考えが一般的です。Webサイトなどに他人が写った写真を公開する場合は、この面での配慮が求められます。一方、タレントなどその人の写真に経済価値がある場合には、パブリシティ権\*があるものと考えられ、これをおかすと経済的損害を与えたことになります。



## 第4章 情報社会と問題解決

この章では、ここまでに学んで来たコンピュータやネットワークなどの情報技術が私たちの社会にどのように役立ち、またどのように影響を与えているのか、それに関連してどのような約束があるのかを学びます。また、これらのことを踏まえた問題解決の進め方についても説明します。

### 4.1 情報社会と情報システム

#### 4.1.1 情報社会とその特徴

私たちが現代の社会を指して情報社会 (**information society**) という場合、大きく分けて次の2つの側面を考えています。

- 情報技術が進歩し、普及した社会 — コンピュータやネットワークに代表される情報技術が進歩し、これまでできなかった多くのことが可能になった。これにより、人間がやってきた単純作業の多くが自動化されたり、情報技術の利便性が認められ多くの人が使うことで、世の中全体にわたって情報技術が広まった。
- 情報により多くの価値が置かれる社会 — 20世紀型の大量生産・大量消費社会が行き詰まりを見せ、人々が物の所有や消費よりも情報を入手・観賞したり自ら生み出す活動に価値を見出すようになった。すなわち、価値の対象が物から情報に移行してきた。

この2つは互いに違うものですが、また関連もしています。つまり、情報技術の進歩と普及のおかげで、情報を受け取ったり生み出すことが容易になり、また逆に情報により価値が置かれるようになったために、情報機器がより発達し普及してきた、という相互作用があるわけです。

では、このような特徴を持つ情報社会において、私たちはどのようなことを知っておく必要があるでしょうか？ まず1番目として、情報技術がどこでどのように使われているか、それはどのような利点や弱点があるのか、ということを知る必要があります。情報技術は非常に便利なものですが、使えなくなったり、想定していないことが起きたりすることもあります。そのときに適切な判断や対処ができるために、これらのことを知っておく必要があるのです。

2番目として、情報や情報技術に関連して、どのような約束ごとがあるか、社会でどのような問題が起きているか、自分が問題を起こさないためにはどうしたらいいかを知る必要があります。これは、共通の約束に従い、他人に迷惑を掛けないために当然のことです。

3番目は、積極的に情報や情報技術をどのように使ったらいいか、ということです。これは「何がしたいか」が人によって違うため、一律に知っておくべきことを決められない面がありますが、たとえば問題に遭遇したときに情報や情報技術をどのように使ってその問題に対処するか、ということは誰にとっても知っておく価値があると思われます。

これらのうち2番目については、前章で既に扱いました。以下この章では、1番目の観点から、世の中で情報や情報技術が具体的にどのように使われているか、情報や情報技術に関わるどのような仕事があるかを取り上げた後、3番目の問題解決を取り上げます。

### 4.1.2 情報システム

#### 情報システムの定義と意義

システム (**system**) とは、複数の構成要素から成っていて、全体としてひとまとまりの機能を実現するようなものを言います。たとえば自動車も、多数の部品からできていて、全体としては人やものを移動させるという機能を提供してくれる、1つのシステムです。

情報システム (**information system**) とは、情報を扱うことを目的として構成されたシステムのことです。たとえば、銀行の情報システムは、誰がどのような講座にいくらお金を持っている、という「情報」を扱うためのシステムです。一番末端ではATMでお金を出し入れしたり通帳に記入したりできますが、それはそれはあくまでもその末端で情報を扱う「ついで」の機能であり、最も重要なのはお金の「情報」を正しく管理することです。

コンピュータが普及する前の銀行はどうしていたのでしょうか？それは、支店ごとに紙の台帳で講座を管理し、人間がそろばんで計算したりして情報を維持管理していました。それは非常に手間が掛かり大変ですし、またすぐにお金を出し入れできるのは自分の口座がある支店だけなので、とても不便でした。情報システムの導入によって、これらの問題は解決されたことになります。

つまり、情報システムを導入する効果は1つには効率 (**efficiency**)、省力化 (**cost cutting**) であり、もう1つには利便性 (**convenience**) の提供なわけです。しかし、情報システムの存在意義はこれだけではありません。たとば今日、Web上で検索サービスを提供している会社は、検索結果を表示するページと一緒に表示される広告の掲載権を販売して

利益を得ています。そして私たちにとっては、いくらか広告を目にすることと引き換えに、大量の情報の中から必要そうなものを素早く検索できることは価値があります。

このようなお金の儲け方は以前には無かったものです。つまり、情報システムは新しいビジネス創造 (**business creation**) や価値創造 (**value creation**) の可能性を提供し得るものであり、これからはむしろこちらの方が重要になっていく可能性も大いにあります。それは、情報システムは情報を扱うものであり、そして情報社会では多くの人が情報により多くの価値を見出して行くようになるからです。

## B2B と B2C

銀行の例のように、情報システムはもともとは特定組織内部の仕事のために導入され、その中で閉じたものでした。たとえば銀行の端末は銀行の支店や出張所などに設置され、利用者はそこまで来てシステムを使います。

しかしネットワークの普及により、利用者の来店を前提とせず、ネットワーク経由で直接取り引きを行うことが可能になると、その利便性から、このような形態の利用が急速に普及しました。これを電子商取引 (**electronic commerce**) ないし e-commerce と呼びます。

ネットショッピング (**online shopping**) は電子商取引の身近な例です。顧客は Web ブラウザなどでショッピングのサイトを開き、商品説明を読んで、欲しいものを注文します。商品は宅配便で届き、代金は振り込み、クレジットカード、代金引き換えなどで決済します。このような、企業と消費者が直接取り引きする形態の情報システムを **B2C (Business-to-Customer)** と呼びます。

私たちにはあまり目にする機会がありませんが、企業どうしの取り引きでも、取り引きをネット経由で行うことは多くあります。この場合は **B2B (Business-to-Business)** と呼びます。B2B は、金額が大きく、同じ取り引き相手と定常的に取り引きが行われ、決済も月単位でまとめて行うなどの点で B2C と異なっています。

## さまざまな情報システム

電子商取引の他にも、私たちの生活は多くの情報システムによって支えられています。スーパーマーケットやコンビニエンスストアなどでは、**POS (Point-Of-Sales)** と呼ばれるシステムを用いて、商品が販売されるごとにその記録を収集し、商品の補充に使ったり、経営のための分析に使ったりします。

鉄道や航空、演劇やコンサートなどの切符はオンライン予約 (**online reservation**) システムによって管理されており、全国どこからでも直前まで予約・購入することができます。

自治体や政府も目的に応じてさまざまな情報システムを運用しています。たとえば住民票などの情報も各自治体の情報システムによって管理されていますし、防災情報などの収集や伝達にもそのための情報システムが用意されています。

私たちの目に見えない部分でも、多くの情報システムが動いています。工場でさまざまな製品を生産しているところでは、生産管理システム (**production management system**) によって材料の仕入れから製造工程、在庫や出荷までを管理しています。流通業や運輸業などでも情報システムによって荷物の流通、トラック等の運行スケジュールを管理しています。航空機や鉄道などでは、全体の運行管理のためのシステム以外に、個々の機体や列車の上で動作し、パイロットや運転士の仕事をサポートするシステムなども存在しています。

### 情報システムの構造

ここまでに見て来たように、情報システムには非常にさまざまなものがありますが、どれも情報を取り扱うという点では同じであり、そのための機能の分担という観点から構造を考えることができます。

- データ処理 (**data processing**) — 情報システムには多数のコンピュータが含まれていますが、その中でも主要なデータの処理を行うところは、ある程度限定されている (まとまっている) のが普通です。
- データベース (**database**) — 多くの情報システムで最も重要なものは、取り扱っているデータであり、データを安全に保管し、システムの中で必要とする箇所からアクセスできるようにする機能であるデータベースは重要な中心的な役割を担います。
- 入力 — 情報システムが扱うデータがどこから来るかは、システムの形態によってさまざまです。Web 経由で利用するシステムの場合は、Web サーバが入力を受け取る箇所になりますし、専用の端末から入力をおこなうシステムもあります。機器などを制御するシステムでは、状態を検知するセンサー (**sensor**) からの入力があるものもあります。
- 出力 — 入力は出力と対になっているのが普通であり、Web 経由で使うシステムでは Web サーバ、専用の端末から使うシステムでは端末画面が出力先になります。機器を制御するシステムでは、モーターやヒータなどを制御する制御装置に対して出力を行うこととなります。

情報システムについて理解したり検討するときには、その情報システムの構造、とりわけ情報の流れを描いた図式 (**diagram**) を用いることが多くあります。そのような図式としては、**データフロー図\*** (**dataflow diagram**) をはじめ、さまざまなものが使われます。

## 4.2 情報社会と情報・情報技術

### 4.2.1 情報と職業

情報社会である今日においては、情報および情報技術に関わる産業や仕事が増えています。代表的なものを、以下に挙げてみます。

- マスメディア — 新聞、雑誌、映画、テレビ、ラジオなどのマスメディアは比較的古くからありますが、情報そのものを扱う産業であり、現代社会において重要な地位を占めています。
- コンテンツ産業 — 広告代理店や映像プロダクションなどは、各種のマスメディアや商用サイトで使用されるコンテンツを製作したり、その仲介を行います。Webサイトの企画や製作をおこなう **Web クリエータ (Web creator)** などこの中に含まれます。
- ソフトウェアベンダー (**software vender**)、ハードウェアベンダー (**hardware vender**) — コンピュータに代表される情報機器を製造したり、その上で動作するソフトウェアを製造する企業は、今日よく知られています。
- インフラ産業 (**infrastructure industry**) — 携帯電話のキャリアや通信事業者、データセンター (**data center**) 事業者など、情報技術に関わる設備を保持することで利益を得ている企業を言います。
- ソフトウェアサービス産業 (**software service industry**) — ソフトウェアに関わるさまざまなサービスを提供する産業全般を指す言葉です。ここに属する企業としては、顧客からの注文を受けて情報システムの開発を手配するシステムインテグレータ (**System Integrator, IS**) や、ソフトウェア開発業務をもっぱらおこなうソフトウェアハウス (**software house**)、情報システムの運用管理を行う運用サービス (**operation service**) 企業などがあります。
- eコマース産業 — B2B や B2C など、ネットを通じてさまざまなものを販売する商売を言います。
- その他の **IT 産業 (IT industry)** — 検索サービス、オンラインゲーム、オンラインモール、各種の情報サイト、SNS やブログなど、ネット上のさまざまなサービスは今日では収益を産むものになっており、これらを事業としておこなう企業が存在しています。

これらの業界における業務内容ないし職種としては、営業、経理、管理など通常の企業と同様のものもありますが、ソフトウェア開発者 (**software developer**) やコンテンツのクリエイター (**creator**) など、特別な技能や創造性を必要とするものも多くあります。

#### 4.2.2 情報に関わる権利とビジネス

情報やコンテンツはさまざまな形で保護されています。保護されているということは、その情報を誰もが自由に利用することはできないことを意味します。そこで、その利用を可能にすることと引き換えに対価を得ることが多くあります。

たとえば、音楽の著作物(曲や歌詞)に対しては、著作権者から委託や権利の譲渡を受けてこれらの使用の許諾を与え、その対価を得る組織が存在しています。また、映像や写真などについても、同様のことをしている企業があります。これらのものを全般に著作権ビジネス (**copyright business**) と呼びます。タレントなどの肖像権については、古くからその所属プロダクションが同様のことを行っているといえます。

パッケージソフトウェア (**packaged software**) など、販売されているソフトウェアは一般に、購入者はソフトウェアそのものを購入するというより、それを手元のコンピュータで使用できる権利を得ることに對して対価を払っています。ですから、これもある意味では著作権ビジネスの一種となっています。

マスメディアは通常、そのメディアが販売しているコンテンツに対して、著作権を持っています。このため、今日では多くのマスメディアがそれらのコンテンツを、(TV番組のDVDボックスなどのように)別形態で販売したり、ネット上で見られる権利を販売するなどの形で、2次利用による収入を得ています。

なお、レコード会社などは楽曲・歌詞に対する著作権隣接権を持つだけなので、2次利用に対しては著作権者の許諾を得ています。また、映画の著作物については、出演している俳優も隣接権者であるため、出演者全員の許諾を得られないなどの理由で2次利用が行えないことがあります。

著作権ビジネスは、著作権による保護(利用の制限)があるゆえに成り立ちます。このため、著作権の期限(著作者の死去から50年)を延長しようとする動きが著作権ビジネスの側からなされることが多くあります。

## 4.3 問題解決と情報

### 4.3.1 問題とその明確化

私たちはさまざまな場面で問題に遭遇し、それを解決しようと行動します。それがうまく行えれば、それは私たちにとってよいことですし、同じ問題を共有している他の人にとってもよいことです。そして情報や情報機器をうまく使うことで、よりより問題解決を行える可能性があります。このため本節では、問題解決について取り上げて行きます。

まず問題 (**problem**) とは一般に、「現在ある状況が望まれる状態と異なっていること」を意味し、問題に対する解 (**solution**) とは「その状態の食い違いを解消する方法」を意味します。

たとえば、「庭が草ぼうぼうで見苦しい」というのは「草ぼうぼう」という状態が「見栄えのよい庭」という望まれる状態と食い違っているという問題なわけです。この問題に対する解としては「草刈りをする」「除草剤をまく」「カーテンをしめて庭を見えなくする」などが考えられます。しかしこれらはその一方で、「疲れる/暑い」「毒性が心配」「部屋が暗くて不快」などの新たな問題を招くかも知れません。

問題に対しては、それが「誰の問題」なのかを考えることも大切です。たとえば、「見栄えのよい庭」を望んでいるのが「私」の母であるなら、それは「私」にとって問題でないかも知れません。しかし、母がその問題のために「私」に文句を言ったり、文句は言わなくても不愉快だったりすれば、そのことが「私」の問題となり、それを解決するために、やはり庭の問題を解決しよう、と思うかも知れません。

これらのことから分かるのは、問題について漠然と考えているだけでは問題解決のために必要なことがらが明確になりにくい、ということです。つまり、問題解決の最初の段階として、問題をきちんと認識し、その問題がどのような問題であるかを明確に記述することが必要です。記述すべきことがらとしては、たとえば次のものが挙げられます。

- 何が問題なのか (望まれる状態、現実の状態の記述)。
- その問題はいつどこでどのように起こっているか (5W1H)。
- 誰にとっての問題か。
- その問題の関係者 (問題を起こしている人、影響を受けている人)。
- どのようになることが、問題の解決と言えるか。

### 4.3.2 問題解決のプロセス

問題解決 (問題の解を求め、その解を実行すること) の際には、ただやみくもに考えるのではなく、一定の手順に従って系統的に行うことが効果的です。このような手順のことをプロセス (**process**) と呼びます。問題

解決に限らず、さまざまな作業において適切なプロセスに従うことは、次のような利点があります。

- 有効性が高いと分かっている順番で作業することができる。
- 必要な作業を抜かしたり、余分な作業をしてしまうおそれの小さくなる。

問題解決において一般的に有効とされるプロセスは、次のようなものです。

- 問題認識と定義 — 問題を認識し、何が問題であるかを文章として記述し、明確にすること(前述)。
- 情報の収集 — 問題の解を導く上で必要な情報や有用な情報を集める。
- 情報の整理と分析 — 集めた情報を整理したり、そこからうまく解が導けるように加工したりする。
- 解の導出と選択 — 可能な解をできるだけ多く集め、それらを比較して実行するものを選ぶ。
- 実行と評価 — 実際に解を実行し、その結果を振り替えて問題点やよかった点を確認し、今後に備える。

**演習** 実際に自分の身の回りにある問題だと思ふことをとりあげ、その問題がどのような問題か、記述してみなさい。記述できたら、他人にそれを見てもらい、過不足ない記述かどうか検討してもらいなさい。

### 4.3.3 情報収集と分析

問題解決において、きちんと情報の収集・分析をおこなわず、自分の予断・期待などに基づいて解を選んでしまうと、選んだ解法が実際には問題の解決につながらない、などの失敗が起きやすくなります。

このようなことを避けるには、関連する情報を洩れなく集める必要があります。たとえば、問題 X について、その原因が A であるだろうと予想した場合でも、A 以外に原因であり得るもの B、C、…をなるべく多く列挙し、A だけでなく B、C などの情報も平等に集めるのがよいでしょう。もちろん、関連する要因が検討の過程で新たに分かることもあるので、その場合はその時点で収集する項目を追加する必要があります。

調べる手段としては、今日ではネットで検索するのが便利かも知れませんが、それだけに限定せず、図書館などで文献を調べる、知っている人にインタビューするなど、複数の手段から適切なものを選ぶようにします。またその際には、情報の信憑性などについても留意する必要があります。

問題が特定の場所で起こっているのであれば、その様子を実地に観察 (**observation**) したり、それを直接見聞した人にインタビュー (**interview**) することは有効な手段です。そのときは、ただなんとなく見たり話を聞くのではなく、できるだけ多くのことに注意を払い、気付いたことがらをメモしたり、繰り返し起きるようなことであれば、その回数や時刻を記録するなどのことが必要です。

このように、何の情報をどのように集めるかを判断するのは簡単ではありません。後から振り替えて見られるように、何と何について、なぜ調べることにしたか、どのようにして調べたかなどを記録しておくことが望まれます。

分析 (**analysis**) とは一般に、問題の解につながるような洞察を得られるようにすることを目的として、データを整理・加工することを言います。

データが「何かがあった/なかった/どうだった」のような、性質について述べているものの場合、定性的\* (**qualitative**) であると言えます。これに対して、回数や数量のような数値の形で表されているものは、定量的\* (**quantitative**) と言えます。

定性的なデータの場合、個々の事項を小さな紙に書いて机などに並べ、共通することがらをグループ化し検討する、**KJ法\*** (**KJ-method**) などの分析方法が使われます。分析結果は、特徴的なことがらを並べて整理した表を作ったり、共通することがらの集まりや関係を図の形で示すなどの形が考えられます。

定量的なデータの場合は、数表やグラフなどの形で整理し、そこから特徴的な関係を見出すことを試みます。値の推移を見るのには折れ線グラフや棒グラフ、複数のことがらの関係を見るには相関グラフ (散布図) など、適切なグラフ形式を活用します。なお、相関があったからといって、その2つのことがらの片方が原因、他方が結果であるとは限らないので、グラフの読み方には注意が必要です。

#### 4.3.4 解の導出と選択

「問題の解を作り出す」というのはアイデアの必要な創造的な行為であり、必ずしも誰でも簡単に行える、というわけではありません。古くから、有効なアイデアのためには、自由にさまざまなことがらを連想し考えを広げる発散的思考 (**divergent thinking**) と、さまざまな考えの中から有効なものを理詰めで精選してゆく収束的思考 (**convergent thinking**) の両方が必要だとされています。

グループで問題解決を行う場合には、「何が一番よいか」などの突き詰めて話し合いになりやすく、収束的思考が中心となりがちです。このため、意図して発散的思考を行うための方法として、ブレインストーミン

グ (brainstorming) などの方法が提唱されています。また、KJ 法なども前半はさまざまなアイデアを出して紙に書いていく発散的な段階、後半は紙をみながらまとめていく収束的な段階、というふうに構成されています。

複数の解決案が得られたら、それらのうち有望と思われるものについて、具体的な実行計画を立ててみます。その過程で、それぞれの案についての利点や弱点、実行の手間やコストなどが分かってくるので、それは表などにまとめて記録します。最終的には、これらの情報をもとにして、最も望ましいと思う解決方法を選ぶことになります。

解を得るまでのアイデア出しや議論の過程、検討した内容、複数案の比較や最終案を選んだ理由などはすべて、記録を残しておきます。これらも、後から振り替えて問題解決プロセスを改良して行く上で重要な材料となるからです。

#### 4.3.5 実行と評価

問題解決において、案をまとめたとしても、実際にその案を実行に移せるかどうかは状況によってさまざまです。たとえば、必要なコストを手当てできなかつたり、許可を得られないなどの理由で実行に移せない場合もあります。

実際に実行に移せる場合は、計画に従って実行しながら、実行時の記録を取るようにします。また、最終的に問題がどの程度解決されたか、新たな問題は生じなかったかなどの記録も取り、全体としてのよしあしを評価します。

最後に評価があるのは、問題解決は通常1回だけ行えば終りというものではなく、類似した問題を繰り返し解くことが多いためです。このとき、繰り返しごとによりよい解が得られて実行できるようになれば、全体としてよりうまく問題解決が進んで行くことになります。このような繰り返しを通じた改善を **PDCA サイクル (Plan-Do-Check-Act cycle)** と呼びます。P は計画 (解の選択まで)、D は実行、C は評価、A は評価の結果に応じて次の問題解決を改良することを意味しています。

**演習** 実際に自分の身の回りにある問題だと思うことをとりあげ、これに対する問題解決を行ってみなさい。

# 索引

- 圧縮, 9
- アナログ, 7
- アプリケーションソフト, 21
- 誤り, 10
- 誤り検出, 10
- 誤り訂正, 10
- アルゴリズム, 24
- 暗号, 31
- 暗号化, 31
- 安全性, 30
- 意味, 27
- インターネット, 28
- インタビュー, 55
- インタプリタ\*, 23
- インフラ産業, 51
- 引用, 44
- ウィルス対策ソフト, 33
- 運用サービス, 51
- エラー制御, 30
- エンコーディング, 12
- オープンソース, 23
- 黄金律, 40
- 応用倫理\*, 40
- オペレーティングシステム, 22
- 音楽プレーヤ\*, 21
- オンライン予約, 50
- 解, 53
- 鍵, 31
- 仮数, 12
- 価値創造, 49
- 可用性\*, 30
- 観察, 55
- 完全性\*, 30
- キーフレーム\*, 15
- 規則, 42
- 規範倫理\*, 40
- 基盤, 36
- 基本ソフトウェア\*, 22
- 機密性\*, 30
- 共通鍵暗号\*, 31
- 虚偽報道, 38
- 記録メディア, 7
- キロバイト\*, 8
- 近似値, 12
- ギガバイト\*, 8
- 組み込みシステム\*, 19
- クリエータ, 52
- グループメディア\*, 7
- グローバルアドレス\*, 29
- 携帯電話, 19
- 経路制御, 29
- 言語処理系\*, 23
- 公開鍵, 31
- 公開鍵暗号, 31
- 更新, 32
- 構文, 27
- 効率, 48
- 個人情報, 44
- 個人情報保護法, 44
- 個人メディア\*, 7
- コミュニケーション, 37
- コンテナ\*, 15
- コンテンツ産業, 51
- コンパイラ\*, 23
- コンピュータウイルス, 32
- コンピュータネットワーク, 28

- サービス, 33
- 産業財産権, 42
- サンプリング, 9
- サンプリング誤差\*, 9
- サンプリング周波数\*, 9
- サンプル, 9
- 資源\*, 22
- 指数, 12
- システム, 48
- システムインテグレータ, 51
- システムソフトウェア\*, 22
- 私的複製, 44
- 収束的思考, 55
- 小数点つき数, 12
- 肖像権\*, 45
- 商用ソフトウェア, 23
- 省力化, 48
- 伸長, 10
- 侵入, 32
- 十進法, 10
- 実時間メディア\*, 7
- 冗長\*, 10
- 情報, 5
- 情報システム, 48
- 情報社会, 47
- 情報操作, 38
- 情報モラル, 40
- 情報倫理, 40
- ジレンマ, 41
- スクリーニング\*, 32
- スタイル, 35
- ストーリーボード\*, 16
- スパイウェア, 32
- スマートフォン, 19
- 図式, 51
- 生産管理システム, 50
- 責任, 42
- セキュリティ, 30
- セキュリティホール\*, 32
- 積極的プライバシー権\*, 44
- センサー, 50
- 選択, 25
- ソースコード, 23
- ソフトウェア, 21
- ソフトウェア開発\*, 23
- ソフトウェア開発者, 52
- ソフトウェアサービス産業, 51
- ソフトウェアハウス, 51
- ソフトウェアベンダー, 51
- 損失のある, 10
- 損失のない, 10
- タグ, 35
- タブレット, 19
- 代入, 24
- 蓄積型メディア\*, 7
- 知的財産, 42
- 著作権, 42
- 著作権ビジネス, 52
- 著作者人格権, 43
- 定性的\*, 55
- 定量的\*, 55
- 哲学\*, 40
- テラバイト\*, 8
- データ, 5
- データ処理, 50
- データセンター, 51
- データフロー図\*, 51
- データベース, 50
- データ量, 6
- デジタル, 7
- デジタル回路\*, 8
- デザイン, 15
- 電子商取引, 49
- 電子署名\*, 31
- 電子ブックリーダ\*, 19
- 電子メール, 34
- 伝達メディア, 7
- 特許権, 42
- トップレベルドメイン\*, 34
- トレードオフ, 41

- トロイの木馬, 32
- ドメイン名, 33
- 二進法, 10
- 2次記憶装置, 20
- 24ビットカラー, 14
- 二の補数\*, 11
- 日本工業規格\*, 13
- 入出力装置, 20
- ネットショップ, 49
- ネットワーク, 28
- ノード, 28
- ハードウェア, 21
- ハードウェアベンダー, 51
- 発散的思考, 55
- ハブ, 28
- バイト, 8
- 番地, 20
- パーソナルコンピュータ, 19
- パケット, 29
- パターンファイル\*, 33
- パッケージソフトウェア, 52
- パブリシティ権\*, 45
- 秘密鍵, 31
- 表計算, 21
- 表現メディア, 7
- ビジネス創造, 49
- ビット, 8
- 描画ソフト, 21
- ピクセル, 14
- ピクセル画像, 14
- 復号, 31
- 符合化, 12
- 符合なし整数\*, 11
- 符合ビット, 11
- フリーソフトウェア, 23
- フルカラー, 14
- フレーム, 15
- フレームレート, 15
- フローチャート, 24
- ブレーンストーミング, 56
- 分析, 55
- プレゼンテーション, 16
- プライバシー権, 44
- プログラミング言語, 22
- プログラム, 8
- プログラム内蔵\*, 21
- プロセス, 53
- プロトコル, 29
- プロトコル群\*, 30
- 変数, 24
- ベクタ画像, 14
- 法律, 42
- 防火壁, 32
- マークアップ, 35
- マスコミュニケーション, 7
- マスメディア, 7
- マルウェア\*, 32
- ミドルウェア\*, 22
- 無線 LAN, 28
- メールサーバ, 34
- メールソフト\*, 34
- 命令, 20
- メインメモリ, 20
- メガバイト\*, 8
- メタ認知, 5
- メディア, 6
- メディアリテラシー, 38
- 文字コード, 12
- 文字セット\*, 12
- 文字列, 27
- モラル, 40
- 問題, 53
- ユニバーサルアクセス, 38
- ライセンス, 23
- ラスト画像, 14
- 利便性, 48
- 量子化, 9
- 量子化誤差\*, 9
- リンク, 35
- 倫理学, 40

レジスタ, 20  
ローカルアドレス\*, 29  
ワードプロセッサ, 21  
  
検索サービス, 48  
  
主記憶, 20  
  
AD 変換, 9  
Android\*, 22  
ASCII, 13  
  
B2B, 49  
B2C, 49  
  
CODEC\*, 15  
CPU, 20  
CSS\*, 35  
  
DA 変換, 9  
DNS, 34  
  
EUC-JP, 13  
  
GIF\*, 14  
  
HTML, 35  
HTTP, 35  
  
IMAP\*, 34  
iOS\*, 22  
IP, 29  
IP アドレス, 29  
IP パケット\*, 29  
IP ヘッダ\*, 29  
IPv4, 29  
IPv6, 29  
iso-2022-jp, 13  
ISP\*, 28  
IT 産業, 51  
  
JavaScript, 26  
JPEG\*, 14  
KJ 法\*, 55  
LAN, 28  
Linux, 22  
Mac OS\*, 22  
MIDI\*, 14  
PDCA サイクル, 56  
PNG\*, 14  
POP\*, 34  
POS, 49  
Shift-JIS, 13  
SMTP\*, 34  
SSL, 35  
TCP\*, 30  
TCP/IP\*, 29  
UNICODE, 13  
URL, 35  
UTF-16\*, 13  
UTF-8, 13  
Web クリエータ, 51  
Web サーバ, 35  
Web ブラウザ, 35  
Web ページ, 34, 35  
Windows\*, 22  
WWW, 34