0. 学習の指針

学習の進め方

この授業ではパソコンを使って、コンピュータおよびコンピュータネットワークの仕組 みと基本的諸概念について学ぶ。情報やデータとは何かからはじまり、コンピュータで 情報を処理するとはどういうことか、情報の検索と発信、コンピュータの重要な応用ソ フトである表計算とデータベースの仕組み、などについて幅広く学んでいく。

授業では、予習・復習および実習を重視し、毎回、予習課題、授業内課題および宿 題に挑戦してもらう。これらの課題を解くに際し、手がかりはこの資料あるいは講義中 に与えられるが、十分な情報が与えられるとは限らない。

インターネットで参考情報を探す、友達同士で教えあう、教員やTAに聞く、などして、 課題を完成させて欲しい。この授業ではそのような体験およびその過程が最も重要だ と考えている。

課題を行なう際には、細かなことでつまずくことも多いと思われる。しかし、その体験 こそがコンピュータに対する真の理解を深め、コンピュータの表面的でない活用能力を つけることにつながる。単に教わった、あるいは、つまずくことなく獲得した知識や技能 は身につくことはないと思って頑張ってほしい。

学習のポイント!

以下を参考に、各自の情報源となるホームページを探し、それをブックマーク(お気に入り)に登録せよ。

1.総合検索サイト

最初の手掛かりはまず総合検索サイトだろう。以下に代表的な検索サイトへのリンクを上げておくので、気に入ったものを自分のブックマークに登録せよ。

- 1. Google
- 2. <u>Yahoo</u>
- 3. Infoseek
- 4. <u>goo</u>

注意!

ー橋大学で使っている学習管理システム(LMS, Learning Management System) WebClassの画面では、ブックマークの登録や印刷などの作業ができない。その 場合、次の手順で、ページを新しいウィンドウで開くとよい。

- 1. 画面上でマウスの右ボタンをクリック(右クリック)する。
- 現れたメニューで、「このフレーム」の「新しいウィンドウでフレームを開く」を クリックする。

2.その他役立つサイト

他にも、辞書、辞典、用語集、等々役立つWebページが多数ある。自分のインターネット上の情報源がどれだけ豊かにできるかが、これからの君たちの学修に大きく影響する。 以下の情報のあるWebページを探して、ブックマークに登録せよ。友人とも情報を交換して、自分にあったページをみつけるとよい。授業中に知らない術語で出会ったときの意味確認、代表的なアプリケーションの基本的な使い方などはインターネットで調べよう。

- 1. コンピュータ用語辞典
- 2. EXCEL入門
- 3. ACCESS入門
- 4. Webページの作り方

4.質問メールの宛先

山崎秀記 <u>cj00059@srv.cc.hit-u.ac.jp</u>

1. 情報とは何か

これから、主にコンピュータを用いた情報の処理および活用のさまざまについて学ぶが、最初に情報とは何かについておさらいしておこう。(参考:川合慧著、「情報」、東大出版会)

1. 情報とデータ

「情報」と「データ」とは、その区別にあいまいなところがあり、意味が必ずしも一定していないが、ここでは、一応次のように定めておく。

情報は「事柄の内容。様子、また、その知らせ」(日本国語大辞典)とあるように、処理・伝 達されるために何らかの意味を包含するものである。それに対し、データは情報を文字や 記号で表現した資料であり、一定の背景・文脈のもとで情報として解釈される。見方を変え れば、情報はそれ自身形を持たない抽象的な概念であり、情報を表現するデータが与えら れて、初めて形を持ち処理・伝達される。

データの解釈が可能になるためには、文脈、すなわち、データ表現に関する規約が必要 である。例えば、一方が日本語を話しているつもりで、他方が英語を聞いているつもりでい れば、当然意味は通じない(情報は伝達されない)。それほど極端な例でなくても、同じ表 現が両者で異なった解釈されることは、日常茶飯であり、注意が必要である。

さらに、データ(を表現する文字や記号)をコンピュータ処理が可能になるように0、1の列 で表現したものを**符号**という。

2. 情報の最小単位ビット(bit)

最も少なくかつゼロでない情報と言えば、2種類の値(Yes/No、ON/OFF、真/偽、左/右、 有/無、晴/雨、0/1、・・・)のどちらかを指定する情報であろう。情報のこのような最小単位 をビット(bit)という。ビットを2つ重ねた2ビットでは4種類の(例えば、

YesYes/YesNo/NoYes/NoNoで表される)情報が表現でき、3つ重ねた3ビットでは8種類の (例えば、000/001/010/011/100/101/110/111で表される)情報が表現できるというよう に、必要に応じてビットを重ねれば、何種類の情報でも、極言すれば、どのような情報でも 表現できる。

コンピュータ内部で物理的には、このビットを電流のON/OFFや磁化の有/無で実現しているが、通常はその違いを無視して単に0/1で表す。なお、画像、音声、文字などといったデータを具体的にどう符号化(ビット表現)するかについては、データ表現の章で詳しく説明する。

問.8ビットでは何種類の情報を表現(区別)できるか。

8ビットを1**バイト(Byte)という。半角の英数字は(8ビットで表現できる種類に収まるので)1** バイトで符号化され、漢字などの全角文字は 2バイトで符号化される。1ギガは10億なの で、20ギガバイトのハードディスクには計算上、最大10ギガ=百億全角文字分の情報を貯 えることができる。

問. 容量20ギガバイトのハードディスクに、日本語の書籍(例えば、約 1000文字 /ページ、 200ページ/冊)何冊分の情報を蓄えることができるか。

3. 対象のモデル化

ところで、われはわれは抽象的な対象であれ、具体的な対象であれ、対象をそのまままるごと認識できるわけではない。何らかのモデルを援用して対象を認識し、情報を抽出し、 データとして表現する。ここではその過程を3つのレベルに分けて整理しておこう。

1)概念モデル

対象を認識するさいに援用するモデル。言い換えると、人は何らかの概念モデルを用いて、対象の特定の側面を認識対象として抽出し、概念を形成する。概念モデルは各個人の、いわば頭の中に形成されるものであるから、同じ対象に対しても個人ごとに、あるいは TPOに応じて異なる。情報はこのレベルに対応すると考えられる。

2)論理モデル

対象(より正確にはその概念モデル)を言葉や文字、記号で表現するレベルのモデル。論 理的構造や関係が記号的に表現され、論理的処理や操作が可能になる。通常、われわれ は、このレベルの情報(データ)を操作し、互いに交換している。

3)物理モデル

物理的な表現のレベルのモデル。例えば、コンピュータ内部でのデータ表現(符号)がこれにあたり、機械的な操作を問題にするときは、このレベルのモデルが対象になる。

ただし、これら3つのレベルのモデルについても、その区分けが明確であるとは限らない。 例えば、この講義ではコンピュータ内部でのデータ表現(物理モデル)は通常0,1の列(場合 によっては文字列や10進数)を用い、その物理的実態である電流のON/OFFや磁化の有 無を議論することはない。しかし、例えば回路設計などの分野においては、0,1の列が論理 モデルで、電流のON/OFFが物理モデルであると考えられる。

4. モデルの性質

一般にモデルを構築する際、そのもととなる対象との関連が問題となる。

1) 完全性

モデルが、対象とするすべての要素に対して、モデル内に対応する要素(表現)があるとき、そのモデルは完全であるという。

2) 一意性

モデルの要素が表現する対象が一意に定まるモデルは一意的であるといい、そうでない モデルはあいまいであるという。例えば、各個人をその氏名で表現するモデルでは、同姓 同名の人がいた場合にあいまいになる。その場合は、学籍番号のように一意な表現を付 加したモデルが作られる。

3) 忠実性

ー意性の逆で、対象の1要素に対して、モデルの1要素(表現)が対応する場合、そのモ デルは**忠実**であるという。忠実でないモデルでは、対象のある要素が複数の表現を持つこ とになり、モデル内での操作は(どちらの要素を操作するか不明なため)複雑になる。した がって、忠実でないモデルにおいては、表現の標準化を行って、忠実なモデルに変換する ことが行われる。

4) 整合性

モデルが表現する対象の世界では通常、各要素が孤立して存在するのではなく、要素間の何らかの関係や操作(演算)が定義され、その関係や演算が満たしている規則が存在する。このような関係や操作がモデルの要素(表現)の間でも定義され、対象の世界で成り立つ規則がモデルの世界でも成り立つとき、モデルは**整合的**であるという。

5) 冗長性

本質的に同じ情報が表現中の複数箇所に現れたり、本質的でない表現が含まれるモデ ル(表現)は冗長であるという。通常われわれが用いる表現(言語、絵画等)は冗長であり、 それゆえその一部が欠落したとしても、情報が正しく伝達されることが多い。それに対して、 冗長でない表現(データ)は、表現が短く効率的であるが、一部でも欠落すると情報が正しく 伝わらない。また、例えば答案用紙に学籍番号と氏名の両方を記入させる場合のように、 誤りを防ぐ、あるいは誤りがあっても情報が正しく伝わるように、冗長性を利用することがあ る。

問. 空欄を埋めよ

実数は原理的に無限の桁数を持つことから、すべての実数に対して、対応する表現を構成することは不可能である。したがって、実数の論理モデルや物理モデルは本質的に 【 】である。

有理数の分数表示は【 】でない(-1/2=2/(-4)=…)ので、(分母が正の)既約分数表示に標準化することによって、【 】なモデルが構成される。

通常コンピュータでは実数を有限桁数で表現している。仮に、(何桁でも同様だが)小数 点以下4桁で四捨五入するとしよう。その場合、0.3325以上0.3335未満の数はすべて0.333 と表現されるので【 】ではなく、それゆえ実数の世界では成り立つ規則1/3×3=1が、モ デルの世界では成り立たず(1/3×3=0.999≠1)【 】ではない。

このように、コンピュータで処理されるデータモデルは、もとの対象を正しく表現しているとは限らないので、注意が必要である。

5. データモデル

対象の論理モデルを構築する際には、対象に対する深い理解のもとに適切な表現を選択する必要がある。ここでは、よく用いられるモデルの代表的なタイプを紹介しておこう。

1) グラフモデル

対象を〇や口で、対象の間の関係を実線や矢印で、図式的に表現したモデルをグラフモ デル、あるいはネットワークモデルという。例えば路線図や会社の組織図など、グラフモデ ルは、実生活の様々な場面で使われている。インターネット上のWebページとその間のリン クや、ゲームの局面と(指し手による)その間の移動なども、グラフモデルでモデル化でき る。

例えば<u>東京メトロ路線図</u>や<u>一橋大学の機構図</u>を見てみよう。

問. Stanley Milgramによれば、全世界の人間を対象にした知人ネットワークグラフで、人間の間の距離(到達に要する実線の数の最小値)の最大値はいくつか。

2) 階層モデル

特に対象の分類(クラス分け)とその間の所属関係を表現したグラフモデルを**階層モデル**という。生物の分類図や図書の分類、住所の階層構造などのように、実生活でもよく使われる。通常、根元(出発点)となる一つの要素から、次々に枝分かれしていくことから、木構造とも呼ばれる。

局面が循環することのないゲームは初期局面を出発点とする木構造でモデル化できる (ゲーム木と呼ばれる)ことからもわかるように、階層モデルは、出発点を持ち、矢印や実線 の列が閉じて(循環して)いないグラフモデルである。

コンピュータのファイルシステムは、ドライブ(記憶装置)を出発点とする階層モデルである (エクスプローラを起動して、表示→エクスプローラバー→フォルダ、を表示させてみよ)。さらに、条件判断を重ねて行われる意思決定の全体をモデル化した木構造は、決定木と呼ばれる。

ヘッケルの系統樹 (<u>http://ja.wikipedia.org/wiki/</u> <u>%E7%B3%BB%E7%B5%B1%E6%</u> <u>A8%B9</u>)



3) 関係モデル

例えば、(学籍番号、名前、住所、連絡先)などのように、いくつかの項目の値の組みを集めたものを関係(表)という。情報を関係(表)の集まりで表現するモデルを関係モデルという。関係モデルについては、データベースの章で詳しく議論する。

課題.次回の授業開始時まで

- 1. 本文中の問について、WebClassのテスト/アンケートから解答せよ。
- 2. 次回の予習課題に答えよ。

2 コンピュータの構成

この章のポイント

――パソコンのカタログを読む――

CPU:2 ギガヘルツ、 メモリ:512 メガバイト、 ハードディスク:120ギガバイト、

パソコンのカタログに並んでいるこれらの数値は何を意味するのだろうか。 この章では、コンピュータの大まかな構成や動作原理を理解し、これらの意味を理 解できるようにしよう。

この章の構成

2.1 コンピュータの構成

<u>2.2 記憶装置の階層</u>

<u>2.3 キャッシュ(cache)</u>

2.4 カタログのその他の情報

ー橋大学 情報リテラシー

2. コンピュータの構成

2.1コンピュータの構成 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報 2.1 コンピュータの構成



コンピュータは通常、CPU、主記 憶装置、二次記憶装置、入力装 置、出力装置などから構成され る。



予習課題1.

インターネットで、適当なパソコンのカタログを表示させ、CPUのクロック周波数、メモリーの 容量、ハードディスクの容量を調べててみよう。あるいは、現在自分が使っているパソコン に関して、

[スタート]→[コントロールパネル]→[システム] を開いて、CPUの速度やメモリの大きさを確認せよ。

予習課題2.次の言葉の意味を調べよ。

1) ヘルツ、ビット、バイト

2)キロ、メガ、ギガ、テラ、ミリ、マイクロ、ナノ、ピコ、

3)2)の数詞の由来(例えばテラは「巨大な怪物」を意味するギリシア語)を調べよ。

1. CPU(Central Processing Unit、中央演算処理装置)

| コンピュータの頭脳部で、計算やその制御を つかさどる。CPU内の各部品は同期をとって (タイミングを合わせて)動作するが、その同期 頻度を クロック周波数 といい、例えば2GHz (ギガヘルツ)であれば、毎秒2ギガ回の同期 がとられる。同種のCPUであれば、この数値 が大きいほど、処理速度が速くなるが、異なる 種類のCPUの比較ではクロック周波数の大き い方が速いとは限らない。 また、CPUの能力を、flops(FLoating-point Operations Per Ssecond)という1秒あたりの 実数(浮動小数点)演算回数で計ることもあ る。手元のパソコンのflops値を計測してみたけ れば、理化学研究所情報基盤センターの <u>姫野</u> |
|--|
| |

Ind. P

<u>ベンチマーク</u>(最小サイズのもの)を利用すると よい。 また、CPUが一度に処理するデータのサイズ (2進桁数)によって、32ビットマシーンまたは 64ビットマシーンという言い方をする。

2. 主記憶装置

主記憶装置(主メモリ)は一定の大きさの区画に分かれていて、例えば下図のように、区画ごとに(2進数の)番地が割り振られ、0と1の列(1バイト)が格納される。

| 番地 | 内容 |
|---------|----------|
| 0000000 | 0000000 |
| 0000001 | 00010010 |
| 0000010 | 00000110 |
| | ••• |

主メモリは、任意の番地に対して直接読み書きができることから、RAM (Random Access Memory)と呼ばれる。RAMには高速で高価な (つまり多量に使えない) SRAMと比較的安価 (で低速)な DRAM があり、主メモリには、DRAMが使われる。 また、メモリ内のデータは電源が切れると失われる (揮発性)。

3. 二次記憶装置

ハードディスクが代表的。他に、フロッピーディスク、DVD-ROM、DVD-RAM(読み 書き可能なDVD)等の外部記憶装置も含まれる。これらの記憶装置は電源を切って も、その内容が失われない(**不揮発性**)。 通常、コンピュータが計算を行なうためのプログラムやデータは二次記憶装置に貯 えられていて、一度主記憶に移された(load)後、実行される。実行結果等必要なデ ータは再び 2次記憶装置に保存 (save)しておかなければ、電源のOFFとともに失 われる。

4. 入力装置

キーボード、マウス、スキャナー、タブレット、ディジタイザー、等。入力装置は入力されたデータを0,1の列に変換して、コンピュータに伝える。

5. 出力装置

ディスプレイ、プリンタ、プロッター、等。出力装置は、データ(0,1の列)を文字や音声、画像などの人間にわかりやすい表現に変換して表示する。

メモリ)の略。

コラム コンピュータで扱われる数詞

英語の数詞は3桁(10³)ごとにキロ(kilo-)、メガ(mega-)、ギガ(giga-)、テラ (tera-)と呼ばれる。日本語の数詞は4桁(10⁴)ごとだから、キロは千、メガは百 万、ギガは十億、テラは一兆にあたる。小数点以下の数詞は 10⁻³ごとに、ミリ (mili-)、マイクロ(miclo-)、ナノ(nano-)、ピコ(pico-)という。

2.1コンピュータの構成 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報

2. コンピュータの構成

<u>2.1コンピュータの構成</u> 2.2 記憶装置の階層 <u>2.3 キャッシュ(cache)</u> <u>2.4 他のカタログ情報</u> 2.2 記憶装置の階層

CPUも中にレジスタと呼ばれる記憶装置を持つ。これらの記憶装置のおおよその動作 速度と容量、値段を次の表にまとめた。

| | 動作速度 | 容量 | 単価 (2010年) |
|-----------|--------|---------|----------------|
| レジスタ(CPU) | ~ 1ナノ秒 | 十数バイト | |
| SRAM | 数ナノ秒 | 数メガバイト | 2メガバイト/2000円 |
| DRAM | 数十ナノ秒 | 数ギガバイト | 2ギガバイト/3000円 |
| ハードディスク | 数ミリ秒 | 数百ギガバイト | 500ギガバイト/5000円 |

問2. ムーアの法則について調べよ。

速度差が何を意味するか考えてみよう。

次回詳しく学ぶが、コンピュータ(CPU)の基本動作サイクルは、

①メモリ(DRAM)からデータを取り出し(読み込み) ②何らかの処理を行い ③メモリにデータを格納する(書き出す)

ことからなる。

ところが、速度差があるので、DRAMに対して読み書きする間に、CPUは数十回の動作 が行なえる。また、ハードディスクに読み書きする間には、CPUは数百万回動作できる し、人間の反応速度は速くても0.1秒程度だろうから、例えば連続したキー入力の合間に でさえ、CPUは約 1億回の動作が行なえることになる。つまり何らかの工夫をしないと、 CPUはそのほとんどの時間、他の装置からの応答を待って遊んでいるだけということにな る。

ー方、ハードディスクなどの動作速度が遅いのは、データの場所を探してそこに読み取 りヘッドを移動させる時間がかかることにある。一度ヘッドが移動してしまえば、その付近 のデータを連続的に読み書きするのは、それほど時間がかからない。 それゆえ、次節で説明する**キャッシュ**という工夫がなされる。

2.1コンピュータの構成 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報

2. コンピュータの構成

<u>2.1コンピュータの構成</u> 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報 2.3 キャッシュ(cache)と仮想記憶

低速記憶装置を高速記憶装置にみせるキャッシュ 小容量記憶装置を大容量に見せる仮想記憶

主メモリ(DRAM) へのアクセスとCPU との速度差を埋めるために工夫されたの が、より高速なSRAMを使ったキャッシュメモリである。CPU が主メモリのデータにア クセス(読み書き) したとき、その付近(ブロック単位)のデータをまとめてキャッシュメ モリに移しておき、以降、その付近のデータを読み書きするときには、キャッシュメモ リにアクセスする。(そして、最後にキャッシュメモリの内容を主メモリに書き出す。) CPUは近くにあるデータに連続してアクセスすることが多いことから、高速化が図ら れる。

したがって、キャッシュメモリの容量が大きいほど、CPUがメモリアクセスに待たされることが少なくなり、コンピュータは高速に動作する。

また、ディスクと主メモリの速度差を埋めるために主メモリの一部をディスク代わり にするディスクキャッシュが使われ、インターネットとハードディスクの速度差を埋め るためにホームページをハードディスクに一次的に貯えておくキャッシュも行われて いる。

キャッシュがいっぱいになると、新しいデータを置くためには、古い (使われそうもない)デ ータを追い出して、主メモリーなど本来の場所に戻す必要がある。どのデータから追い出す かの判断基準 (戦略)としては、次の二つが知られている。 1.FIFO(First In First Out)

最初に入れた(First In)、つまり最も古いデータから追い出す(First Out)。これは、レジや銀行窓口などにおける行列と同じだから考えやすいだろう。

2.LRU(Least Recently Used)

最後に使用(アクセス)された時間が最も古い(Least Recently Used)データを追い出 す。これを実現するためには、アクセスするたびにそれを行列の最後にまわし、先頭 のデータから追い出せばよい。これを文書の整理法として提唱したのが、野口悠紀 雄著『「超」整理法』である。

キャッシュは低速(大容量)の記憶装置中のデータを必要に応じて高速(小容量)の記憶 装置に移しておくことによって高速化を図る技術であり、いわば低速の記憶装置中のデー タを、高速の記憶装置上にあるかのように見せる技術である。これは見方を変えれば、小 容量(高速)の記憶装置の記憶空間を、大容量(低速)装置上に広げる技術でもある。この ような考え方で、主メモリーに入りきらないような大容量データの計算を、ハードディスクを 活用して可能にする技術が仮想記憶である。

問.スワップ領域について調べてみよ。

コラム ブラウザの更新ボタン

キャッシュのところで述べたように、ブラウザは一度表示したWebページをハー ドディスクに保存しておき、再度表示要求があったとき、保存しておいたページを 表示する。したがって、ニュースのなど頻繁に更新されるページでは、古いペー ジが表示されることがある。このときは、ブラウザの更新(再読み込み)ボタンをク

リックするとよい。

<u>2.1コンピュータの構成</u> 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報

2. コンピュータの構成

<u>2.1コンピュータの構成</u> 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報 2.4 他のカタログ情報

1.ビデオメモリ

ディスプレイに表示される内容を保持しているメモリ。CPUはビデオメモリに表示内容 を書き込み、ディスプレイへの表示作業は、専用のビデオカード(グラフィックス・アクセ ラレータ)が受け持つ。

2.表示モード

画面の横と縦の**ドット**(画面を構成する点、ピクセル)数と、ドットごとに設定可能な 色数で指定される。色数は通常約1677万色で、光の3原色(赤緑青、RGB)がそれぞ れ2⁸=256段階の明るさで指定される。

3.システムバス

システムバスはCPU とメモリをつなぐ伝送路で、バス幅(通常32ビット)に対応する本数の信号線を束ね、同期をとって伝送する。バスクロックが500MHz ならば、メモリへの読み書き動作の同期が毎秒5億回とられることを意味する。

コラム 2進数の概算

2¹⁰=1024 ≒10³を覚えておくと、2進数の大体の見積もりができる。例えば ディスプレイへの表示色数では、3原色の1色当たり2⁸通りであるから、全体で

2²⁴=2⁴×2¹⁰×2¹⁰=16×1000×1000=1600万

ということになる(正しくは 16777216色)。

また、コンピュータ関係では2¹⁰=1000として、キロ(=2¹⁰)、メガ(= 2²⁰)、ギガ(=2³⁰)ということも多い。

学習がすんだら小テストを行い、学習事項を確認しよう。

<u>2.1コンピュータの構成</u> 2.2 記憶装置の階層 2.3 キャッシュ(cache) 2.4 他のカタログ情報

3 コンピュータの仕組み

この章のポイント

――万能性と微小化の仕組み――

コンピュータはプログラムによって、ワープロに、メール送受信に、画像作成に、 様々な機能を果たす。コンピュータはプログラムを実行する機械であることによっ て、その万能性を獲得している。また、その大きさはどんどん小さくなり、携帯電話 をはじめとして、家電やその他多くのものに搭載されている。コンピュータはなぜそ のように小さくできるのか。その原理を探る。

この章の構成

<u>3.1 CPUの動作原理:プログラム内蔵方式</u>

- 3.2 論理回路
- 3.3 集積回路

3. コンピュータの動作原理

3.1 CPUの動作 3.2 論理回路 3.3 集積回路

3.1 CPUの動作(プログラム内臓方式)

コンピュータ(つまり、CPU)はメモリーに貯えたプログラム(命令の列)を逐次実行する。最初にこ ンピュータを設計したノイマン(Von Neumann)に因んで、ノイマン型プログラム内蔵方式と呼ばれ、 の基本原理は変わっていない。その手順簡単に説明すると以下のようになる。

- 1. 命令カウンタが示す番地(アドレス)をアドレス選択回路に送る。
- アドレス選択回路が見つけたメモリーの内容(つまり、命令)を、読み書き回路が命令レジ: る。
- 3. 命令レジスタ中の命令が命令デコーダに送られ、命令カウンタの値が1(命令が占める領域 やされる。
- 4. 命令デコーダが命令を解読し、演算回路等が必要な処理を行う。例えば、
- 「LOAD GR, 1000」(1000番地の内容を汎用レジスタGRに取り出せ)なら、 番地(1000)をアドレス選択回路に送り、読み書き回路を通して、 その内容を汎用レジスタに入れる。
 「ADD GR, 1001」(1001番地の内容を汎用レジスタGRに加えよ)なら、 番地(1001)をアドレス選択回路に送り、読み書き回路を通して、 その内容を取り出し、演算回路を使って汎用レジスタに加える。
 「STORE GR, 1002」(汎用レジスタGRの内容を1002番地に格納せよ)なら、 番地(1002)をアドレス選択回路に送り、読み書き回路を通して、 汎用レジスタの内容をメモリーに格納する。
 「JUMP0 GR 100」(GRの内容が 0 なら100番地に飛べ)なら、 汎用レジスタの値を調べて、0なら番地(100)を命令カウンタに入れる。
 5、 "1"に戻って繰り返し。



命令の種類

CPUが実行する基本的な命令には、

- 1. メモリーからのデータの取り出し(LOAD)命令
- 2. メモリーへのデータの格納(STORE)命令
- 3. 四則などの演算命令やデータの大小の比較命令
- 4. 次に実行する命令(の番地)を指定する分岐命令
- 5. IN, OUTなどの入出力命令

などがある。他にも種々あるが、詳しくはCASL(基本情報技術者試験で使用されるアセンブリ言語) を調べてみよ(CASLII入門講座を参照せよ)。

コラム 機械語とアセンブラ

CPUが直接理解(実行)できる形のプログラム(0,1の列)を機械語プログラムといい、それを ような形で(人間に分かりやすいように)記号で表現したものをアセンブラという。機械語の命 アセンブラの命令は基本的に一対一に対応する。

それを、数式で表現するなど、さらに人間に分かりやすい形でプログラムを書くことを可能 のがFORTRAN、COBOL、LISP、PASCAL、C、JAVAなどの高級言語である。これらの高級言 は機械語に翻訳(コンパイル)された後、コンピュータで実行される。

コラム 処理の逐次性と局所性

コンピュータは高速かつ大容量であることを見てきた。しかし、コンピュータは大量のデータ 瞬のうちに処理でき(るように見え)ても、それらを同時並列的に処理できるわけではない。ニ ュータ(CPU)は、前節で見たように比較的な単純な命令を逐次的に実行し、しかも一度に扱 データの大きさは通常32ビットである。これは画像で言えば、一つの点の持つ色の情報のナ (24ビット)程度に過ぎない。

例えば、人は画像全体を見渡して、「空の絵である」とか「青っぽい」などという判断をする。 し、コンピュータが「青っぽい」と判断するためには、画像のすべての点を一つ一つ調べて、 ↓ 点が何パーセント以上含まれるか、といった計算をする必要がある。また一般的に、与えら なが「空の絵」であるといつも正しく判断するようなことはまだコンピュータにはできていない。 よく「木を見て森を見ず」と言うが、まさに、コンピュータは「森を見るにも、一つ一つの木を」 としかできない」。コンピュータ処理の、このような逐次性と局所性を理解しておくことは、これ コンピュータと付き合っていく上で重要なポイントである。これについては、梅津信幸著「あな コンピュータを理解していますか」(技術評論社)を薦める。

3.1 CPUの動作 3.2 論理回路 3.3 集積回路

3. コンピュータの動作原理

<u>3.1 CPUの動作</u> 3.2 論理回路 3.3 集積回路

3.2 論理回路

CPUでの演算は、足し算にせよ、他の演算にせよ、結局のところ、与えられた複数の 32ビット入力列から32ビット出力列を求めることに他ならない。ここでは、それが AND と OR と NOT という3種類の素子からすべて構成できることをみてゆこう。

1 AND、OR、NOT の真理値表

AND、OR、NOT の素子は以下のような真理値表を持つ。ここで 0⇔偽、1⇔真と対応 させたとき、それぞれ、NOT(でない)は真偽を反転、AND(かつ)は入力がともに1(真) のときのみ1(真)、OR(または)は入力のどちらかが1(真)のときに真 になるという働き をもつ。

| NOT(¬) | の真理値表 | $AND(\Lambda)$ |)の真理値表 | OR(V) | の真理値表 | |
|--------|-------|----------------|--------|-------|-------|--|
| Х | ¬x | ХY | ХЛҮ | ΧY | ХVY | |
| 0 | 1 | 0 0 | 0 | 0 0 | 0 | |
| 1 | 0 | 01 | 0 | 01 | 1 | |
| | | 10 | 0 | 10 | 1 | |
| | | 11 | 1 | 11 | 1 | |
| | | | | | | |

2 双対性

AND と OR の真理値表を比べてみると、AND の真理値表で 0 と 1 を置き換えた ものが OR の真理値表になり、逆に ORの真理値表で 0 と 1 を置き換えると AND の真理値表が得られることがわかる。そして、NOTの真理値表は0と1を置き換えて も変わらない。そこで、AND、OR、NOTから構成される論理式に対して、0 と 1、人と Vを交換した論理式を元の論理式と双対な論理式 と呼ぶ。このとき、2つの論理式 が等しければ、それらに双対な論理式同士も等しい。これを論理式の双対性とい う。

> 例 $X \land \neg X = 0 \Leftrightarrow X \lor \neg X = 1$ $X \lor (X \land Y) = X \Leftrightarrow X \land (X \lor Y) = X$

3 与えられた真理値表を持つ論理式(論理回路)の設計

さて、コンピュータのCPUでは、四則演算やその他様々な演算が行なわれている が、それがどのような原理で実現されているか見てゆこう。前にも述べたように、 CPUでの演算は1個または、2個の固定長(32 ビット)の0, 1の列から同じ長さの0、 1の列を計算する。 では、AND、OR、NOTの基本素子を使って、任意の演算を行なう回路を組み立て るにはどうすればよいだろうか。まず、まず組み立てたい回路の真理値表(入出力 関係を定める表)を作成する。

例えば、以下の真理値表を持つ演算(関数)Fを計算する回路を作る手順は以下 の通りである。簡単のために、入出力ともに1ビットで2変数や3変数の場合を考えて いるが、ビット数や変数が増えても同様の考えかたができる。

与えられた真理値表を持つ論理式(論理回路)の設計:2変数の例

- 真理値表で値が1の行に対し、その行でだけ1で他はすべて0である真理 値表を持つ論理式を求める。そのために、X=0を意味する(真偽が一致する) 論理式が「Xであり、X=1を意味する論理式がXであることに注意しよう。した がって、求める論理式は得るには、考えている行において、 入力値が0である変数にはNOT(「)を付けた上で、すべての変数をAND (A)で結ぶ。
- 2. 与えられた真理値表を持つ論理式を得るには、上で求めた**論理式をOR(V)** で結ぶ。

X Y F(X,Y)の値 この行だけ1で他は0の真理値表を持つ論理式

- 00 0
- 0 1 1 ¬X∧Y
- 10 1 X∧¬Y
- 11 0

 $F(X,Y) = (\neg X \land Y) \lor (X \land \neg Y)$

さらに、上で求めた論理式を実現する論理回路は以下のようになる。



与えられた真理値表を持つ論理式(論理回路)の設計:3変数の例

| ΧΥΖ | F(X,Y,Z)の値 | この行だけ1で他は0の真理値表を持つ論理 式 |
|-----|------------|-------------------------------|
| 000 | 0 | |
| 001 | 1 | $\neg X \land \neg Y \land Z$ |
| 010 | 1 | $\neg X \land Y \land \neg Z$ |
| 011 | 0 | |
| 100 | 1 | X∧¬Y∧¬Z |
| 101 | 0 | |
| | | |

$$1 1 0 0$$

$$1 1 1 1 X \land Y \land Z$$

$$F(X,Y,Z) = (\neg X \land \neg Y \land Z) \lor (\neg X \land Y \land \neg Z) \lor (X \land \neg Y \land \neg Z) \lor (X$$

$$\land Y \land Z)$$

4 半加算器:1ビットの X と Y の加算回路

具体的な例として、1ビットの加算を行なう回路を考えてみよう。加算の結果は、桁 上がりを考慮すると2ビットになるので、上下の桁それぞれの回路を構成して組み 合わせる。次の表からわかるように、下の桁の回路は、上で作成した2変数の例に 他ならず、上の桁は、入力がともに 1のときのみ 1 であるから、AND そのものであ る。 XΥ 上の桁 下の桁 00 0 0 0 01 1 10 0 1 11 0 1 χ-下の桁 Y 上の桁 このように、コンピュータの演算回路はAND(Λ)とOR(V)とNOT(\neg)の 素子があれば組み立てられることがわかる。

<u>3.1 CPUの動作</u> 3.2 論理回路 3.3 集積回路

3. コンピュータの動作原理

3.1 CPUの動作 3.2 論理回路 3.3 集積回路

3.3 集積回路

前節では、すべての論理回路がAND、OR、NOTの3種類の素子から組み立てられることを見てきた。では、コンピュータのCPUである集積回路では、これら3種類の素子を多量にかつ複雑に配置してつないでいるのだろうか。実はこれらの素子は、1種類の仕組み(素子)から組み立てることができるのである。

CPUでは、0、1は電流のON(1)とOFF(0)で表される。そこで、一方の電流のON、 OFFによって、もう一方の電流のON、OFFを制御する仕組みがあれば、任意の論理 回路が構成できることを見てゆこう。歴史的には、このような仕組みとして、リレース イッチ、真空管、トランジスタなどが使われてきた。

下図で、Xに電圧がかかる(X=1)のとき、Y と Zが遮断され、X=0のとき、Y と Z が つながって電流が流れることができるとしよう。



Yに電圧をかける(Y=1にする)と、Zは X=1のとき 0 で、X=0 のとき 1 だから、下 図の素子はNOT の働きをする。



さらにこの素子を直列につないで、上の一につなぐと、以下のOR素子 ができる。



また、並列につないで、その出力を一につなぐと、AND素子になる。



このように、1種類の素子のつなぎ方だけを変えることで、AND、OR、NOT の素子が

実現でき、ひいては任意の論理回路が実現できる。したがってきわめて単純化して 言えば、この素子を多量に格子状に並べておいて、その間の結線を工夫すれば、原 理的にはCPUを実現できることになる。これを一枚の基板上にその配線とともに直接 作りこもうというのが**集積回路のアイディア**である。集積回路では格子のメッシュを細 かくすればするほど、その集積度を上げることができる。



<u>3.1 CPUの動作</u> 3.2 論理回路 3.3 集積回路

4 OSの働き

この章のポイント

パソコンの電源を入れたとき、ワープロや表計算、インターネットブラウザなどのソフトウェア(プログラム)が直接動くわけではなく、通常ウィンドウシステムが立ち上がり、その中で目的や必要に応じて、適切なソフトウェアやデータファイルを選択して、起動する。

このように、われわれがパソコンを使うときには、OSと呼ばれる基本ソフトウェア 上でワープロなどの応用ソフトウェアを起動して利用するので、まずOSに対する基 本的な知識が必要である。

OS(Operating System、オペレーティングシステム)はコンピュータをより使いやす く、より効率的に使えるようにするための基本ソフトウェアであり、本章ではOSの学 習を通して、パソコンの代表的なOSであるWindowsの操作の基本的な概念を学 ぶ。

この章の構成

<u>4.1 OSの誕生</u>

<u>4.2 OSの役割</u>

<u>4.3 Windowsの基本</u>

4. Operating System の働き

4.1 OSの歴史 <u>4.2 OSの役割</u> <u>4.3 Windowsの基本</u>

4.1 OSの歴史

OS(Operating System)の歴史は、コンピュータを如何に人が使いやすいもの変えていくか、という工夫の歴史である。

- 1. その起源は、プログラムごとに毎回同じような入出力部分を組み込むことが面倒 なので、それをサブプログラム (サブルーチン)として独立させ、共通に使えるようにするところにあった。
- プログラムを機械が直接認識できる 0、1の機械語でなく、それを人間に分かり やすい表記に置き換えたアセンブリ言語で書けるようにすることが行なわれた。 アセンブリ言語で書かれたプログラムを機械語に翻訳するプログラムをアセンブ ラという。
- 次に考えられたのが、プログラム中で数式が使えるようにならないか、ということだった。これが高級言語と呼ばれるプログラミング言語の始まりで、科学技術計算用のFORTRAN (FORmula TRANslator)がその最初である。続いて英語に近い表現で、理解しやすく書きやすい言語を目指して、事務処理用のCOBOL (COmmon Business Oriented Language)が開発された。
- 4. コンピュータの性能が上がるにしたがって、高価なコンピュータに対し、CPUを無駄に遊ばせることなくいかに効率よく使うかが問題になってきた。
 - マルチタスク・マルチプログラミング:あるプログラム(タスク)が入出力処理 を実行している間に、他のプログラム(タスク)に CPUを割り当てるなどし て、複数のプログラムを(見かけ上)同時並行的に動かすようになる。ここ で、タスクとはOSから見た処理の実行単位である。
 - 2. TSS(Time Shearing System):1台のコンピュータに多くの端末をつないで 複数の人が同時に利用できるように、コンピュータが細かく区切られた時 間間隔で端末からの要求を処理したり、プログラムを実行する。多数の利 用者が(見かけ上)同時に1つのコンピュータを利用できる。
 - マルチユーザ管理:多くのユーザが1つのコンピュータ(システム)を共用 するために、ユーザの認証を行う、ユーザごとにハードディスクの占有領 域を確保する、ユーザごとのアクセス権限を設定するなどのユーザ管理を 行う。
- コンピュータ同士を回線で結び、遠隔地の(高性能)コンピュータを、回線を通して別のコンピュータの端末から使うようになった。このコンピュータネットワークを利用して、ファイルや電子メールのやりとりをするようになったのが、現在のインターネットの始まりである。
- 一方、人間とコンピュータとの対話処理を円滑に進める仕組みや工夫を ユーザ インターフェースという。最近は、GUI(Graphical User Interface)があたりまえに なってきており、コンピュータ内の様々な対象(装置、プログラム、データ等)に ア イコンという視覚的表現が与えられ、マウスなどを用いた視覚的・感覚的操作に より、コンピュータ操作が行なえるようになっている。

現在、パソコンのOSとしては、Windows、MacOS、Linux などが知られている。 4.1 OSの歴史 <u>4.2 OSの役割</u> <u>4.3 Windowsの基本</u>

4. Operating System の働き

<u>4.1 OSの歴史</u> 4.2 OSの役割 <u>4.3 Windowsの基本</u>

4.2 OSの役割

OSの歴史でも述べたように、OSの役割には以下のものがある。

1. 入出力装置の制御

入出力装置を利用するためのドライバを組み込んで、これらの装置を使用できるようにする。

- プログラム(タスク)の実行管理 実行中のプログラムの状態や入出力装置、記憶装置などの使用状況を管理して、プログラムが効率よく実行されるようにする。
- 3. **障害の検知と回復** コンピュータの信頼性と安全性を確保するために、障害をいち早く検知し、被害を最小限に とどめて、システムを回復させる。
- 記憶管理 キャッシュの機能を実現して、データの読み書きが効率よく行なわれるようにする。また、 逆に2次記憶装置を利用して主メモリのサイズを超えたデータをプログラムで扱えるように 仮想記憶の機能を実現する。
- 5. **ユーザ管理** 複数のユーザで共同で利用するマルチユーザシステムでは、ユーザ管理のために、ユー **ザ認証**(loginとパスワード)やハードディスク上の各ユーザの占有領域の管理を行なう。
- 6. 通信管理 コンピュータ間で通信を行なう機能を実現し、インターネットなどが使えるようにする。
- 7. 言語プロセッサー 高級言語(FORTRAN、COBOL、BASIC、PASCAL、C、JAVA、等)で書かれたプログラムを 機械が直接実行可能な機械語に翻訳し実行する。
- 8. **ユーザインターフェース** GUI 等のユーザが使いやすいインターフェースを実現する。

コラム. ターンアラウンドタイムとレスポンスタイム

コンピュータに指示を与えたとき、その指示に対して返答を開始するまでの時間をレスポンス タイムといい、処理を終了して結果を出力するまでの時間をターンアラウンドタイムという。 ウインドウズの例でいえば、砂時計(プログレスバー)が現れるまでの時間がレスポンスタイム で、砂時計(プログレスバー)が消えるまでの時間がターンアラウンドタイムである。 コンピュータの処理能力を向上させてターンアラウンドタイムを短縮させることが重要なのは 言うまでもないが、ユーザインターフェースの面からはレスポンスタイムの方がより重要であろ

<u>4.1 OSの歴史</u> 4.2 OSの役割 4.3 Windowsの基本

う。

4. Operating System の働き

<u>4.1 OSの歴史</u> <u>4.2 OSの役割</u> 4.3 Windowsの基本

4.3 Windowsの基本

ここでは、パソコンの代表的なOSであるWindowsの基本的な操作概念を理解しよう。 1.マルチウィンドウ

WindowsのようなGUIでは、複数のアプリケーションを同時に起動できる。アプリケーショ ンごとに対応するウィンドウが開き、そのウィンドウを通してアプリケーションを対話的に操 作できる。複数のアプリケーションを同時に起動できても、ウィンドウを通して対話的に操 作できるのは1つのアプリケーション(ウィンドウ)だけであり、そのウィンドウをアクティブウ ィンドウという。ここでは、アクティブウィンドウの切り替えを行ってみよう。 実習1. マルチウィンドウの理解を深めるために、以下の操作を行な え。 (1) 複数のアプリケーションを起動せよ。 アプリケーションの「起動」には、 1. 基本画面上のアプリケーション(のアイコン)をダブルクリックす る方法の他に、 2. 画面左下の「スタートボタン」(のクリック)から「プログラム」内の アプリケーションを クリックする方法がある。 (2) ウィンドウの辺や角をマウスでドラッグして、サイズを変更せよ。 (3) 一方のウィンドウでマウスドラッグで移動させて、他方のウィンドウ を隠せ。 (4)隠された方のウィンドウをアクティブにして、最前面に出せ。 操作法:画面一番下にある**タスクバー**上で対応するボタンをクリッ クする。

2.オブジェクト

コンピュータにつながっている入出力装置やプログラム、データ等のファイル、ウィンドウな どコンピュータ内の対象(もの)は、一般にオブジェクトと呼ばれる。オブジェクトには、アイコ ンという視覚表現が与えられ、マウスなどによる視覚的な操作が可能である。

オブジェクトは、プロパティ(属性)とメソッド(受け入れ可能な命令、操作)を持つ。マウスポ インタでアイコンを指し、マウスの右ボタンをクリックすると代表的なメソッドが表示されるの で、その中から希望の操作を選ぶことができる。またオブジェクトの属性は、やはり右ボタン のクリックで表示されるメニュー中のプロパティ欄を選択すれば、表示される。

実習2. マウスの右ボタン操作による、オブジェクト(アイコン)のコピー操作

以下の操作手順にしたがって、アイコンのコピー操作を行なえ。この操作はアイコンに限 らずコンピュータ内のすべてのオブジェクトに共通で、ファイルのコピーや文章や画像、音 声データのコピーにも応用できる。

(1) 画面(デスクトップ)上のアイコンにマウスポインタを合わせて右ボタンをクリックする。 (2) 現れたメニュ中にコピーがあればそれを選択する。 (3)次にコピーを置きたい場所(この場合はデスクトップ)で右ボタンをクリックし、貼り付けを選ぶ。
 (4)(3)で作成したコピーを、上記手順を参考に、削除せよ。

このように、現在のグラフィカルなコンピュータ操作は、もの(オブジェクト)を中心として、 1. 何を(対象、目的語) 2. どうする(操作、動詞) の順に(マウス動作等で)指定するのが基本である。これは日本語の語順と一致しているこ とに注意しよう。これに対し、以前の、命令をキーボードから入力するタイプのCUI(Character User Interface)では、英語ベースの命令を使って、操作(動詞)、対象(目的語)の順に指定 する。

3.マルチタスク

Windows では実行されているプログラムを**タスク**といい、Windows は複数のタスクが同時に実行されるマルチタスクシステムである。Ctrl+Alt+Delete の3つのキーを同時に押すと、タスクマネージャーが現れ、実行中のアプリケーションやプロセスのリストを見ることができる。

また、タスクマネージャーでは実行中のタスク(アプリケーション)を強制終了させることができるので、操作ミス等で操作(終了)不可能になってしまったアプリケーションは、タスクマネージャーから終了させることができる。

実習 3. タスクマネージャーの操作

以下の手順に従って、アプリケーションが操作不能に陥ったときの強制終了を行ってみよう。 (1)適当なアプリケーションを起動する。

(2)Ctrl+Alt+Delete キーを同時に押し、タスクマネージャーを起動する。

(3)**アプリケーションタブ**をクリックし、終了させたいアプリケーション(1で起動したもの)をク リックする。

(4)タスクの終了ボタンを押す。

(5)指定したアプリケーションが終了したら、タスクマネージャーを閉じる。



<u>4.1 OSの歴史</u> <u>4.2 OSの役割</u> 4.3 Windowsの基本

5 ファイル

この章のポイント

ファイルはコンピュータにおけるデータやプログラムのまとまりである。記憶装置(メ モリやハードディスク)に置かれたデータやプログラムはすべてファイルとして管理さ れる。ここでは、データやファイルについて様々な側面から学ぶ。

この章の構成

<u>5.1 データの種類</u>

<u>5.2 ファイルの種類</u>

5.3 ファイルシステムとファイル操作

5.4 ファイル圧縮

5. ファイル

5.1 データの種類 5.2 ファイルの種類 5.4 ファイルシステムとファイル操作 5.4 ファイル圧縮

5.1 データの種類

コンピュータが「O」と「1」で動いていることは、よく知られている。では、さまざまなデータを コンピュータ内部ではどのように表現しているのだろうか。

0.2進法、16進法

通常われわれが使っている記数法は10進法で、例えば

323=3×10²+2×10¹+3×10⁰=300+20+3 である。ここで、323の左端の3は300を意味し、右端の3は3を意味するというように、10進法 では、0~9の10種類の数字がその位置によって意味(表す数)が異なり、k 桁目(右から k 番 目)の数字 a は a×10^{k-1}を表す。このような記数法を、基数10の位取り記数法(10進法)とい う。人類がいかにして10進法を獲得したか、またその重要さについては、吉田洋一著「0の発見 一数学の生い立ち」岩波新書、が名著である。

一般に基数N(≧2)の位取り記数法(N進法)では、N種類の数字を用い、

 $\begin{array}{l} a_{k}a_{k-1}\cdots a_{2}a_{1}a_{0}^{,} \overset{}{N}_{,} a_{k}^{,} \times N^{k} + a_{k-1}^{,} \times N^{k-1} + \cdots + a_{2}^{,} \times N^{2} + a_{1}^{,} \times N^{1} + a_{0}^{,} \times N^{0} \\ & \varepsilon_{\overline{s}} \overset{}{\sigma}_{,} \overset{}{\varsigma}_{,} \overset{}$

であるから、与えられたN進数a_ka_{k-1}···a₂a₁a₀が表す値を求める(10進→N進変換)には、

a_kから始めて、N倍して次の数字(が表す数)を加えることを、a₀まで繰り返せばよい。

逆に与えられた数のN進表現を求める(10進→N進変換)には、

Nで割ってその余りを求めることを繰り返して、 a_0 、 a_1 、 a_2 、···、 a_{k-2} 、 a_{k-1} 、 a_k を順に求めればよい。

2進法

5 ••• 1

2)

N=2の2進法では、0.1を用いて数を表す。 例えば2進数の10110が表す値は、上で述べたように $(((1 \times 2 + 0) \times 2 + 1) \times 2 + 1) \times 2 + 0$ Î $=((2 \times 2 + 1) \times 2 + 1) \times 2 + 0$ 1 $=(5 \times 2 + 1) \times 2 + 0$ 10 101 $=11 \times 2 + 0$ Ť 1011 = 2210110 という手順で求まる。この手順は言い方を変えれば、 1、10、101、1011、10110が表す数を順に求め ていることに他ならない。 逆に10進数の22を2進数にするには 2) 22 2) 11 ··· 0

| 2進法 | 10進法 | 16進法 | 2の補数 |
|------|------|------|------|
| 0000 | 0 | 0 | 0 |
| 0001 | 1 | 1 | 1 |
| 0010 | 2 | 2 | 2 |
| 0011 | 3 | 3 | 3 |
| 0100 | 4 | 4 | 4 |
| 0101 | 5 | 5 | 5 |
| 0110 | 6 | 6 | 6 |
| 0111 | 7 | 7 | 7 |
| 1000 | 8 | 8 | -8 |
| 1001 | 9 | 9 | -7 |
| 1010 | 10 | A | -6 |
| 1011 | 11 | В | -5 |
| 1100 | 12 | С | -4 |
| 1101 | 13 | D | -3 |
| | | | |

| 2) | 2 | • • • | 1 |
|----|---|-------|---|
| 2) | 1 | ••• | 0 |
| | 0 | • • • | 1 |

| 1110 | 14 | Е | -2 | | |
|-----------------|----|---|----|--|--|
| 1111 | 15 | F | -1 | | |
| 2の補数表現についてはコラムを | | | | | |
| 参照せる | ۲. | | | | |

として、求めた余りを下から並べて 10110 とすればよい。 これが、2進数10110が表す数を求める手順の逆になって いることは、明らかだろう。

1

16進法

基本はN=16のN進法であり、16個の数字には上の表のようにO~9とA~Fを用いる。したがって、16進数11B が表す(10進数)値は(1×16+1)×16+11=283である。

16=2⁴なので、16進法には、2進4桁あるいは4ビットをまとめて1文字で表す、という意味合いもある。

細かいところまで理解する必要はないが、以下の欄で、整数、小数点数、および文字の内部表現 を表示させてみよう。

1.整数

0以上の整数は2進法で表す。負の数は2の補数表示(コラム参照)で表現され、32ビットの整数ならば、-2³¹=-2147483648~2³¹-1=2147483647の整数を表すことができる。計算の途中で表現範囲(32ビット)を越えてオーバーフローを起こしても無視されて計算が続けられてしまうことは覚えておこう。以下の整数型電卓で、表現範囲を超えたときの計算結果を確認してみよう。



上下の欄の数値で演算を行い、結果は上の欄に表示される。半角数字で入力すること。

問. 上の電卓を用いて、以下の計算を行え。 1)1000000×1000000 2)2³² またその理由は何か。

2.小数点数

浮動小数点数(0.31415926×10¹¹のような、仮数部×基数^{指数部}の形の数)として表現され る。通常の10進表現では基数は10であるが、コンピュータ内部では2進表現なので基数も2であ る。計算は仮数部の有効桁数内での近似計算になるので、誤差がさけられない。通常は64ビッ トの倍精度浮動小数点数が使われ、仮数部の有効桁数は10進15~16桁である。以下の浮動 小数点型掛け算電卓で実験してみよう。

| 乗 | .0E0 | 0 | | |
|---|-------|----------|---|---|
| | ÷ | × | _ | + |
| | - | <u>^</u> | | |

上下の欄の数値で演算を行い、結果は上の欄に表示される。半角数字で入力すること。また 「E11」は「×10¹¹」を意味する。

問. 1)1.0を10.0で何回か割っていくと何回目で誤差が現れるか。

2)1)の結果に割った回数分10.0を掛けていっても1.0には戻らないことを確認せよ。

3)1.0+1.0E−15 の計算結果はどうなるか。1.0+1.0E−17 ではどうか。

上の問の3)からもわかるように、(相対的に)大きな小数点数に小さな小数点数を加えても、値 は増えない(正しく計算できない)。したがって、以下の実験からもわかるように 1+1/2+1/3+・・・ +1/n と 1/n+1/(n-1)+・・・+1/2+1 とは、数学的には等しいが、コンピュータでは計算結果が異な る。前者は n がある程度以上大きくなると和の値がそれ以上増えなくなり、正確に計算できて いない。この例のように非常に多くの数の和を求める際は、小さな数から順に加えていく等のエ 夫が必要になる。

| 計算 | 3000000 | |
|-------------|--------------|--|
| +1/2+1/3+ • | ・・+1/n と | |
| /n+1/(n-1)+ | ・・・+1/2+1の比較 | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

計算時間を短くするため、単精度浮動小数点数(有効数字7~8桁)を用いている。

問. n=2000000, 3000000, 4000000 に対して 1+1/2+1/3+···+1/n と 1/n+1/(n-1)+···+1/2+1 の計算結果を比較せよ。

3.文字

英数字(半角文字)は1バイト(8ビット)で、漢字やひらがな(全角文字)は2バイト(16ビット)で符号化れる。文字を表す符号を2進数とみなすことによって、その文字の番号(何番目の文字か)が定まる 半角文字の符号化では ASCII符号が、全角文字の符号化には シフトJIS符号 が標準的である。 た、最近では世界中のすべての文字を2バイトで符号化するユニコードも使われている。



文字の表現に使われている符号を誤認すれば文字化けを起こす。WWWブラウザで[表示]→[文= エンコーディング]を変更してみよ。

問. 英字の小文字と大文字では番号がいくつ違うか。この事実を利用して大文字を小文字に変換 るにはどうすればよいか。

4.画像データ(カラー)

ドット(ピクセル、画素)ごとに 光の3原色(RGB)それぞれの輝度(明るさ)を1バイト(256通り)で 指定する。

下のカラーパレットで代表的な色(赤、青、黄、茶、紫、白、黒、・・・)とRGB値の関係を調べてみ よ。



問. 1)光の3原色RGBを各1バイトで表わすとき、色はすべてで何色あるか。 2)1000×800画素の画像データは、圧縮をまったく行わないとき、何メガバイトになるか。 3)自分が使っているパソコンのモニターの画素数を、スタート→コントロールパネル→画面→設 定、で確認せよ。

5.音声データ

細かく区切った時間単位で、波形の高さを数値化して、2進表示する。例えばCD 音楽の場合、約44KHz(毎秒44,000回)で2パイト(65536段階)に数値化している。 問、2時間分のステレオ音楽のCDデータは何メガバイトになるか。

6.動画データ

毎秒30枚程度の画像を、音声データと同期をとって表示させる。 問. 1000×800画素の動画2時間分の画像データは、圧縮をまったく行わないとき、何ギ ガバイトになるか。

7.AD変換

画像や音声のような本来連続的なアナログデータをコンピュータで扱える離散的なデジタルテ ータに変換することをAD変換という。AD変換の手順は、1)標本化、2)量子化、3)符号化の3段階 に分かれる。

- 標本化では、アナログデータを細かな区間に分けて、それぞれの区間での値(標本値という)をとる。画像ではドット'画素)ごとの色をとることが、CD音楽では細かな時間間隔 (1/44000秒)ごとに波形の高さをとることが、標本化にあたる。標本値は基本的に連続量 である。
- 量子化では、標本化で得た標本値を、あらかじめ定められたレベルに分けて離散数値化 (デジタル化)する。カラー画像では光の3原色ごとに各々256段階に、CD音楽では65536 段階に量子化している。
- 3. 符号化では、量子化で得られた量子値(離散数値)を0,1の列で表す。例えば、カラー画像では3バイト、CD音楽では2バイトの符号になる。

コラム 2の補数表現

コンピュータ内部で0以上の整数は2進数で表現されるが、負の整数 -n はどう表現すればよいだろうか。ヒントは、整数を表すビット数 k(通常 k=32)が固定していることを利用して、減算に加算回路を使えるようにすることにある。

今 -n の内部表現に 2^k-n の2進表現を使うことにしよう。このとき 2^k に対応する k+1ビット目 (桁あふれ)の1を無視すれば

 $m+(-n) \Rightarrow m+(2^{k}-n)=2^{k}+(m-n) \Rightarrow m-n$

となる。ここで 2^k-n を -n に対する2の補数(k ビット整数の場合)という。

仮に整数が k=4ビット(実際は32ビット)で表されているとしよう。扱える範囲は -2³=-8~ 2³-1=7 であり、次のように対応する。ここで負の数はすべて最初のビットが1で非負整数の最初の ビットは0になっているので最初のビットから符号がわかることにも注意しよう。

| 0111=7 | 0110=6 | 0101=5 | 0100=4 |
|---------|---------|---------|---------|
| 0011=3 | 0010=2 | 0001=1 | 0000=0 |
| 1111=-1 | 1110=-2 | 1101=-3 | 1100=-4 |
| 1011=-5 | 1010=-6 | 1001=-7 | 1000=-8 |

例えば、3に対する2の補数は2⁴-3=13であるから、-3は13の2進表示 1101 で表される。こうしておけば、例えば、

| 5 | 0101 | | 6 | 0110 |
|-------|---------|-----------|------|------------------------|
| +) -3 | +) 1101 | | +) 7 | +) 0111 |
| | | | | |
| 2 | (1)0010 | (桁あふれは無視) | 13 | 1101 ⇒ <mark>-3</mark> |

のように計算できる。一般に、コンピュータの整数計算ではオーバーフローはエラーでなく単に 親され、4ビット整数で 6+7 を計算すると結果は 13 ではなく -3 になる。

5.1 データの種類 5.2 ファイルの種類 5.4 ファイルシステムとファイル操作 5.4 ファイル圧縮

5. ファイル

<u>5.1 データの種類</u> 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮 5.2 ファイルの種類

コンピュータが扱うデータの一まとまりをファイルと呼ぶ。ファイルはその中身に応じて、次のように分類され、拡張子(ファイル名の.以下3文字)で区別される。

- (1) **テキストファイル** 文字と改行記号から構成されるファイル。通常、ファイル名は….txt。
- (2) 実行ファイル 機械語命令から構成されるプログラム。ファイル名は….exe。
- (3) **画像ファイル** ファイル名は、….bmp、….jpg、….gif、….png など。
- (4) **音声ファイル** ファイル名は、….mp3、….wav、….aiff など。
- (5) 動画ファイル ファイル名は、…mpg、…avi、…mov など。
- (5) **圧縮ファイル** ファイル名は、….lzh、….zip、….tar など。
- (6) その他のデータファイル アプリケーションで利用されるデータから構成されるファイル。 その形式はアプリケーションによって異なる。

実習1. ファイルの「保存」と「開く」

WORDやEXCEL等のアプリケーションで作成したデータは、外部記憶装置 (ハードディスク)にファイルとして保存しなければ電源OFFとともに消えてしま う。ファイルの「保存」と保存したファイルを「開く」操作等は、ウィンドウズOSの 機能を利用しているためほとんどのアプリケーションで共通である。ここでは、 そのことの理解とともに、次節の準備のためにいくつかの種類のファイルを作 成して保存しておこう。

- (1)メモ帳(エディタ)を起動して、テキストファイルを作成し、個人領域(Zドライ ブ)に新しいフォルダを作りそこに保存せよ。
 - 1. メモ帳を起動し、データ(テキスト)を適宜入力する。
 - 2. ファイルメニューから「名前をつけて保存」を選ぶ。

| 名前を付けて保存 | | | | | | | | ? 🛛 |
|------------|---------------------------------------|----------------|---|---|---|---|----|--------|
| 保存する場所の | 27 FF1X2F | | ~ | 0 | 1 | P | | |
| 最近使ったファイル | ● My Webs ■ マイ ピクチャ ● マイ ミュージック | | | | | | | |
| デスクトゥブ | | | | | | | | |
| 71 F#1X>F | | | | | | | | |
| ۱۹- ۲۷/۱-۶ | | | | | | | | |
| | | | | | | | | |
| マイネットワーク | 77-11-名(N): | ファイル名txt | | | | | ~ | (#存(S) |
| | ファイルの種類(1): | テキスト文書 (*.txt) | | | | | ¥. | キャンセル |
| | 文字コード(匠) | ANSI | | _ | | 1 | ~ | |

- 3. 保存する場所として、個人領域(Zドライブ)を選ぶ。
- 4. 「新しいフォルダの作成」ボタン(上段右から二つ目)を押し、作成され た新しいフォルダに名前(例えば、情報基礎)をつける。
- 5. 今作成した新しいフォルダをクリックして、そのフォルダを開く。
- 6. ファイル名 (通常、拡張子 . txt は省略する)を入力して保存ボタンを 押す。

(2)WORDを起動して、WORDのデータファイルを作成し、(1)と同様に保存せよ。

(3)ペイントを起動して、画像ファイルを作成し、(1)と同様に保存せよ。 (4)WORDを再起動し、(2)で保存したWORDファイルを開け。 (5)ペイントを再起動し、(3)で保存した画像ファイルを開け。

<u>5.1 データの種類</u> 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮

5. ファイル

<u>5.1 データの種類</u> 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮</u> 5.3 ファイルシステムとファイル操作

1. 階層構造

まず、ウィンドウズ標準のファイル操作アプリケーションであるエクスプローラを起動して、パソコン のファイルシステムを眺めてみよう。実習1で行ったように、ファイルはフォルダ(ディレクトリともいう れて整理される。フォルダにはフォルダを入れる(作る)こともできる。このような構造は、データモデ ころでも述べたように、一般に、階層構造(木構造)と呼ばれる。

実習2 エクスプローラによるファイル操作

(1)エクスプローラを起動し、左側のフォルダウィンドウの[+]のついているフォルダ(または ドライブ)をいくつかクリックせよ。ドライブは、記憶装置に対応しており、その下にフォル ダ(またはファイル)が格納される。

ー橋大学のシステムでは、個人領域はZ**ドライブ**である。

(2)次のファイルを探し出し起動せよ。 C:¥Program Files¥Microsoft Office¥Office11¥WINWORD.EXE

意味は「CドライブのProgram Filesフォルダ中のMicrosoft Officeフォルダ 中のOffice11フォルダ中のWINWORD.EXEファイル」ということ。

ファイル名は、それが存在する場所(フォルダ)示すパス、狭義のファイル名、拡張子とからなる。 子はファイルの種類を示すもので、これをもとにそのファイルを処理するアプリケーションが関連付る。

コラム URL(Universal Resource Location)

インターネット上でのファイルの場所指定は(URLUniform Resource Locato)と呼ばれる形式で、 種類://マシーンの名前.場所/ファイルの場所/名前

のように指定される。たとえば、

http://www.hit-u.ac.jp/laboratories/index.html

であれば、httpはいわゆるWebページ(より正確にはWebページの要求と送受信を行うための規約 意味し、wwwがマシーン名、hit-u(一橋大学)、ac(教育研究機関)、jp(日本)を意味する。そのマ ン(上のWebページ用のフォルダ内)のlablratoriesフォルダの index.htmlという名前のファイルという になる。インターネットでのマシーンの場所指定は欧米流の順序で階層(木)構造の下(葉)から上 に向かって登るのに対し、マシーン内のファイルの場所指定は日本流で上(根)から下(葉)に降り くことに注意しよう。

2. ショートカット

1. では、ファイルシステムは階層構造であると述べたが、ショートカット(近道)の機能があるためには正しくない。ショートカットは階層構造の奥深くにあるフォルダやファイルへの簡便なアクセス(j バイパス)を提供するもので、このことを考慮するとファイルシステムのデータモデルは循環する路 クル)を持たないグラフモデルになる。

例えば C:¥Program Files¥Microsoft Office¥Office11¥WINWORD.EXE へ至る経路には C:⇒Program Files⇒Microsoft Office⇒Office11⇒WINWORD.EXE

の他に、パソコンの設定によって異なるが デスクトップ⇒Wordアイコン スタート⇒すべてのプログラム⇒Microsoft Office⇒Wordアイコン
などがある。

3. ファイルのコピー、移動、削除

実習3 ファイル操作:切り取り、コピー、貼り付け、削除、名前の変更

- (1)エクスプローラを起動し、Zドライブに新しいフォルダを作れ(フォルダ名は適当に)。 操作:**ファイル→新規作成→フォルダ**
- (2)実習(1)で作成したフォルダを開け。
- (3)実習1で作成したテキストファイルを(1)で作成した新しいフォルダにコピーせよ。 手順は以下の通り。
 - 1. コピーしたいファイルにマウスを合わせて右クリックし、現れたウィンドウでコピーを 選ぶ。
 - この結果、選択したファイルのコピーが、クリップボードと呼ばれる場所に作られる。
 - 3. コピーを作成したいフォルダにマウスを合わせて右クリックし、現れたウィンドウで 貼り付けを選ぶ。
 - 4. この結果、クリップボード上のファイルが、指定された場所に貼り付けられる。
- (4)実習1で作成したWORDファイルを、(1)で作成したフォルダに移動させよ。

(2)と同様の手順であるが、ファイルを「コピー」でなく、「切り取り」する。

(5)(1)で作成したフォルダをフォルダごと削除せよ。 このように、ファイルだけでなく、フォルダごと、コピー、移動、削除などの操作が可能である。

コラム オブジェクト指向

従来のコンピュータは、アプリケーション(プログラム)が中心で、データはその付随物(処理 対象)だと考えられていた。したがって、何か処理を行わせる手順は、まずどうするかを指定 (アプリケーションを起動)し、次にその対象を指定するという、英語の「動詞+目的語」の順 であった。

現在は、オブジェクト指向と呼ばれるデータを中心とする概念で設計されており、実習3で見たように、何をどうするという(日本語の)順番で指定する。また、対象を右クリックすると、その対象に可能な処理(受け付ける命令)の一覧表が表示される。

また、対象をダブルクリックすれば、それに対する**デフォルト**(標準)の処理が行なわれ、対象をアプリケーションにドラッグ&ドロップすれば、そのアプリケーションで処理されることも 覚えておこう。

ー般にコンピュータ内のすべての対象(オブジェクト)は、その内容、サイズなどの属性(プロパティ)だけでなく、それが受け付ける命令(メソッド)を持つという考えかた(設計方針)をオブジェクト指向という。

5.1 データの種類 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮

一橋大学

5. ファイル

<u>5.1 データの種類</u> 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮 5.4 ファイル圧縮と誤り検出・訂正符号</u>

種々のファイルに対し、圧縮操作を施し、その大きさを比べてみよう。

実習4 実習1で作成したファイルに対し、圧縮操作を施し、その大きさを比べる。

- (1)エクスプローラから(スタート→最近使ったファイル、でもよい)、実習1で作成した画像フ ルを右クリックし、編集を選べ(ペイントで開くことになる)。
- (2)ファイルの種類を24ビットビットマップ(BMP)にして、保存せよ。
- (3)ファイルの種類をJPEG(JPG)にして、保存せよ。
- (4)(2)で保存したファイルにマウスポインタを合わせて右クリックし、プロパティを選び、ファ サイズと画像のサイズ(縦・横のピクセル数)を調べ、サイズ=ドット数×3バイトになって ることを確認せよ。
- (5)(3)で保存したJPEGファイルにマウスポインタを合わせて右クリックし、プロパティを選び ァイルサイズを調べよ。
- (6)(4)、(5)で作成したファイルおよび実習1で作成したテキストファイルや文書ファイルを ツールで圧縮して、そのファイルサイズを調べよ。
 - ・マウスを合わせて右ボタンで「圧縮」から 圧縮の種類 を選ぶか、
 - ・「Lhaplus」などの圧縮ツールのアイコンにドラッグ&ドロップする。
- (7)結果をWebClassから報告せよ。

データ圧縮のアイディア

データ(ファイル)圧縮とは、元のデータが表す情報に対するより簡潔な表現を工夫して、元のデー に同じ情報を表しながら、よりサイズの小さなデータに変換することを言う。この際、圧縮データがう まったく同じ情報を表し、圧縮データからもとのデータを復元できるような圧縮を**可逆圧縮**という。そ 例えば画像や音声データの圧縮の際には、人間にはわからない程度の情報省略を行うことによっ を縮める手法がとられることがある。このような圧縮では、もとのデータを復元することはできない0 逆圧縮と呼ばれる。以下、圧縮の際に利用される基本的なアイディアを解説する。実際の圧縮では アイディアを組み合わせた、より高度な手法が使われている。

(1)ハフマン符号

モールス信号は、ツー(ー)とトン(・)という長短2種類の信号の組み合わせで文字を表す。英語ておける各文字の出現割合を調べ、よく出現する文字ほど短く表す工夫をして、文章を送信する際の げるようにしている。(因みに日本語では、おおよそabc順にイロハを割り当てたので、カナ文字の出 は対応していない。)

このように、文字(あるいは一定長の0,1列)の出現確率を調べ、確率の高いものほど短い符号を ものにハフマン符号がある。これによって、平均的な符号の長さを短くすることができ、ファイルのE がる。

問. 日本語および英語のモールス信号における各文字の符号を調べよ。

例. ハフマン符号を実際に求める例として、文字列 nakatanaka を考えよう。この文字列中の a, k, r 確率はそれぞれ 0.5, 0.2, 0.2, 0.1 である。まず出現確率の低い2文字 n, t に仮に 0, 1 の符号を割 れらをまとめた {n:0, t:1} の出現確率を 0.3 とする。すなわち、文字(集合) a, k, {n:0, t:1} の出現確³ れ 0.5, 0.2, 0.3 となる。同様に出現確率の低い2つ k, {n:0, t:1} に 0, 1 の符号を割り当て、これらを {n:0, k:10, t:11} の出現確率を 0.5 とする。最後に a と {n:0, k:10, t:11} にそれぞれ 0, 1 を割り振り、 n:10, k:110, t:111} を得る。

| a, 0.5 | k, 0.2 | n, 0.2 | t, 0.1 |
|--------------------------------|------------------------|----------|---------|
| a, 0.5 | k, 0.2 | {n:0, t: | 1}, 0.3 |
| a, 0.5 | {k:0. n:10, t:11}, 0.5 | | |
| {a:0. k:10, n:110, t:111}, 1.0 | | | |

このとき、元の文字列 nakatanaka は 110010011101100100 という長さ 18 のビット列で符号化され は、1 文字 8 ビットのASCII符号に比べれば 18/80=22.5% に圧縮されており、4 文字しか出現しなし 目してそれらを均等に 2 ビットで符号化する場合に比べても 18/20=90% の圧縮になっている。

ところで、符号列 110010011101100100 から元の文字列を復元するにはどうすればよいだろうか (左)から順に当てはまる(一致する)文字に置き換えてゆけば、110010011101100100⇒n0100111 ⇒na10011101100100⇒nak011101100100⇒・・・のように復元できる(試してみよ)。

実際、ハフマン符号を構成する際には、出現確率の低い文字(集合)をまとめては符号ビットを頭 けることを繰り返しているので、ハフマン符号列を頭から読んでゆくと、その構成過程を逆にたどっ 字に到達できるのである。

文字列 nakatanaka に限らず、文字 a, k, n, t の出現確率がそれぞれ 0.5, 0.2, 0.2, 0.1 であるとき(符号の平均符号長は 3×0.1+3×0.2+2×0.2+1×0.5=1.8 であり、4 文字すべてに同じ長さの k:01, n:10, t:11}を割り当てた場合の平均符号長 2 より小さい。なお、理論上最小の平均符号長を 均情報量(以下のコラム参照)は、この場合 -0.5log,0.5-0.2log,0.2-0.2log,0.2-0.1log,0.1=1.76 であ

問. 例題中のハフマン符号で列 10011001110 を復元せよ。

(2)ラン・レングス法

同じ0,1列が一定回数以上繰り返される場合に、繰り返し全体を、その列と繰り返し回数で置き換 よって、圧縮を図る方法。例えば、0000000が100回続くとき、8ビット×100=800ビットの0,1列が、 グス法では0000000の8ビットと回数100を表す8ビットの16ビットですむ。特に画像などで背景に同 ときに有効である。

(3)差分表現

動画における一連の画像列では、ある時刻の画像が直前の画像とほとんど変わらないことが多い な場合、各時刻で画像全体を保持するのではなく、直前の画像からの変化(差分)を保持するよう データのほとんどは0000000(変化なし)ということになって、(2)のラン・レングス法が有効に働く。

(4)非可逆圧縮

非可逆圧縮には、様々な手法があるが、いずれも、人間の耳や目では認識できないほど細かな変 る情報をカットして圧縮を図るものである。例えば、音声波形は様々な周波数の波を重ねて得られ うち細かな変化を表す高周波数成分をカットしても、人間の耳には違いが感じられない。同様に、 ても色調の大きな変化(低周波数成分)や細かな変化(高周波数成分)をわけて抽出することがで はその変化を感じられないような高周波数成分をカットすることによって、圧縮につなげることがで また、このように(離散)データを周波数成分に分けることを(離散)フーリエ変換という。

問.

紅月燐火氏によるフリーのグラフィックコンバータ「<u>BJT32</u>」をダウンロードし、jpeg変換の圧縮率と 係を確認せよ。

エラー検出・訂正符号

圧縮はどのようなデータに対しても可能なわけではない。もしそのようなことが可能なら、すべて0 そのサイズを限りなく小さくできてしまうが、それはあり得ない。実際、圧縮可能ということは、元デ-性があって、よりコンパクトな表現が可能なことを意味する。

一方、普段我々使う表現(文章、話し言葉、画像など)は、冗長性に満ちており、そのおかげで、た

が欠落したとしても、欠落部分を想像で補い、問題なく情報が伝達されることが多い。 ここでは、データを、冗長性を持った符号で表現することによって、転送(計算)時の誤りを検出し、 きたりする符号化について解説する。

(1)チェック・ディジット

受験番号や学籍番号、社員番号などで、番号の末尾にチェック・ディジットと呼ばれる英数字1文 いることがある。この最後の1文字は、本来の番号からある計算規則によって定められているもの 入力ミスや、(計算規則を知らない)第三者による悪用の場合に、対応する(番号+チェック・デジッ しないことが多く、これをを防ぐ目的がある。運転免許証や銀行の口座番号などでは、どれがチェッ ットかすぐにはわからないように数字で埋め込まれているという。

(2)パリティ・ビット

0,1の列の最後に0または1を1桁付け加えて、列中の1の個数が常に偶数(奇数でも同じ)にする トをパリティ・ビットという。言い方を変えれば、パリティ・ビットを付け加えることによって、どの0,1列 比較しても2か所以上異なっているようにする。

こうしておくと、1文字だけ誤って通信されたときは、1の個数の偶奇が異なる((正規の符号から1; 異なる)ため、対応する符号が存在せず、誤りがあったことが検出できる。これを1誤り検出符号とし 以上偶数個数の誤りは検出できないが、その確率は低い。)

例えば、半角英数字を表す8ビットASCII符号は、実質7ビットなので頭の1ビットをパリティ・ビット ることがある。

| (3)ハミング符号 | 情報ビット |
|---|---|
| (7,4)ハミング符号と呼ばれる右表の符号をみてみよう。左4ビットが本来の情 報を符号化したビットで、右3ビットは誤りの検出・訂正を可能にするための追 | x ₁ x ₂ x ₃ x ₄ |
| 加ビットである。詳しい作り方は省略するが、 y_1, y_2, y_3 はそれぞれ、 $x_2x_3x_4$ 、 | 0 0 0 0 |
| x ₁ x ₃ x ₄ 、x ₁ x ₂ x ₄ に対するパリティビットになっている。 | 0 0 0 1 |
| 表を調べると、異なる符号は互いに必ず3か所以上異なっていることがわか | 0 0 1 0 |
| る(確認せよ)。この場合、2か所以内の誤りであれば、誤りがあったことが検出 | 0 0 1 1 |
| できる。 | 0 1 0 0 |
| 一般に、すべての符号が互いにn箇所以上異なっているとき、(n-1)箇所の以 | 0 1 0 1 |
| 内の誤りまで検出可能で、[n/2](小数点以下切り捨て)箇所以内の誤りを訂正 | 0 1 1 0 |
| り hE Cの る。 | 0 1 1 1 |
| 問. (7,4)ハミング符号(右表)について以下の問に答えよ。 | 1 0 0 0 |
| 1) すべて互いに3か所以上異なることを確認せよ。 | 1 0 0 1 |
| 2) ビット列 0101101 は正してはとのようなビット列にと考えられるか。 3)すべての符号が互いにn箇所以上異なっているとき、(n-1)箇所の以内の誤 | 1 0 1 0 |
| りまで検出可能で、[n/2]箇所以内の誤りを訂正可能である理由を述べよ。 | 1 0 1 1 |
| | 1 1 0 0 |
| | 1 1 0 1 |
| | 1 1 1 0 |
| | |

コラム シャノンの情報量

上記の実習からもわかるように、同じ情報でも、それを表現するデータの(ファイル)サイズ きく異なることがある。言い換えると、データの大きさとそれが持つ情報の量とは、別の概念⁻ る。Shannonは以下のような観点から、**情報量**という概念を定義した。

シャノンは、よく起こる事象に対し、その情報量(その事象が起きたという情報の価値)が低めったに起きない事象はその情報量が大きいと考え、事象Pの情報量はその生起確率 p カ

-+

定義される関数 f(p) であるとした。

また、二つの(独立な)事象 $P_1 \ge P_2$ が同時に起きたことの情報量はそれぞれの情報量 f($\ge f(p_2)$ の和になると考えた。 $P_1 \ge P_2$ が同時に起きる確率は p_1p_2 だから、 $f(p_1p_2)=f(p_1)+f(p_1)+f(p_2)$ でなければならない。

このような条件を満たす関数として、シャノンは確率pで起きる事象の持つ情報量を -log₂p 定義した。ここで、対数の底を2にしたのは、2者択一で確率1/2の事象の情報量を情報の最 位1(ビット)に合わせて 1 とするためである。

学習がすんだら、小テストに進み、学習事項を確認しよう。

5.1 データの種類 5.2 ファイルの種類 5.3 ファイルシステムとファイル操作 5.4 ファイル圧縮

file://C:\Documents and Settings\yamasaki\デスクトップ\秀記\Siryou_JouhouKiso201... 2010/12/13

6. 検索と置換

この章のポイント

googleやyahooなどの検索サイトで、キーワードを入力して必要な情報を検索したり、WORDなどの文書作成ソフトで文字列を検索したり置き換えたりすることがよく行われる。この章では、より進んだ検索と置換の技法を学ぶとともに、テキスト処理への応用について理解を深める。

この章の構成

<u>6.1 論理式検索</u>

<u>6.2 WORDにおける検索と置換</u>

6.3 正規表現によるパターン検索

6. 検索と置換

6.1 論理式検索 6.2 WORDにおける検索と置換 6.3 正規表現によるパターン検索

6.1 論理式検索

1. 論理式検索

通常の検索ではいくつかのキーワードを並べて、それらがすべて含まれるホーム ページや文書を検索する。しかし、検索サイトでは、いくつかのキーワードを積 (and)、和(or)、差(not)で組み合わせて検索できる。論理回路の章ですべての論 理回路が \land 、V、 \neg の3種類の素子から構成できることを説明したが、それと同様 に、キーワードの現れる、現れないを含めたあらゆる組み合わせは、積(and)、和 (or)、差(not)を用いて表現できることに注意しよう。 論理式 B A and B A and B +ーワードA、Bがともに現れるページを検索する。 A or B A or B

A not キーワード A が現れるページから、Bが現れるページを B 除外する。

2.Googleでの論理式の利用例

Googleの「検索オプション」ボタンを押して現れる画面では、 「すべてのキーワードを含む」(and) 「フレーズを含む」 「いずれかのキーワードを含む」(or) 「キーワードを含めない」(not)

などの欄に、キーワードやフレーズを入力できるようになる。 さらに、「ヘルプ」の「詳しい検索方法」により進んだ説明がある。

2.論理式の利用例

フレッシュアイ(<u>http://www.fresheye.com/guide/wizzard.html</u>)では、簡単なガイドに 従って積(&)、和(|)、差(~)の演算子を使った論理検索式の作成ができ、その仕組 みがよくわかる。

注意!

以下の実習にあたっては、半角文字と全角文字の違いに留意せよ。 とくに、全角の空白が思わぬ働き(悪さ)をすることがる。

実習1.以下は、検索式の例である。実際に検索してみよ。

1. コンピュータまたはインターネットに関する雑誌を知りたい 雑誌&(コンピュータ | インターネット)

- 2. 日本橋の会社で日本橋本町以外の所を調べたい (日本橋[~]本町)&会社
- 3. 神奈川県の横浜か川崎の住宅・住宅情報を見たい 神奈川県&(横浜|川崎)&(住宅|住居)
- 東京の本屋さん(書店も)で古本以外を探したい 東京&(本屋|書店)[~]古本
- 5. 東京にある金融機関でノンバンク、消費者金融以外のところを調べたい

金融&東京~(消費者金融|ノンバンク)

6.1 論理式検索 <u>6.2 WORDにおける検索と置換</u> <u>6.3 正規表現によるパターン検索</u>

6. 検索と置換

<u>6.1 論理式検索</u> 6.2 WORDにおける検索と置換 6.3 正規表現によるパターン検索 6.2 WORDにおける検索と置換

1.WORDにおける進んだ検索

| キーワード検索では、論理式を用いたとしてもキーワードの組み合わせによる検 索しかできないが、WORDの検索・置換機能では、単なるキーワードの検索・置換か ら、ワイルドカードを用いたパターン指定による検索・置換が可能である。以下、そ の機能を見てゆこう。 まず、WORDを起動し、適当な文書ファイルを開くか作成したら、検索あるいは置 換を選択すると、「検索と置換」ウィンドウが開く。そこで【オプション】キーをクリックし て、「ワイルドカードを使用する」にチェックを入れる。これによって以下のワイルドカ ードが利用した検索ができるようになる。 なお、詳しくは、WORDのヘルプで「文字列やその他の項目を検索して置換する」 ページの「ワイルドカードを使って検索または置換する」、「検索と置換で使用するワ イルドカード」を見よ。 | | | |
|--|--------|---|--|
| 目的 | 記法 | 使用例 | |
| 任意の 1 文字 | ? | "s?t" で、"sat" や "set" を検索でき る。 | |
| 任意の文字列 | * | ″s*d″ で、″sad″ や ″started″ を検索で きる。 | |
| 単語の先頭 | < | " <inter" で、"interesting"="" や<br="">"intercept" は検索できます が、"splintered"は検索できません。</inter"> | |
| 単語の末尾 | > | "in>" で、"in" や "within" は検索でき ますが、"interesting" は検索できませ ん。 | |
| 指定した文字のいずれか | [] | "w[io]n" で、"win" と"won" の両方を 検索できます。 | |
| 指定した範囲内の任意の 1 文 字 | [-] | ″[r-t]ight″ で、″right″ や ″sight″ を検 索できます。範囲は昇順で指定します。 | |
| 括弧内の範囲に含まれる文字 を除く任意の 1 文字 | [!x-z] | "t[!a-m]ck" で、"tock" や "tuck" は検 索できますが、"tack" や "tick" は検 索できません。 | |
| 直前の文字または式を n 個 | {n} | "fe{2}d"で、"feed" は検索できます が、"fed" は検索できません。 | |
| 直前の文字または式を n 個以 上 | {n,} | ″fe{1,}d″ で、″fed″ や ″feed″ を検索で きます。 | |
| 直前の文字または式を n ~ m 個 | {n,m} | ″10{1,3}″ で、″10″、″100″、および ″1000″ を検索できます。 | |
| 直前の文字または式を 1 個以 上 | @ | "lo@t" で、"lot" や "loot" を検索でき ます。 | |

例えば、数字列を検索するには [0-9]{1,}、小文字列の検索には、[a-z]{1,}とすれば よい。また、20世紀の年号(1900以上1999以下の数)を探したければ 19[0-9]{2}と する。

メモ

括弧を使ってワイルドカードと文字列を入力すると、検索条件の優先順位を指定することができます。たとえば、「<(pre)*(ed)>」と入力すると、括弧内のpreやedの条件が優先されるので、 "presorted" や "prevented" が検索されます。

2.WORDにおける進んだ置換

次に、進んだ置換機能について説明しよう。ワイルドカードを使用した検索式の一部 を上のメモにあるように括弧 "("、")"で括ると、その部分の検索が優先されるだけで なく、その部分とマッチした文字列を ¥n (nは数)で表すことができる。 ここで ¥n はn 番目の括弧対で括られた式にマッチした文字列を表す。したがって、例えば以下のよ うなことができる。

検索する文字列:(米国)(*)(日本) 「米国・・・日本」とマッチし、¥1=米国、¥2=・・・、¥3=日本となる。

置換後の文字列: ¥3¥2¥1 とすると、文書中の「米国」をその後方にある「日本」と入れ替える。

実習2.

(1)WORDを起動し、カット&ペースト機能を使って以下の4行を文書にコピーせよ。

山田 太郎 (Taro Yamada) 金岡 健二 (Kenji Kanaoka) 田中 正一 (Syouichi Tanaka) 文書中の「米国」をその後方にある「日本」と入れ替える。

(2)この文書上でワイルドカード置換を使って米国と日本を入れ替えよ。

(3)この文書上でローマ字からなる単語を検索せよ。

(4)この文書上でローマ字の名前と苗字の順番を入れ替えよ。

ワイルドカード

ワイルドカードは、トランプ等のカードゲームにおいて、ジョーカー等他のカードの 代わりとして使えるカードをさす言葉であるが、情報科学においては他の文字 (列)の代わりに使える(マッチする)特殊な文字を指し、一般に、任意の文字列を 表す*と、任意の1文字を表す?が使われる。 例えば、ファイルの「開く」や「名前をつけて保存」のウィンドウで「ファイルの種類」

を指定する欄には、すべてのファイル(*.*)、Word文書(*.doc)が指定できる。 *.doc を指定すれば、vb.doc、perl.doc、文書.doc のような・・・.docの形をしたすべ てのファイル名を意味し、このパターンにマッチする名前のファイルだけが、一覧 表示される。



<u>6.1 論理式検索</u> 6.2 WORDにおける検索と置換 6.3 正規表現によるパターン検索

6. 検索と置換

6.1 <u>論理式検索</u> 6.2 WORDにおける検索と置換 6.3 正規表現によるパターン検索

6.3 正規表現によるパターン検索

前節ではWORDにおけるワイルドカードを学んだが、インターネット上では、正規表現によるパターン指定が可能な検索サービスがあり、指定されたパターンにマッチ(適合)する文字列を教えてくれる。ここでは、正規表現による検索・置換について学ぼう。

例:インター?フ[エェ](イ|ー)ス は、インターフェイス、インターフェース、インターフエイス、インター フエース、インタフェイス、インタフェース、インタフエイス、インタフエース、の8種類の文字列とマ ッチする。?、[]、|、の意味については以下を参照せよ。

1. 正規表現

| 以下に正規表現の記法について述べる が違っていたりするものなので、注意せ | o。赤字の部分は、WORDのワイルドカードにはなかったり意味 ・よ。 | |
|--|--|--|
| (1) 基本演算 正規表現の基本演算は次の3種類の源 記法 説明 | 算である。 使用例 | |
| 連接 正規表現を続けて書けば、そ けた文字列とマッチ | れらを続 abcとdefの連接は abcdef | |
| 和(または) 。どちらかのパタ- チすればマッチ。 | ーンとマッ(this that) is a sample → this is a sample、that is a sample とマッチ。 | |
| 直前の文字(文字列)の 0回」 * り返しとマッチ | <mark>以上の繰</mark> 12* → 1、12、122、・・・とマッチ。 (ab)*a →a、aba、ababa、・・・とマッチ。 | |
| また、演算の順序を表すために括弧(, を表すには、直前に"¥"をつける。例え |)を用いる。さらに、*、 、(、)など特別な意味を持つ文字自身 ば、¥* は * と、¥¥ は ¥ とマッチする。 | |
| 正規表現では、基本演算の他に | こ便利のために種々の演算・記法が用意されている。 | |
| (2)文字の集合 | | |
| 記法 | 使用例 | |
| . 改行文字を除く <mark>任意の1文字</mark> | windows →windowsNT、windows98、windows3. などwindowsの後に2文字続く文字列とマッチ | |
| [] []内に並べた文字の集合 | [ab]=a b | |
| - 区間。[]内で用いる | [a-z]=a b ・・・ z a-zの小文字アルファベット全て [0-9]=0 1 ・・・ 9 全角の数字 [0-9A-Z] 大文字のアルファベットか数字 | |
| ^ 否定。[]内で用いる | [^a-z] 小文字のアルファベット以外全て | |
| (3)繰り返し指定 | | |
| 記法 | 使用例 | |
| + 前にある文字の1回以上の繰 | <mark>り返し</mark> 12+ → 12、122、・・・ とマッチ 12+=122* | |
| ? 前にある文字の0,1回の繰り込 | ^図 し 12? → 1、12 とマッチ 12?=1 12 | |
| {n} n回繰り返し | (ab){3} → abababにマッチ | |
| {n, } n回以上繰り返し | a{3,} →aaa、aaaa、・・・にマッチ | |
| {m, n} m回以上 n回以下繰り返し | a[3,5] →aaa、aaaa、aaaaalこマッチ | |
| ★ マッチする文字列は、通常はパターンに適合する最長の文字列(最長マッチ)であるが、"?"を 付加する事によって最短マッチとなる。 12+ → 1222222222 10桁目の12222222までとマッチ 12+? → 122222222 2桁目の12までとマッチ | | |
| | | |

٦

(4)特殊な意味を持つ文字 記法 使用例 説明 行頭 ^abc →行頭にある abc とマッチ \$ 行末 abc\$ →行末にある abc とマッチ その他特殊な意味を持つ文字 説明 記法 説明 記法 ¥w アルファベット、数字又は下線 ¥W アルファベット、数字、下線以外 ¥d 数字 ¥D 数字以外。[^0-9]と同じ ¥s 空白文字(スペース、タブ、改行) ¥S 空白文字以外 ¥b 単語の区切り ¥B 単語の区切り以外 ¥n 改行 ¥r リターン(復帰) ¥t タブ (5)置換

正規表現においても、WORDのワイルドカードと同様に、置換を行うことができる。正規表現において n 番目の括弧対で括られた部分にマッチした文字列は、\$n (nは数)で表すことができる(¥ではなく\$で あることに注意)。

実習3. 正規表現検索を用いた英文分析

正規表現による検索を用いると、例えば英語の学習に役立つばかりでなく、文書・文献の特徴を数量的に抽出し、著者の異同や時代の推定、思想と文体との関係等を分析することも可能になる。秋田大学の以下のサイトで、実験してみよう。

- <u>http://www.mis.med.akita-u.ac.jp/~taka/FreeSCORP/</u>では、正規表現検索を応用して、電子的に保存されている英文の解析を行うことができる。このページを表示させよ。
- 2. Search Word欄に ¥s(for|in|on|to|with)¥s[a-z]+ing¥s(for|in|of|on|to|with)¥s を入れてみよう。これが何を意味するか考えてみよ。
- 3. 適当な動詞(過去、過去分詞、三単元を含む)の出現数を調べよ。

実習4.

以下の欄で、正規表現による検索・置換の実験をしよう。検索文字列や置換文字列を指定して 検索・置換ボタンをクリックすると結果が下の欄に表示される。以下2)~7)については文字(数 字)列を表す正規表現を求めよ。 1)実習3をWordで行ったのと同じことを行え。 2)3桁の数字 3)3000未満の非負整数 4)aで始まりaで終わる6文字以下の英小文字列 5)数字0~9で始まり句点。で終わる文 6)3で割り切れる2進数 7)aaを含まないa,b上の文字列 検索文字列(正規表現使用可): 検索開始 置換文字列(変数 \$n 使用可) : | 置換開始 検索対象 : 💿 一つ目 🔵 全て 大文字小文字の区別: 💿 あり 🔘 なし

検索·置換結果:

2. 正規表現とグラフモデル

正規表現は、情報の基本モデルであるグラフモデル(ネットワークモデル)と密接な関係がある。グラフは対象 を〇や口で図示し、その間の関係を矢印(→)や実線(—)で図示したものであった。ここで、特に対象を「状 態」と呼んで〇で表し、関係を「状態遷移」と呼んで→で表したグラフモデルを状態遷移図(有限オートマトン) と呼ぶ。



左図 (<u>http://blogs.wankuma.com/aqua/archive/2007/07/11/84720.aspx</u>から引用)はキー入力による電卓の 状態遷移を表現した図である。右図 (<u>http://www.is.s.u-tokyo.ac.jp/vu/vu_lesson.php</u>から引用)は初期状態q₀

(→O)から0,1の列を入力したとき、それが表す2進数を3で割った余りnを状態q_nで表した状態遷移図である。 状態遷移図で、ある初期状態から、ある終了状態への遷移列のパターン(集合)は、正規表現で表されるこ とが知られている。(鉄道路線図で言えば、出発の駅から目的の駅までの経路のパターンは正規表現で表さ れる。)したがって、あるパターンの正規表現を求めようとするとき、そのパターンを表す状態遷移図を考えて みるとよい。

また逆に、正規表現に対して、それを遷移列(経路)集合に持つ状態遷移図を作ることもできる。これらの詳細については、言語理論、オートマトン理論に関する書物を参考にしてほしい。

問.3で割り切れる2進数を表す正規表現を求めよ。

ヒント.上右図で初期状態q₀から3で割った余り0の状態q₀への入力列(経路)パターンを表す正規表現を求める。

<u>6.1 論理式検索</u> 6.2 WORDにおける検索と置換 6.3 正規表現によるパターン検索

7 インターネット

この章のポイント

インターネットとは、LAN(Local Area Network)と呼ばれる会社や大学などの組織 や構内のネットワークを結ぶネットワーク(ネットワークのネットワーク)として発達し てきた。今回はインターネットの仕組みと利用上の注意、そしてなぜそのような注意 が必要なのかを学ぶ。

この章の構成

7.1 インターネットの歴史

7.2 通信の仕組み

7. インターネット

7.1 インターネットの歴史 7.2 通信の仕組み 7.1 インターネットの歴史

1.インターネットの歴史

インターネットの歴史を知り、その性格を理解しよう。

| 年号 | イベント |
|-------------------|---|
| 1964 | ポール・バランは、 情報をパケット(小包)化し、いくつかの中継所が遮断さ れても情報を迂回させて目的地まで伝達されるシステムの研究報告書を 提出。 |
| 1969 | UCLA、スタンフォード、UCサンタバーバラ、ユタ大学間に 24時間回線を 繋げっぱなしの computer network が誕生 。ARPA(国防総省高等研究 計画局)が指揮するこのプロジェクト はARPANETと名づけられた。 |
| 1970 ~ 1980 | 大学や研究所を結ぶ、ARPANETに似た さまざまなNetworkが誕生し た。 |
| 1984 | 情報の漏れを防ぐため軍関係のネットワークをMILNETとして分離させた。 JUNET(Japan University/Unix NETwork)が東大、東工大、慶應大間で構 築された。 |
| 1986 | アメリカNSF(全米科学財団)がNSFNETを構築した。 |
| 1988 | 民間企業も参加したWIDE (Widely Integrated Distributed Environment)プロ ジェクトでネットワーク技術等の実験が行われ、その技術と方式は現在の インターネットへと受け継がれている。 |
| 1989 | 一橋大学 JUNET参加 |
| | 政府機関や研究機関によって運営されたネットワークは当初私的・商業的 利用を禁止していた。インターネットの商用利用が可能になったのは、米 国で1990年、日本では1993年。以降、インターネット利用者が急激に増加 した。 |
| 1992 | ティム・バーナーズ・リーが, インターネットに繋がるサーバにある文書が簡 単に相互に閲覧・転送が可能になるようにURL(共通の文書名・住所の表 記)とHTML(共通の書式)とHTTP(転送プロトコル)を考案。この仕組み をWWW(World Wide Web)と名づける。 |
| 1993 | アメリカ・イリノイ大学の学生たちが、WWW閲覧プログラム「 モザイク」 を 開発し、画像も表示可能になった。 インターネット人口が爆発的に増加した。マスコミにも「インターネット」と いう言葉が登場し、一般にも知られるようになる。 |
| 1995 | Microsoft社もモザイクのライセンスを受け「インターネット・エクスプローラー」を開発した。インターネット人口は増え、日本でもブームとなる。 ー橋大学学内LAN開始。 |
| 1997 | 日本のインターネット人口572万人程、ドメイン名にjpの付く日本のホスト がアメリカに次いで第二位(約73万台)となる。プロパイダ数約1,650社。 |
| 2001 | 携帯によるインターネット接続が可能に。 |
| | |

ı.

| 2003 | インターネット人口5,645万3千人。家庭からのブロードバンド利用者数 1,596万2 千人。利用世帯の39.3%がブロードバンド接続。(インターネット 白書) |
|------|--|
| 2009 | インターネットの利用者数は9408万人、人口普及率は78.0% 携帯電話・PHSの普及率は96.3%(総務省「通信利用動向調査レポート」) |

この歴史からもわかるように、インターネットは大学内や研究所内のローカルなネットワー クを相互に回線で結んだものを起源にしており、電話や郵便システムのように、全体を責任 をもって管理する特定の組織や業者があるわけではない。



7.1 インターネットの歴史 7.2 通信の仕組み

7. インターネット

<u>7.1 インターネットの歴史</u> 7.2 通信の仕組み 7.2 通信の仕組み

1.パケット通信

インターネットは、最初に述べたようにLAN 同士を結ぶネットワークである。LANか らインターネットへの出口には、ルータ(あるいはゲートウェイ)と呼ばれるコンピュー タがあって、通信データの制御をしている。そのようなルータ同士をネットワークで結 んだものが、インターネットである。 インターネットにおける通信の 基本はバケツリレー式のパケット 通信にある。インターネット上を 流れるデータは、一定の大きさの パケット(小包)に分割され、それ インターネット ぞれに宛先や番号が付けられ て、送出される。 LANのルータは自分の所に来 たパケットの宛先を調べて、それ が自分のLAN内であれば、LAN 内に流し、そうでなければ、最適 LAN と思われる近接ルータにパケット を転送する(**ルーティング**)。した H がって、パケットは目的のLANに 到達するまでに、いくつものルー タを経由することになり、同じ宛 先のパケットであっても、そのル ートは一定せず回線の混雑状況 などによって異なってくる。 このように、インターネットは参 加者(ルータ)が隣接ルータへの 転送を互いに(ボランティア的に) 保障することで成り立つ、オープ ンで統一的な管理組織のないネ ットワークである。 このような仕組みであるため、インターネットを実質的に構成する各ルータ(の管理) 者)は、ルータに流れるパケットを、原理的には、盗み見や改竄できる。言い換える と、インターネットでの通信はその安全・秘密が保障されない、いわば「はがき」によ る通信であり、クレジットカード番号などの情報を、暗号化せずにそのまま流しては いけない。

2.IP(Internet Protocol、ネットワーク層プロトコル)

インターネットの通信規約について簡単に説明しよう。回線上で通信を行なうには 一定の手順や約束ごとが必要である。そのような約束・手順のことをプロトコル(通 信規約、通信手順)という。 まず、インターネットにつながるすべてのコンピュータはすべて原則として、固有の

アドレス(住所)を持つ。これをIPアドレスという。現在のIPv4(Internet Protocol Version 4)は 8bit × 4 の32ビットの番号が割り振られているが、徐々に128ビットの番号(IPv6:Internet Protocol Version 6)に移行しつつある。IPでは、インターネット

| を構成する各ルータが、宛先にしたがって適切な経路選択を行い、パケットを転送 するだけで、その到着、到着順序、到着データ内容を保障するわけではない。 | | |
|--|--|--|
| コラム インターネットプロトコルの設定 インターネットにつながるパソコン等で、インターネットプロトコル | | |
| (TCP/IP)のプロパティを調べると、パソコンの設定を見ることができる(する必要がある)。 通常は「IPアドレスを自動的に取得する」と「DNSサーバーのアドレスを自動的に取得する」が選択されていて、なにもする必要がない。この場合、DHCP(Dynamic Host Configuration Protocol)と言って、プロバイダー等のDHCPサーバーがインターネットに、接続しようとするパソコンに自動的に必要な情報を設定する。したがって、接続の度に設定されるIPアドレスが変わるので、例えば自前でWebページを持つようなことはできない。 一方、固定のIPアドレスを使う場合には、「次のIPアドレスを使う」と「次のDNSサーバーのアドレスを使う場合には、「次のIPアドレスを使う」と「次のDNSサーバーのアドレスを使う」を選んで、必要な情報を設定しなければならない。以下、設定情報をみてゆこう。ここで、DNS(Domain Name System)サーバーとは、例えば <u>www.hit-u.ac.jp</u> のようなサーバー(コンピュータ)のドメイン名から、IPアドレスを検索してくれるサーバーである。 | | |
| OIPアドレスを自動的に取得する。 O次のIPアドレスを使う IPアドレス ←このパソコンのIPアドレス サブネットマスク ←LANの内外を分ける情報 デフォルトゲートウェイ ←LANのゲートウェイ(ルータ)のIPアドレス | | |
| ODNSサーバーのアドレスを自動的に取得する O次のDNSサーバーのアドレスを使う 優先DNSサーバー 代替DNSサーバー | | |
| IPアドレス192. 168. 3. 15 サブネットマスク 255. 255. 255. 0 サ の場合192. 168. 3. * 宛てのパケットは宛先がLANの ブネ 内側なので、他のルータに渡されることなくLAN内に流 ット される。これ以外の宛先の場合、他のルータにパケッ マス トを渡して中継を頼む。どのルータがどの宛先のパケ ク ットを中継できるかという情報を 経路情報 といい、各ル ータは経路情報によってパケットを渡すルータを定め ることになる。 | | |
| コラム IPアドレスの枯渇 現在のIPv4では、IPアドレスが足りなくなってきており、コンピュータがイ ンターネットに接続したときだけー時的にIPアドレスを割り当てる方法 (DHCP、Dynamic Host Configuration Protocol)がとられることも多い。 次世代のIPV6では、いわゆるコンピュータに限らず、情報家電、や商品タ グなどすべてのものが固有のIPアドレスを持って、インターネットにつなが る ユピキタスコンピューティング 環境が構想されている。 | | |

問. インターネット上の経路表示ページ(<u>http://visualroute.visualware.com/</u>)で経路を調 べてみよ。

3.TCP(Transmission Control Protocol、トランスポート層プロトコル)

IPでは、パケットが分割された順番どおりに相手先へもれなく届くことを保障しているわけではない。IPの機能を使って、届かないパケットの再送を要求したり、届いたパケットを整理してもとのデータを再構成し、データの正しい送受信を保障するのがTCPの役割である。

具体的には、TCPを使って送受信されるパケットには一連の番号が付けられており、これを使ってデータの抜けや重複が検査される。データを受け取った側は、どのシーケンス番号のパケットを受け取ったかを送信元へ返し、これにより、送信側では、データが正しく相手に届いたことを確認する。もしある程度の時間待っても返信がなければ、途中でデータが消失したか、相手がパケットを取りこぼしたと判断し、再度パケットを送信する。

同様の働きをするトランスポート層プロトコルとして、TCPのような複雑な確認操作 を行わないUDPも使われている。UDPは転送速度は高いが信頼性が低く、TCPは信 頼性は高いが転送速度が低い。したがって、送受信に関する信頼性の高いネットワ ークではUDPが使われ、インターネットのような信頼性の低いネットワークではTCP が使われる。

| 4.インターネット上のサービスとP | ORT |
|-------------------|------------|
|-------------------|------------|

| インターネット上では様々なサービスが行なわれているが、そのサービスと提供するコンピュータをサーバーといい、サーバーに対し、そのサービスを受ける(要求する)側を クライアントという。サーバーとクライアントは、上で述べたTCP/IP上で定めれたそれぞれのプロトコル(通信規約) に従って交信を行なっている。 | | | |
|--|---------------|--|------------|
| サービ ス | 内容 | 使用するプロトコル | PORT番 号 |
| www | Webページの閲覧 | HTTP(Hyper Text Transfer Protocol) | 80 |
| メール | 電子メールの送受 信 | SMTP(Simple Mail Transfer Protocol) | 25 |
| FTP | ファイルの送受信 | FTP(File Transfer Protocol) | 20 |
| サーバーは正確には、そのコンピュータ上で常駐して動き続けているサーバープ ログラムであり、1台の機械が複数のサービスを提供することもある。PORTはコンピ ュータ(のOS)が提供する入出力の口であり、サーバープログラムごとに使用する標 準的なPORT番号が定まっている。 | | | |

問. このページ(<u>http://www.ugtop.com/spill.shtml</u>)で、WebブラウザがWebサーバーに送 信する情報を確認してみよう。

5. 通信速度

インターネット接続には様々形態(回線)がある。以下に代表的な形態とその速度 を掲げておこう。上では様々なサービスが行なわれているが、そのサービスと提供す るコンピュータをサーバーといい、サーバーに対し、そのサービスを受ける(要求す る)側を**クライアント**という。サーバーとクライアントは、上で述べたTCP/IP上で定め

れたそれぞれのプロトコル(通信規約)に従って交信を行なっている。

| サービス名 | 回線 | 通信速度(最高値) | 備考 |
|---------|----------|------------------|-------------|
| FTTH | 光ファイバー | 100Mbps | |
| VDSL | 建物内電話線 | 1 2Mbps、↓ 52Mbps | 建物までは光ファイバー |
| ADSL | 電話回線 | ↑2Mbps、↓10Mbps | 通常の電話と共用 |
| 携帯 | 無線 | 28.8Kbps~2Mbps | 方式により異なる |
| PHS | 無線 | 64Kbps | |
| LAN(有線) | Ethernet | 10Mbps~1Gbps | 規格により異なる |
| LAN(無線) | 無線 | 11Mbps~54Mbps | 規格により異なる |

(Kはキロ、Mはメガ、Gはギガ、bps(bit per second)は毎秒のビット数。↑は上り、↓ は下りの速度)

問.

1) インターネットの通信速度測定サイトを利用して、使用環境の通信速度を求めよ。 実際の状況により、上記の数値は達成されないし、測定の度に数値が異なるの が普通である。

2)通信速度が10Mbpsのとき、100MBのファイルのダウンロードに何秒かかるか。 通信速度の単位はbps(毎秒のビット数)であり、バイト数でないことに注意せよ。



7.1 インターネットの歴史 7.2 通信の仕組み

8 Webページの作成

この章のポイント

Webページ作成の基本を学び、マークアップ(markup)言語という考え方を学ぶ。

この章の構成

<u>8.1 HTML文書の基本構造</u>

<u>8.2 ホームページの作成</u>

<u>8.3 ホームページの登録・公開</u>

8. Webページの作成

8.1 HTML文書の基本構造 8.2 ホームページの作成 8.3 ホームページの登録・公開 8.1 HTML文書の基本構造

1. マークアップ言語

Webページの記述言語をHTML (Hyper Text Markup Language)という。Hyper Text と は、Webページのように、他のページへのリンクや画像などを含む文書をいう。また、ワープ ロのように見たままの文書が得られる WYSIWYG (What You See Is What You Get)方式と 異なり、HTMLでは文書の論理的構成や体裁の情報が埋め込まれたテキストをエディタで 作成し、ブラウザで表示させる。

このように副次情報をテキストに埋め込み、それを利用して整形・表示などを行なう方式 をマークアップ(markup)方式という。この方式は、ブラウザに通さないと表示結果を確認で きないという欠点を持つ一方、文書の編集が特定のソフトに依存せず行える、文書作成者 は細かな文書整形作業から開放され、章立てや節立て、箇条書きなどの文書の論理的構 成に集中できる、といった利点を持つ。

とくに、Webページの場合、それを表示するブラウザやウィンドウ・サイズは一定でないので、それらに依存した体裁をとるべきではないため、マークアップ方式が採用されている。 HTMLでは、文書の構成や体裁に関する情報はタグで指定される。タグは一般にくと> で

囲まれ、例えば開始タグ <zzz> と終了タグ </zzz> で囲まれた部分が、zzz要素であることを示す(タグは半角文字)。

2. .HTMLの構造

| HTMLファイル(すなわちホームページのソース)は、一般に次のような構造を持 | |
|--|---|
| つ。 | |
| | |
| <html></html> | |
| <head></head> | |
| <title>タイトル</title> | |
| | |
| <body></body> | |
| ▲× / ¬♪ ↓> | |
| | |
| | |
| | |
| ここで例えば、〈HTML〉・・・〈/HTML〉タグはこれに挟まれた部分がHTMLであるこ たまし、/TTTLE〉・・・〈/TTTLE〉はこの部分がタイトルである(ブラウザのタイトルバ | ٤ |
| を扱い、、IIILE/、IIILE/はこの部分がダイドルでのる(ノブブリのダイドルバ に表示される)ことを示す | |
| 他にタグでは次のような事柄が指定できる。 | |
| 可能な事柄 | |
| ページ屋性 背呂色 本文の文字色 リンクの色 | |
| | |
| | |
| | |
| 太子、 <u>ト稼</u> 、料体、竹巻+ 大きさ+ 色 の指定、 | |
| | |
| 見出し | |
| | |

| | 見出し2 | |
|---|--|--|
| | 見出し3 | |
| 文字属性 | 見出し4 | |
| | 見出し5 | |
| 文書の構造・整 形情報 | 見出し6 改行、センタリング、段落、左揃え・中央揃え・ 右揃えの指定 マーク(・)付き箇条書き、番号付き箇条書き、タ イトルつき箇条書き | |
| 挿入オブジェクト | リンク、ターゲット(他からのリンク先)、画像、横 罫線、表 | |
| またHTMLのタグには改行タグのように、終了タグが省略可能なものや存在しないものまある | | |

注意!

ホームページ作成上達の近道は、気にいったホームページのソースを調べ、その 真似(と種々の改良)をすることである。また、「ホームページ入門」、「HTMLタグ」な どのキーワードで、総合検索サイトから解説ホームページを検索し、自分のブックマ ークに登録しておくとよい。

実習1.

- 1. このページのソースを表示(右クリックで選択)させ、ページの表示と比較 せよ。
- 2. ホームページ作成の解説サイトを探し、気に入ったものを自分のブックマ ークに登録せよ。

コラム HTMLからXMLへ

HTMLは、タグを使って、テキストにその表示(整形)情報を付加する。これをさらに進めて、文書の構造情報や整形情報だけでなく、必要に応じ、意味情報も含めて自由にタグを付加できるようにするのが XML(eXtensible Markup Language)である。

例えば、「問題」、「例題」といった見出しと、重要事項とをともに太字で表示す るとしよう。HTMLでは、「太字で表示する」ということだけで、これらの区別はな い。XMLでは、これらに別のタグ(例えば〈midasi〉・・・〈/midasi〉、〈juuyou〉・・・ 〈/juuyou〉)を用意し、タグによる意味づけと、これらのタグで囲まれた要素は太 字で表示するという、表示(整形)指定とを分離する。こうすれば、例えば重要事 項だけは赤字で表示する、などといった変更にも柔軟に対応できる。

また、「山崎」という語はそれだけでは人名なのか地名なのかはっきりしない。 したがって、「山崎」をキーワードにインターネット検索をすれば(してみよ!)、地 名の「山崎」を含むページと人名の「山崎」を含むページ等が混在してリストアッ プされるだろう。これもXMLタグを使って、例えば<person>山崎</person>、 <place>山崎</place>などのように意味情報を付加することができ、かつ検索に それを利用する仕組み(セマンティックWeb)が整えば、上のような問題も解決さ れるだろう。

8.1 HTML文書の基本構造 8.2 ホームページの作成 8.3 ホームページの登録・公開

8. Webページの作成

<u>8.1 HTML文書の基本構造</u> 8.2 ホームページの作成 8.3 ホームページの登録・公開 8.2 ホームページの作成

1.alphaEdit(フリーソフト)の利用

ここでは<u>KraftWorks</u>が提供するフリーソフトの <u>alphaEdit</u> をダウンロードして利用する 方法を解説する。alphaEditではタグの入力がある程度自動化され、**半-WYSIWYG方** 式でホームページを作成 するモードと、HTMLを直接編集するソース編集モードを切り 替えながらホームページが作成できる。一般に、細かな(高度な)指定には、HTMLの ソースファイルを直接編集する(タグを書き込む)必要がある。

起動方法

<u>alphaEdit</u>のページからダウンロードした圧縮ファイルを解凍してできたフォルダ 「aled2003」をZドライブの適当な場所にコピーする。上記フォルダ内の alphaEDIT.exe をダブルクリックして起動する。

実習2.

alphaEditでホームページを作成・保存せよ。またページには氏名、学籍番号を記入すること。

2. リンクを張る

他のWebページへのリンクはalphaEditの機能を利用してリンクを挿入すればよい。(ホームページとして最初に開かれるファイルは、index.html というファイルであることに注意せよ。)

注:ファイル名には日本語(全角文字)を使用しないこと。特に画像など、ダウン ロードしたファイル名に注意せよ。

3. 画像の挿入手順

画像は以下の手順で挿入する。

- 挿入する画像ファイルをWebページと同じディレクトリ(フォルダ)に置く。その際、ファイル名を半角英数字にしておく。
 alphaEditでは、「画像挿入ウィンドウで画像をHTMLファイルと同じ場所に コピーする」をチェック
- 2. 登録するときも同じディレクトリ(フォルダ)に置くことに注意
- 3. 画像の挿入ボタンを押し、代替テキスト(画像が表示されない場合に表示 される)と挿入する画像を指定する
- 4. 画像の参照が相対参照になるように、ソース編集画面で、挿入部の内容 が例えば IMG height=60 src="Waterlilies.jpg" width=80 border=0 となるようにする。

コラム 相対参照と絶対参照

ファイルなどの資源を現在位置からの相対的な位置で指定する方式を相対参照、絶対的な位置で指定する方式を絶対参照という。

例えば、Webページ(HTMLファイル)が z:¥aaa フォルダにあり、画像ファイル image1.jpgが 同じフォルダにあるとすると、画像ファイルの相対参照は単に image1.jpg で、絶対参照は z:¥aaa¥image1.jpg である。

重要:本章で説明しているリンクや画像の挿入およびフレームの作成においては、インター ネット上にある他の画像やWebページを参照する場合を除いて、すべて相対参照にし、画 像やリンク先ファイル等の関連ファイルを原則同じフォルダに置かなければならない。 alphaEDITによるリンク・画像の挿入では絶対参照が作成されることがあるので、相対参照 にすることを忘れないこと。



4. フレームの作成

| フレームに分けて複数のページを表示させるには,例えば,page1.html, page2.html, page3.html が同じフォルダ内に用意されているとして, |
|---|
| <html> <head> <title>Frame Sample</title> </head> <frameset cols="220,*">←2列(第1列 220ドット分と残り第2列)に分割 <noframes> お使いのブラウザは、フレーム機能をサポートしていません。</noframes></frameset></html> |
| <frame name="frame1" src="page1.html"/> ←左側 (220ドット分) に表示される <frameset rows="50%,50%">←第2列を2段 (50%ずつ) に分割 <frame name="frame2" src="page2.html"/> ←第1段に表示される <frame name="frame3" src="page3.html"/> ←第2段に表示される </frameset> |

file://C:\Documents and Settings\yamasaki\デスクトップ\秀記\Siryou_JouhouKiso201... 2010/12/13

</html>

のようにする。ここで、<body>・・・</body>タグは使わないことに注意しよう。また、nameで指定されるフレーム名は、例えばリンク先のページを表示するフレームを指定するのに用いられる。以下のサンプルのHTMLコードを参考にせよ。
(alphaEditでも分割フレームを作成できるが、ここでは省略する。)

<u>フレームを使ったHTMLサンプル</u>

5. 著作権について

またホームページ作成に当たっては、情報倫理・著作権に十分注意すること。 ホームページは一般に公開されるもの(学内のみの公開であっても同様)である から、倫理的な問題に加えて、著作権にも留意しなければならない。また他人 のホームページから情報(データ)を取り出したときには、必ずその出所を明示 せよ。詳しくは ネットワークとコンピュータ(講習会資料)を参照せよ。

8.1 HTML文書の基本構造 8.2 ホームページの作成 8.3 ホームページの登録・公開

8. Webページの作成

<u>8.1 HTML文書の基本構造</u> 8.2 ホームページの作成 8.3 ホームページの登録・公開 8.3 ホームページの登録・公開

ー般に、コンピュータシステム内に個人が作成したファイル(データ)は、その個人の占有領 域に置かれる。そして、その占有領域は、本人(と管理者)以外は読み書きができないよう に、保護(アクセス制限)される。一方、Webページは、一般に公開することを目的として作 成されるデータであるから、その保護を外さなければならない。より正確には、本人は読み 書きできるが、本人以外は、読むことはできても、書く(編集する)ことはできないようにしな ければならない。

このとき、2通りの考え方があって、別のコンピュータ(サーバー)上の公開領域にアップロード(登録)する方法と、個人の占有領域中に公開用の特別のフォルダ(ファイル)を作成し、そのフォルダ(ファイル)へのアクセス制限を弱める(閲覧を許す)方法とがある。

1.一橋大学でのホームページ公開

登録とその確認方法は<u>情報基盤センターに説明</u>がある。ただし、学外からのアク セスは禁止されているので学内からのみの閲覧となる。また、ファイル名のデフォル ト値 はindex.htmlなので最初に見せるWebページ(ホームページ)のファイル名は index.html とすること。

コラム マルチユーザシステムとファイルのアクセス権

ー橋大学では、個人領域(Zドライブ)は、システム全体を管理するコンピュータ上にとられている。このように、一つのコンピュータ(上のファイルシステム)を複数のユーザで利用するシステムをマルチユーザシステムという。

マルチユーザシステムでは、他人に覗かれないように、個人のファイル(やフォルダ= ディレクトリ)は、その持ち主しかアクセスできないように厳重に管理されなければなら ない。

ー方、ホームページのように他人に公開するファイルは、誰でも読めるように設定して おかなければならないので、そのように設定された特定の場所に登録(アップロード) する形をとる。

また、Webページで画像等を表示(参照)するときには、Webページとともに画像ファイ ルもアップロードして公開領域におくともに、その参照を相対参照としなければならない 点に注意せよ。

2.情報倫理·著作権

情報倫理·著作権

ホームページは一般に公開されるもの(学内のみの公開であっても同様)であるから、倫理的な問題に加えて、著作権にも留意しなければならない。また他人のホームページから データを情報(データ)を取り出したときには、出所を明示せよ。詳しくは **利用の手引き**を参 照せよ。

実習3.実習2で作成したホームページを登録せよ。

登録したら、山崎の授業ホームページからたどれることを確認せよ。うまくいかないときは、TAまたは山崎に相談すること。



課題

- ネチケットのホームページなどを参考に、インターネットおよび電子(携帯)メールにおける情報倫理・マナーを解説するホームページを作成せよ。その際、著作権についての十分留意すること。
- 作成したホームページが、山崎の<u>授業ホームページ</u>(情報教育棟ホームページ⇒講義時間割⇒この授業 でたどれる)から たどれることを確認すること。

山崎秀記 <u>cj00059@srv.cc.hit-u.ac.jp</u>

8.1 HTML文書の基本構造 8.2 ホームページの作成 8.3 ホームページの登録・公開

9 表計算ソフト(Excel)

この章のポイント

表計算ソフトは、会社の事務・経理の処理から、家庭や個人の家計簿にいたるまで 幅広く利用される。さらに統計的処理や意思決定にも応用される。本章では、表計 算ソフトEXCELを例に、その多様な機能を体験してみよう。

この章の構成

<u>9.1 表計算ソフトの仕組み</u>

<u>9.2 基本操作</u>

<u>9.3 最適化分析</u>

9. 表計算ソフト

9.1 表計算ソフトの仕組み 9.2 基本操作 9.3 最適化分析 9.1 表計算ソフトの仕組み

1.基本原理

表計算ソフトは、2次元の表(**ワークシート**という)の各セル(ます)に値や数式を入 力すると、それらを自動的に計算するものである。セルには、横にA,B,C,・・・、縦に1, 2,3,・・・の番号が振られており、その組み合わせで、A2、B5、などのように表す。ま た、ワークシートの集まり、すなわち Excelが扱う対象を **ワークブック**という。

実習1:エクセルの計算機能を利用して、摂氏→華氏変換を実現しよう。

華氏= 摂氏×9/5+32である。

- 0. エクセルを起動する。
- シート1のセルA1に文字列「摂氏入力」、セルB1に文字列「華氏出力」、セルB2に式「=A2*9/5+32」を入力せよ。この結果、A2に入力された数値(摂氏温度)が華氏温度に変換されてB2に表示される。
- 2. セルA2にいくつかの数値(例えば20,35,40等)を入力して、これが正しく働くことを確かめよ。
- 実習2 ワークシートの適当な場所に実習1と同様に、華氏→摂氏変 換機能を実現し、正しく働くことを確かめよ。
 - 1. 実習1と同様に、華氏入力のセルと摂氏出力のセルを決め、華氏か ら摂氏を求める式を考えよ。
 - 2. "1"で考えた式を摂氏出力のセルに入力せよ。
 - 3. 華氏入力のセルに数値を入力して、これが正しく働くことを何回か確 かめよ。

2.数式における関数の利用

セルに「=」を入力するとEXCELは数式入力モードになり、数式に用いる関数を選 択するメニューが使える。「=」を入力するとウィンドウのどこが変化するか、確認せ よ。

和を求める関数「SUM」などでは、変数としてセル範囲を指定できる。セル範囲は、 長方形区域内のすべてのセルの集合で、セル範囲を指定するにはその左上のセル と右下のセルを: (コロン)で区切って、A4:E8のように指定する。

また、「数式」では単なる数の計算だけでなく、文字列処理を含む非常に多くの関数を利用できる。詳しくはEXCELヘルプで「数式を入力する」を参照せよ。

9.1 表計算ソフトの仕組み 9.2 基本操作 9.3 最適化分析

9. 表計算ソフト

<u>9.1 表計算ソフトの仕組み</u> 9.2 基本操作 9.3 最適化分析

- 9.2 基本操作
- 1.連続データのオートフィル

| ビルにく力するときにはオートフイル機能が使える。この機能を利用して等差数が 等比数列の(予測値の)入力が簡単に行えるので、数表(例えば各温度における え、華氏の変換表)を作成したり、データを予測したりできる。 にくはEXCELヘルプの「行または列内のデータを予測したりできる。 にくばEXCELヘルプの「行または列内のデータを予測したりできる。 とくばEXCELヘルプの「行または列内のデータを予測したりできる。 とくばEXCELへルプの「行または列内のデータを予測したりできる。 とくばEXCELへルプの「行または列内のデータを予測したりできる。 とくてまならる関数SUMを利用する。 ことに、和を求める関数SUMを利用する。 ことに、和を求める関数AはSUMだった。では平均を求める関数 Act? 3 - ビー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 | <text><text><text><list-item><list-item><list-item></list-item></list-item></list-item></text></text></text> | 1 | 131 | における | 1月、2月 | l、 <u>3</u> 月な | どの連続 | したデー | タを一列 | (一行)の連 | 続し |
|--|---|----------------|--|--|--|--|--|--|---|---|---|
| 第12.83700で実施設)を作成したり、データを予測したりできる。 第4.50で変換表)を作成したり、データを予測したりできる。 第2.50で変換表)を作成したり、データを予測したりできる。 第2.500でするしたのルブの「行または列内のデータをコピーする」を参照せよ。 第2.500 第2 | <text><text><text><text><text><text><list-item><list-item></list-item></list-item></text></text></text></text></text></text> | セルにノ | した |)するとき | には オー | トフィル | 筬能 が使 | える。こ0 ので 粉目 | D機能を | ·利用して等君 ギタ 泪 座にも | との とします こうしん こうしん こうしん こうしん こうしん こうしん こうしん しんしん いっしん しんしん しんしん しんしん しんしん しんしん しん |
| <text><text><text><text><text><text><list-item><list-item><text></text></list-item></list-item></text></text></text></text></text></text> | <text><text><list-item><list-item><list-item><list-item><list-item></list-item></list-item></list-item></list-item></list-item></text></text> | 守比奴? 〔氏」一苗。 | バリク | ノア別順の赤協主、 | 100/八月 | か間里 たい デ | <1]んつ(| ひじ、奴名 海川 たいっ | 又()別んし でキス | よ谷洫皮にの | 110 |
| <text><text><text><text><text><text><list-item><list-item></list-item></list-item></text></text></text></text></text></text> | <text><text><text><text><text><text><list-item><list-item></list-item></list-item></text></text></text></text></text></text> | ジンマギュ ジンノナロ | -_U = \/ (| ノ友侠仪。 | プの[/= == | ンごり、) ヒナン1十万川 | ーラぞっ? カのデー | 別したり 内 た コピ - | してつ。 オス は | おおまた | |
| <text><text><text><text><text><text><text><text></text></text></text></text></text></text></text></text> | <text><text><text><text><text><text><text><text><text></text></text></text></text></text></text></text></text></text> | | | | 507,119 | トノニームクリト | - ((UP | 7616- | - 9 ' @]2 | | |
| 1. 下図のセル範囲 A4:E8 のようなデータを入力し、1月の総支出額が 求るように数式を入力せよ。 上と:和を求める関数SUMを利用する。 2. 食費の月平均を求める関数名はSUMだった。では平均を求める関数 ない? 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 やり欄がすべて求まるようにせよ。 正 要:コピーされた数式はもとの数式とどう違っているか? 3. コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 ※ ************************************ | 1. 下図のセル範囲 A4:E8 のようなデータを入力し、1月の総支出額が x5 x5 | 実習: | 3 | | | | | | | | |
| 1. 「「私日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日 | 1. Pixel Arke Duby, and a set of the se | 1 | _ ہ | 図のすこ | . 筋囲 ∧ / | 1·E0 | うたデー | なた 入 力 | 180 | り総支中対が | ĩ |
| Lント:和を求める関数SUMを利用する。 食費の月平均を求めよ。 とント:和を求める関数名はSUMだった。では平均を求める関数 名は? 1ピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 重要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Microsoft Excel = Book1 アナルの 編集(表示の 排入の 書式の ツールの テータの ウンドワ) ー す メ バノクの ー す まい キー を 押す。 Multi * × × ケ = 1 摂氏入力 糞氏出力 糞氏入力 摂氏出力 4 変計薄 食費 光熱費 書籍 総支出 5 1月 42000 16500 0000 17800 9600 17800 9600 17800 9600 18960 17800 9600 17800 9600 19960 1000 1000 1000 1000 1000 1 | と)::和を求める関数SUMを利用する。 2、食費の月平均を求めよ。 と)::和を求める関数名はSUMだった。では平均を求める関数 なに。 3. ゴビー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。。 正 要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 渡た取り消すには ESCキーを押す。 ※ Microsoft Excel = Book ※ MMULT * * * * * ※ # # # # # # # # # # # # # # # # # # # | 1. | 求 | (まるよう) | こ数式を | 入力せよ | | | | の心又田留の | • |
| 2. 食費の月平均を求めよ。 とゝh:和を求める関数名はSUMだった。では平均を求める関数 名は? 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 正 要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Wierosoft Excel = Book1 Pr/ルビ 編集() 表示の 排入() またの ツール〇 データの ウルドウ() MMULT Y イルビ 編集() 表示の 排入() またの ツール〇 データの ウルドウ() MMULT Y イルビ 編集() 表示の 排入() またの ツール〇 データの ウルドウ() Y イルビ 編集() 表示の 挿入() またの ツール〇 データの ウルドウ() MMULT Y イルビ 編集() 表示の 挿入() またの ツール〇 データの ウルドウ() Y イルビ 編集() 表示の 挿入() またの ツール〇 データの ウルドウ() Y イルビ 編集() 表示の 前入() なん シーク ジード() Y イルビ 編集() 表示の 前入() なん シーク ジービ () Y イルビ () Y イン () Y イン () Y イン () Y イルビ () Y イン () Y イン () Y イルビ () Y イン (| 2. 食費の月平均を求めよ。 とゝh:和を求める関数名はSUMだった。では平均を求める関数 名は? 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 正 要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 ※ Microsoft Excel = Bookt ※ プイルを 編集を 表示の 排入 またの データ ウンドウッ パンクック・レンドウッ パンクック・レンドウッ パンクック・レンドウッ パンクック・マンドウッ パンク・マンドウッ パンクック・マンドウッ パンクック・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウェーマン パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ マンドウッ パンク・マンドウッ パンク・マンドウッ パンク・マンドウッ マント・マン マン・マン マン マン マン マン マン マン マン マン マン マン マン マン マ | | - | ヒント:利 | 旧を求め | る関数SI | UMを利 | 用する。 | | | |
| とっと:和を求める関数名はSUMだった。では平均を求める関数 名は? 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 正 要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 パレロ データの ウレビウ(の) クレビウ(の) クレ(の) クレ(0) クレ(0 | <text><text><text><text><text></text></text></text></text></text> | 2. | 食 | 費の月平 | □均を求め | めよ。 | | | | | |
| Alt? 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 ッ均欄がすべて求まるようにせよ。 重要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 ● Microsoft Excel > Bookt ● アイルや 編集9 表示の 排入9 書式の ツールの データの ウインドウω ● パングサーク キャン キャン 美 二 日 ● ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | Ait? 1. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月、 ア均欄がすべて求えるようにせよ。。 重 要:コピーされた数式はもとの数式とどう違っているか? コピー方方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すしては ESCキーを押す。 ● Microsoft Excel = Book1 ● アイルビ 編集U 表示U #A またU アールU データU ウィンドウW ヘルプU ● アイルビ 編集U 表示U #A またU アールU データU ウィンドウW ヘルプU ● アイルビ 編集U 表示U #A またU アールU データU ウィンドウW ヘルプU ● アイルビ 編集U 表示U #A またU アールU データU ウィンドウW ヘルプU ● 日本 日 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | | | ヒント:利 | 1を求め | る関数名 | はSUM ^た | どった。で | は平均を | を求める関数 | |
| 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月平均欄がすべて求まるようにせよ。 重要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 ※ Microsoft Excel = Book1 ※ Microsoft Excel = Book1 ※ Microsoft Excel = Book1 ※ Fixed **** ******************************* | 3. コピー機能を利用して、各月の総支出、光熱費、書籍、総支出の月 平均欄がすべて求まるようにせよ。 重要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 ※ Microsoft Excel - Booki *********************************** | | 名 | は? | | | | | | | |
| 平均欄がすべて求まるようにせよ。 重要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Microsoft Excel = Book1 ・ ************************************ | 平均欄がすべて求まるようにせよ。 重 要:コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 べいていび (()) () () () () () () () () () () () (| 3. | \square | ピー機能 | を利用し | て、各月 | の総支出 | 出、光熱費 | と、書籍、 | 総支出の月 | |
| 重 要:=ピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Microsoft Excel = Book1 アイル(P) 編集(P) 表示(P) 排入(P) 書式(P) ツール(P) データ(P) ウル)ドウ(P) NJ7(P) アイル(P) 編集(P) 表示(P) 非入(P) 書式(P) ツール(P) データ(P) ウル)ドウ(P) NJ7(P) アイル(P) 編集(P) 表示(P) キス(P) キス(P) ウル)ドウ(P) NJ7(P) アイル(P) 編集(P) 表示(P) キス(P) きょう(P) ウル) NJ7(P) アイル(P) 編集(P) 表示(P) キス(P) キス(P) ウル) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) NJ7(P) | 重 要: コピーされた数式はもとの数式とどう違っているか? コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 「Microsoft Excel = Bookt 「アイルの 様本の またの アールの テータの ウィンドウン ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ | | Ŧ | 均欄がす | レベて求る | まるように | こせよ。 | | | | |
| コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 [●] アr/ルビ 編集(D 参示(V 排入(P 書式(D) ツール(D) データ(D) ウルンドウ(V) | コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 | | | 重要: | コピーさ | れた数式 | はもとの | 数式とど | う違って | いるか? | |
| コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Microsoft Excel = Book1 ディル化 編集 表示い 挿入 またの ツール テータの ウインドウツ ヘルプロ ・ * * * * * * * * * * * * * * * * * * * | コピー方法は、今までと同様(選択して右クリック)である。また、セルの 選択を取り消すには ESCキーを押す。 Microsoft Excel = Book1 デアイル① 編集① 表示② 挿入① 書式② ツール① データ② ウィンドウジ ヘルプサ) ・・・ × × × を = A B C D E F G 1 摂氏入力 華氏出力 華氏入力 摂氏出力 2 25 77 3 な計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000目 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 × + * N Sheet1 (Sheet2 (Sheet3/ | | | | | | | | | | |
| 選択を取り消すには ESCキーを押す。 ■ Microsoft Excel = Book1 ■ ファイル (D) 編集(D) 表示(D) 挿入(D) 書式(D) ツール(D) データ(D) ウル)ドウ(D) | 選択を取り消すには ESCキーを押す。 ■ Microsoft Excel = Book1 ■ ファイル (D) 編集(D) 表示(V) 挿入 (P) 書式(D) ツール (D) データ(D) ウルドウ(V) & × | コピ | `— ` | 方法は、 | 今までと | 同様(選 | 択して右・ | クリック)で | である。き | また、セルの | |
| Microsoft Excel = Book1 アッイル① 編集① 表示② 挿入② 書式② ツール① データ② ウインドウω ヘルブ(屮) ー 部 × ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● | Microsoft Excel = Book1 アナイル(D) 編集(D) 表示(D) 挿入(D) また(D) アール(D) データ(D) ウルンドウ(D) ヘルブ(D) ー 6 × ー 6 × ー 6 × ー 6 × ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● | 166 1 1 - | | a state to a | | | | | | | |
| 割 ファイル(E) 編集(E) 表示(W) 持入(P) 書式(Q) ツール(D) データ(Q) ウィンドウ(W) | 割 ファイル(E) 編集(E) 表示(M) 排入(P) 書式(M) フール(E) データ(E) ウィンドウ(M) | 进 (1) | を耳 | なり消すに | こは ESC | キーを押 | す。 | | | | |
| All 70B | へルブ(4) - 6 × □ ● 目 包 ● 1 兆 电 ○ + 色 Σ + 4 和 値 ⑦ * 三 四 ○ + * MMULT • × √ & = A B C D E F G 1 摂氏入力 華氏出力 華氏入力 摂氏出力 2 25 77 3 - </td <td>进打</td> <td>を耳 I M</td> <td>対り消すに icrosoft Exe</td> <td>こは ESC cel - Book1</td> <td>キーを押</td> <td>す。</td> <td></td> <td></td> <td></td> <td></td> | 进 打 | を耳 I M | 対り消すに icrosoft Exe | こは ESC cel - Book1 | キー を押 | す。 | | | | |
| Image: Model (A) A B O ● E F G * MMULT * × √ A = F G • * | Image: Model and | 进扒 | | なり消すに ierosoft Exe ファイルE 編 | こは ESC cel - Book1 課値 表示 | キーを押 〕 ◎ 挿入Φ | す。 春式(2) :: | ッール田 デー! | 9(D) ウィン | | |
| A B C D E F G 1 摂氏入力 華氏出力 華氏入力 摂氏出力 2 25 77 3 3 | A B C D E F G 1 摂氏入力 華氏出力 華氏入力 摂氏出力 2 25 77 3 3 | 選択 | | 文り消すに icrosoft Ex ファイル(E) 編 フ%U | こは ESC cel - Book1 雑心 表示 | キ −を押 [●] 挿入Φ | す。 書式(2) * | ッール田 デー | タ(Q) ウィン | □□区 応づ型 - ff × | |
| A B C D E F G 1 摂氏入力 華氏山力 華氏入力 摂氏出力 2 25 77 3 3 | A B C D E F G 1 摂氏入力 華氏出力 華氏入力 摂氏出力 2 25 77 3 3 - - - 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 = 6 2月 42000 16500 0 - 7 3月 38000 17800 9600 - - 9 - - - - - - 10 - - - - - - # * * * * * * * | 選択 | | 文り消すに ierosoft Ex ファイル(D 編 プロ) 学園 18 | Cit ESC pel - Booki 課(E) 表示 愛 Dal よ | キーを押 ◎ 挿入Φ ■ い・ | す。 書式の * 後 Σ・ᢓ | yール① デー: え: <u> </u> ② | אשט (שפ ערי (שפ ויין ד | 「日本」 「「日本」 「日本」 「日本」 「日本」 | |
| 2 25 77 3 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 = | 2 25 77 3 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 = | 選択: | | 文り消すに icrosoft Ex ファイル(D 編 ブ(B) MMULT | には ESC cel - Booki 課(E) 表示 語 Da メ - × ✓ | キーを押 ◎ 挿入Φ ■ い・ <u>&</u> = | す。 書式(2) : 後 Σ・2) | yール① デー! え! ② | ¢© າ~) ເ∵ື່≣≣ | 「□図 ^約 型 - ff × 翻 ♪・ ? | |
| 3 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 日 6 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 | 3 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 1 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 | 選択 | | 文り消すに icrosoft Ex ファイル(D 編 ブ(D) | には ESC cel = Booki 葉(E) 表示 母 G、 メ マスマ 日 華氏出力 | キーを押 ◎ 挿入Φ 喩 い・ ゑ = C | す。 書式(2) : | yール① デー: ミ 論 ② E 摂氏出力 | າງ ອີງ ອີງ ອີງ ອີງ ອີງ ອີງ ອີງ ອີງ ອີງ ອີ | ► ► × = = × = = × = = × | |
| 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 日 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 | 4 家計簿 食費 光熱費 書籍 総支出 5 1月 35000 17000 5000 = 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 10 14 ◆ M\Sheet1/Sheet2/Sheet3/ ↓ 編集 | 選択 | | 文り消すに icrosoft Ex ファイルの 編 プロ) 「日 MMULT A 摂氏入力 25 | には ESC cel - Book1 課(E) 表示 の (A) A ・ × V B 華氏出力 77 | キーを押 ◎ 挿入Φ ◎ ∽・ を = ○ | ま式(2) : 書式(2) : (名) Σ ・ 2) 単氏入力 | ッール① デー 31 🏙 団 <u>E</u> 摂氏出力 | עלי שני געילי (ו≣ F | F7₩ - f × B ♪ · * | |
| 5 1月 35000 17000 5000日 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 10 減集 「 ▲ ◆ N\Sheet1/Sheet2/Sheet3/ ▲ | 5 1月 35000 17000 5000目 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 ば・・ M\Sheet1/Sheet2/Sheet3/ ・ | 进伏: | | 文り消すに icrosoft Ex ファイル① 編 プピ) ■ ■ 電 MMULT A 摂氏入力 25 | には ESC cel - Book1 課(E) 表示 る (A) メ マ × ✓ 日 華氏出力 77 | キーを押 ◎ 挿入Φ ■ い・ <u>&</u> = C | ま式(2) * 書式(2) * 優 Σ・2) 単氏入力 | ッール① デー: - え ② E 摂氏出力 | ¢© າເບ າ ເ∵ື≣ F | F7₩ - F × B ♪ · ? G • | |
| 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 ★★ ₩ 4 ★ ₩\ <u>Sheet1/Sheet2/Sheet3/</u> 4 ★ 編集 | 6 2月 42000 16500 0 7 3月 38000 17800 9600 8 月平均 9 10 10 14 ↓ ▶\ <u>Sheet1/Sheet2/Sheet3/</u> ↓ 編集 | 进代: | を耳 M ペル 1 2 3 4 | 文り消すに Terosoft Ex ファイル(E) 編 ブ(E) 副 副 MMULT A 摂氏入力 25 家計簿 | には ESC cel - Book1 課(E) 表示 る (A) み マ × √ B 華氏出力 77 食費 | キーを押 ◎ 挿入Φ <u>◎ </u> ● | ま式Q) 3 ● Σ • 2 ■ 単氏入力 書籍 | ッール① デー: ス 値 ② <u>E</u> 摂氏出力 総支出 | ზდ ბით I ? I≣ F | 「□図 「つ ¹⁰⁰ - ff × 留 ♪・ ? G ▲ | |
| 7 3月 38000 17800 9600 8 月平均 9 10 10 10 10 10 10 10 10 10 10 10 10 10 | 7 3月 38000 17800 9600 8 月平均 9 10 IV ↓ N\Sheet1/Sheet2/Sheet3/ ↓ ↓ ↓ ↓ ↓ ↓ ↓ | 进伏: | を耳 3 M 1 2 3 4 5 | 文り消すに Terosoft Ex ファイル(E) 編 ブ(B) 副 副 MMULT A 摂氏入力 25 家計簿 1月 | には ESC rel = Book1 課(E) 表示 通 2、 素示 一 × √ B 華氏出力 77 食費 35000 | キーを押 ◎ 挿入Φ ■ い・ <u>&</u> = C 光熱費 17000 | ま式Q) 3 ● Σ ・ 2 ■ 単氏入力 書籍 5000 | yール① デー: ス 値 ② <u>E</u> 摂氏出力 総支出 | 9© ბი∪ I ? I≣ F | 「□図 「つW」 - ff × 日 ・ ・ ・ ・ ・ ・ ・ ・ | |
| 8 月平均 9 10 4 + H\ <u>Sheet1</u> /Sheet2/Sheet3/ | 8 月平均 9 10 H 4 + H\Sheet1/Sheet2/Sheet3/ 4 | 进伏: | を耳 3 M 2 M 1 2 3 4 5 6 | 文り消すに Terosoft Ex ファイル(E) 編 ブ(B) 副 副 MMULT A 摂氏入力 25 家計簿 1月 2月 | には ESC rel = Book1 課(E) 表示 参 2、参 マ × √ B 華氏出力 77 食費 35000 42000 | キーを押 ◎ 挿入Φ ■ い・ <u>を</u> = C 光熱費 17000 16500 | ま式Q) : ● Σ ・ 2 ■ 単氏入力 書籍 5000 0 | yール① デー: え 値 ② 超 | タロ ウイン ・* ■ F | □○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○ | |
| 9 10 H 4 + H <u>Sheet1</u> /Sheet2/Sheet3/ 4 | 9 10 14 + H\Sheet1 /Sheet2 /Sheet3 / 【 / 】 | 进伏: | を耳 図 M 別 1 2 3 4 5 6 7 | 文 り消すに icrosoft Ex ファイル(E) 編 ブ(U) 回 記 MMULT A 摂氏入力 25 家計簿 1月 2月 3月 | には ESC cel = Book1 課(E) 表示 通 3、 ▼ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ い・ <u>を</u> = C 光熱費 17000 16500 17800 | ま式Q) : ● Σ ・ 2 ■ 単氏入力 書籍 5000 0 9600 | yール① デー: え 圖 図 <u>E</u> 摂氏出力 総支出 | 9D ウイン ?!≡ F | □○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○ | |
| 10 4 4 + H \Sheet1 /Sheet2 /Sheet3 / | 10 14 + + N <u>Sheet1</u> /Sheet2/Sheet3/ • • • • • • • • • | 进伏: | を耳 4 M 1 2 3 4 5 6 7 8 | 文 り消すに icrosoft Ex ファイル(E) 編 ブ(U) 副 副 MMULT A 摂氏入力 25 家計簿 1月 2月 3月 月平均 | は ESC cel = Book1 雑(E) 表示 参 2. メ マ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ い・ <u>を</u> = C 光熱費 17000 16500 17800 | ま式Q) : ● Σ ・ 2 ■ 単氏入力 書籍 5000 0 9600 | yール① デー: え 圖 圖 ② 唐 摂氏出力 総支出 | 9D ウイン ?!≡ F | □○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○ | |
| IN → N\Sheet1/Sheet2/Sheet3/ IN N | i | 进伏: | を耳 4 ヘル 1 2 3 4 5 6 7 8 9 | 文 り消すに icrosoft Ex ファイル(E) 編 ブ(U) 副 副 MMULT A 摂氏入力 25 家計簿 1月 2月 3月 月平均 | には ESC rel = Book1 課(E) 表示 参 (A) 参 マ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ い・ <u>を</u> = C 光熱費 17000 16500 17800 | ま式Q) : ● Σ ・ 2 ■ 単氏入力 書籍 5000 0 9600 | yール① デー: ス↓ 値 ② 唐 摂氏出力 総支出 | 9D ウイン ?*!≣ F | 「□図 「つW - ff × 日 ・ ・ ・ ・ ・ ・ ・ ・ | |
| 編集 | 編集 | 进伏: | を耳 ペル 1 2 3 4 5 6 7 8 9 10 | 文 り消すに icrosoft Ex ファイル(D 編 ブ(U) 副 副 MMULT A 摂氏入力 25 家計簿 1月 2月 3月 月平均 | は ESC rel = Book1 課(E) 表示 参 (A) メ マ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ い・ <u>を</u> = C 光熱費 17000 16500 17800 | ま式Q) : ● Σ ・ 2 ■ 単氏入力 書籍 5000 0 9600 | 9-ル① デー: え 圖 ② <u>E</u> 摂氏出力 総支出 | 9D ウイン ?!≡ F | 「□区 「つW」 - 冊 × 日 - 冊 × 日 - 冊 × 日 - 冊 × 日 - 冊 × | |
| | And the second sec | 进伏: | を 1 2 3 4 5 6 7 8 9 10 11 11 11 11 11 11 11 11 11 | 文 り に た で の で し で し で し で し で し で し で で し で し で で し し で し で し し で し で し し で し し て て し し て し し て て し し し て つ し し て つ し し て つ し し て つ し し て つ し こ ち つ つ つ つ つ つ つ つ つ つ つ つ つ | は ESC cel = Book1 課(E) 表示 通 (A) メ マ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ ∽・ <u>☆</u> = C 光熱費 17000 16500 17800 | ま式Q) : 書式Q) : ● Σ・ 2 単氏入力 書籍 5000 0 9600 | 9-ル① デー: え 値 ② 度 摂氏出力 総支出 | タロ ウイン ・* ■ F | □ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ | |
| | | 进机 | を耳 ペル 1234 5678910 | 文 り た た で の で し し で し し で し し て て し し て つ し し て つ し し て つ し し て つ し し て つ し こ ち つ し こ ち つ つ つ こ ち つ つ つ つ こ ち つ つ つ つ つ つ つ つ つ つ つ つ つ | は ESC cel = Book1 課(E) 表示 通 (A) メ マ × √ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ ∽・ ☆ = ○ 光熱費 17000 16500 17800 Sheet3/ | ま式Q) : 書式Q) : ● Σ・会 単氏入力 書籍 5000 0 9600 | 9-ル① デー: え 値 ② 度 摂氏出力 総支出 三 | タロ ウインI I ? I I I I I I I I I I I I I I I I I | □ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ | |
| | | 进伏: | を M A A A A A A A A A A A A A | 文 り た た で の で し し て て し し て て た た た た た た た た た た た た た | は ESC cel = Book1 葉(E) 表示 毎 (A) メ マ × ✓ B 華氏出力 77 食費 35000 42000 38000 | キーを押 ◎ 挿入Φ ■ ∽・ ☆ = C 光熱費 17000 16500 17800 Sheet3/ | 書式Q) : (4) Σ • 2 (5) Σ • 2 (7) Φ (7) Φ<!--</td--><td>9-ル① デー: え 値 ② 度 摂氏出力 総支出 三</td><td>タロ ウインI I ?? ■ F</td><td>□ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○</td><td></td> | 9-ル① デー: え 値 ② 度 摂氏出力 総支出 三 | タロ ウインI I ?? ■ F | □ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ | |

2.関数

実習3で、総和を求める関数や平均を求める関数を使った。表計算ソフトには、これ以外にも多くの関数が用意されており、インターネットからもその使い方を知ることができる。例えば、「EXCEL関数」で検索してみよ。

3.相対参照と絶対参照

重要:相対参照と絶対参照

実習3で、例えば食費の月平均を求める式がそのまま光熱費の平均欄にコピーされたとすれば、食費の平均値が表示される筈であるのに、光熱費の平均値が(正しく?)表示される。どういう仕組みになっているのだろうか?数式のコピー操作が期待した(?)通りに働くのは、数式中で使われているセルへの参照が、数式のあるセルからの相対位置(位置関係)によって定まっているからで、このような参照を相対参照という。したがって、あるセルの数式を他のセルにコピーするとそれに伴って相対参照セルの番地も変わる。

しかし、これでは都合が悪い場合もある。数式内のセル参照を、数式のある位置に 関わらず絶対的なもの(絶対参照)にしたい場合は、セルを指定する列記号(A、B、 …)や行番号(1、2、…)の前に\$記号をつける。例えば、\$E5とすれば、行番号5は相 対参照なので別のセルにコピーするとそれに伴って値が変化するが、列記号Eは絶 対参照なので、別のセルにコピーしても値は変化しない。

実習4

総支出における各支出項目の占める割合(%)を、各月ごとに求めた別 表を、次の手順でワークシートの適当な場所に作成せよ。

- 1. 実習3で作成した表全体をワークシートの適当な場所にコピーせよ。
- 1月の食費欄の数値データを、総支出に占める割合を求める式に書 き換えよ。
- "2"で作成した式を、各月の各項目の欄にコピーせよ。
 ヒント:2)で作成した数式のセル参照の一部を絶対参照にしないと 正しく働かない。

4.グラフ表示

グラフ表示したいデータをドラッグ操作などで選択した後、グラフボタンまたはメ ニューの挿入→グラフで、グラフウィザードを表示して、適切なグラフを選ぶ。項 目名もあわせて選択すれば、自動的に項目名が表示される。 また、円グラフなどでは、系列リスト中のトップにある系列のみが表示されるの で、他の系列を削除するとよい。

実習5

実習課題3および4で求めた表を、それぞれ適切なグラフを用いてグラフ表示せよ。

5. グラフの適性

データの意味・特徴や、グラフで示したい情報によって、適切なグラフを選択する必要 がある。それぞれの適性について、以下の表におおまかにまとめておこう。

| | データの単複 | 数値比較 | 値の推移 | 構成比 | 値の分布 | 値の相関 |
|----------|--------|------|------|-----|------|------|
| 棒グラフ | 単/複 | 0 | Δ | | | |
| 積み上げ棒グラフ | 複 | | Δ | 0 | | |

| 単/複 | Δ | 0 | | | |
|-----|--|---|---|---|---|
| 複 | Δ | 0 | Δ | | |
| 単 | | | 0 | | |
| 単/複 | | | | 0 | 0 |
| 単/複 | 0 | | | | |
| | 単/複 複 単/複 単/複 単/複 単/複 | 単/複 △ 複 △ 単 単/複 単/複 O | 単/複 △ ○ 複 △ ○ 単 - - 単/複 - - 単/複 ○ - | 単/複 △ 〇 複 △ 〇 △ 単 「 ○ ○ 単/複 「 □ □ 単/複 ○ □ □ | 単/複 Δ O Image: Constraint of the system 複 Δ O Δ 単 Image: Constraint of the system O 単/複 Image: Constraint of the system O 単/複 O Image: Constraint of the system |

<u>9.1 表計算ソフトの仕組み</u> 9.2 基本操作 9.3 最適化分析
9. 表計算ソフト

<u>9.1 表計算ソフトの仕組み</u> 9.2 基本操作 9.3 最適化分析

9.3 最適化分析

1. ゴールシーク

目的の値(ゴール)を実現する変数値を求めるツールである。Excellで、ツール→ゴールシ ーク、として

現れるウィンドウで、数式入力セル、(数式入力セルの)目標値、変化させるセル(1つ)、を設定すると、

目標値を実現する変数値(の1つ)を求めてくれる。

| ゴール シーク | ? 🔀 |
|----------------------|---------|
| 数式入力セル(E): | |
| 目標値(⊻): | |
| 変化させるセル(<u>C</u>): | |
| ОК | キャンセル |

問. 新しいシート上で、C1セルに数式=(A1-B1)*(A1-1)を入力し、数式入力セル:C1、目標 値:0、変化させるセル:A1、としてゴールシークを実行せよ。A1には、方程式(A1-B1)*(A1-1)=0の解A1=B1、1のどちらかが求まるはずであるが、A1の初期値やB1の値を色々変え て、実験してみよ。

2. 最適化分析

ワークシート上の、目的となる数式セルの値を最適(最大、最少)にするような変数 セルの値を求めることを最適化分析といい、Excelではそのためにソルバーという機能がある。Excelのサンプルを利用して、ソルバー機能を体験してみよう。

実習6

- フォルダC:¥Program Files¥Microsoft Office¥Office11¥Samplesをフ ォルダごと個人領域(Zドライブ)中に作成した授業用フォルダーにコ ピーせよ。
- 2. *"*1"でコピーしたフォルダ中のSOLVSAMP. XLSファイルを(Excelで) 開け。
- 概要シート中の説明(以下の囲みの中)にそって、年間利益を最大にする四半期ごとの広告費を求めよ。
 この課題で重要なのは、まず理解できるまで指示をよく読み、指示通り一つ一つ忠実にやってみること。うまくいかなければ何度でも最初からやり直すこと。次に、理解できたと思ったら、応用をいろいろ試してみることである。 遠回りなようでもそれが結局最短の道である。
- 制約条件(広告費の合計)を種々変更して見込まれる最大利益を計算し、広告費に対する投資額を経営者になったつもりで決定せよ (各人の判断でよい)。

<u>注意!</u> ツールメニュー中にソルバーという項目がない場合は、

制約条件(U)

提出期限

内

容

| | ツール→アドインで、ソルバーアドインのボックスをチェッ クする。 |
|---|---|
| | 以下、作業の理解を深めるために、モデルおよび課題の説明を付け加えておこう。こ のモデルでは、各四半期ごとに、 純益=粗利ー総経費 =売上予測数×(製品価格ー製品コスト)ー(販促費+広告費+社内経費) |
| | =35×季節指標×(広告費+3000) ^{0.5×} (製品価格-製品コスト)-(販促 費+広告費+社内経費) という式で純益を予測している(この予測があたるとは限らない)。なお、記号 ¹ はベ キ乗を表すので、(・) ^{0.5} は(・)の平方根を意味する。 |
| | マンションションションションションションションションションションションションション |
| | 2)年間の純益(第1~4四半期の純益の和)を最大にする第1~4四半期の広告費を 求める(多変数最適化問題)。 実は、各四半期ごとに純益最大の広告費を求めることに等しいが、季節指標の 影響で各四半期の最適値が異なる。 3)年間広告予算を決めた上で、最適(純益最大)な広告費配分を求める(制約条件 |
| | けや間次日子年を八のたまで、最適(純血銀八)な次日貨配力を六のの(前前朱川 付き多変数最適化問題)。 4)モデルの妥当性や資金調達手段等の条件を考慮して、年間広告予算額を定め る。 |
| צ | /ルバー: パラメータ設定 🔀 |
| | 目標値: ●最大値(M) ●最小値(N) ●値(V): 0 閉じる |
| | 変化させるセル(B): |

追加(A)

変更(<u>C</u>)

削除(D)

2. 実習6(4)の結果(決定した広告投資額に対するもの)を1. で作

3. 2. の作業で行った事柄(判断)の理由を、Sheet2の適当な場所

4. 1.2.3. を行なった結果のExcelワークブックをWebclassから提出

このときコピー位置がずれないように注意せよ(絶対参照が働か

1

1. 実習1~5をEXCELワークブックのSheet1上で行え。

成したワークブックのSheet2にコピーせよ。

次回の授業開始時まで

なくなるので)。

に記入せよ。

オプション(0)

リセット(B)

ヘルプ(王)

せよ。

<u>9.1 表計算ソフトの仕組み</u> 9.2 基本操作 9.3 最適化分析

10 マクロプログラム(Excel)

この章のポイント

----操作の自動化. EXCELを例に-----

この章では、EXCELを例に、アプリケーションにおける操作の自動化の仕組みを学習する。一連の操作を記録し再生する仕組みと、さらに進んで、操作手順をプログラミングする方法を紹介する。

この章の構成

<u>10.1 マクロ</u>

<u>10.2 マクロプログラミング</u>

10. マクロプログラム

10.1 マクロ 10.2 マクロプログラミング

10.1 マクロ

WORDやEXCELなどのアプリケーションでは、マクロを使って一連の作業を自動化 できる。例えば、新しいマクロの記録を開始してから終了するまでに行われた一連の 作業は、記録されたマクロの実行によって、繰り返し実行できる。ここでは、クラス (仮に10人とする)分の点数データをもとに、その平均値と標準偏差を計算し、点数 を偏差値に変換する一連の手順をマクロに記録し、繰り返し実行させてみよう。

実習1.マクロの記録

エクセルを開いて、次の手順を実行せよ。操作を間違えると、間違えたままの操作が記録されるので、操作を間違えないように慎重に行なうこと。 注意:最初に、ツール→マクロ→セキュリティで、セキュリティレベルを中に 設定すること。これを忘れると、保存して再度実行させようとしたときにマクロの実行を拒否される。

- 1. ツール→マクロ→新しいマクロの記録
- 2. マクロ名を設定してOK
- 3. セルA1をクリックしてそこに 点数 と記入
- 4. セルB1をクリックしてそこに 偏差値 と記入
- 5. セルC2をクリックしてそこに 平均点 と記入
- 6. セルC4をクリックしてそこに 標準偏差 と記入
- 7. マクロの記録を停止
- 8. Sheet1のセルA2からA11に10人分の点数を記入する。
- 9. ツール→マクロ→新しいマクロの記録
- 10. マクロ名を設定してOK
- 11. セルC3をクリックしてそこに =AVERAGE(A2:A11) と記入
- 12. セルC5をクリックしてそこに =STDEVP(A2:A11) と記入
- 13. セルB2をクリックしてそこに =50+(A2-\$C\$3)*10/\$C\$5 と記入
- 14. セルB2を セルB3~B11にコピー
- 15. マクロの記録を停止

この結果、マクロ名を特に指定しなければ、Macro1とMacro2という2つのマクロにこれらの手順が記録される。

実習2. マクロの実行と内容の確認

- ツール→マクロ→マクロ→実習1で記録した最初のマクロ(Macro1) を実行する
- 2. Sheet2のセルA2からA11に10人分の点数を新たに記入する。
- 3. ツール→マクロ→マクロ→実習1で記録した2番目のマクロ(Macro2)
 を実行する
- 1から3の作業がうまくいかない場合には、記録したマクロを削除し、 実習1をやりなおすこと。
- 5. ツール→マクロ→マクロ→実習1で記録したマクロ(Macro1又は Macro2)を選び→編集で表示されるマクロの内容を見て、作業とそ

の記録との対応を確認せよ。 例えばC3セルIC=AVERAGE(A2:A11)と記入した操作に対応する Macro命令や B2セルIC=50+(A2-\$C \$3)*10/\$C\$5と記入した操作に対応する Macro命令は何か。 特に、セルの相対参照と絶対参照がマクロ中でどのように記録され ているかに注意せよ。

10.1 マクロ 10.2 マクロプログラミング

10. マクロプログラム

<u>10.1 マクロ</u> 10.2 マクロプログラミング 10.2 マクロプログラミング

マクロは、前節の実習2で確認したように、一連の作業をひとつのプログラムとして記録したものである。 ここでは、前節で作成したマクロをもとに、 (1)偏差値が40未満の点数セルは黄色で塗り、60以上の点数セルは青色、それ以外は緑で塗るVBAプログラムを作成し、 (2)さらに、クラスの人数が何人であっても大丈夫なように、プログラムを拡張することを目標とする。

1. 最初のVBAプログラム

実習3. VBAプログラミング

まず、前節で作成した二つのマクロ(Macro1、Macro2)を順に実行するマクロ(プログラム)を書いてみよう。s

- 1. ツール→マクロ→VisualBasicEditor
- 2. 現れたエディタに以下のVBAプログラムを記述せよ。
 - Sub Hensati() Macro1 Macro2 End Sub
- 2. 作成したマクロ(プログラム)を実行せよ。
 VBEditor内では「マクロの実行」、
 Excelからは ツール→マクロ→Hensati を選んで実行

2. 書式とプログラム例

以下のプログラム例は、偏差値が40点未満の点数セルは黄色で塗り、60点以上の点数セルは 青色、それ以外は緑で塗るVBAプログラムである。この例を参考にプログラム(マクロ)の書き方を 説明しよう。

書 式:マクロは次の形式をとる。VisualBasicEditor では、1行目の「Sub マクロ名()」を入力 すると自動的に最後の行「End Sub」が付け加わるので、気をつけよう。

Sub マクロ名() マクロの中身 End Sub プログラム例 Sub 成績判定()

```
300 成績刊20
'n はデータの個数
```

```
n = 10
Cells.Interior.ColorIndex = xINone
For i = 2 To n + 1
```

```
If (Cells(i, 2) < 40) Then
        Cells(i, 1).Interior.ColorIndex = 6
Elself (Cells(i, 2) > 60) Then
        Cells(i, 1).Interior.ColorIndex = 8
Else
        Cells(i, 1).Interior.ColorIndex = 4
End If
Next i
End Sub
```

3. 条件と繰り返しの記述方法

```
繰り返しの書き方
   For i = 2 To n + 1
     i=2,3,…,n+1に対しここを繰り返す
   Next i
   Do While (条件)
     条件が成り立っている間ここを繰り
     返す
   Loop
条件文の書き方
   If (条件1) Then
    条件1がなりたつときここを実行
   ElseIf (条件2) Then
    条件1が成立たなくて条件2が成立つときここを実行
   Else
    条件1,2が成立たないときここを実
    行
   End If
命令文の解説
   'n はデータの個数: 'で始まる行はコメント行で、実行されない。
   n=10:変数nに値10を代入する。
   Cells(i,j): i は縦方向、j は横方向の番号を表す。Cells(2,3) はC2セルのこと。
   Cells.Interior.ColorIndex = xlNone: 各セルの内部の色設定を無しにする。
   Cells(i, 1).Interior.ColorIndex = 6: A列の i 番目のセルの内部の色を黄色(6)
    番の色)に設定。
```

4. 実習

実習4. VBAプログラミング

1. ツール→マクロ→VisualBasicEditor



せよ。

5. 今日の課題

| 提出期 限 | 来週の授業時間まで |
|----------|---|
| 実習1~6 | で作成したマクロの内容(Macro1、修正後のMacro2、Hensati、修正後の成績判 |
| 定、成績処 | ユ理)をWebClassの提出欄にコピーして提出(採点ボタンをクリック)せよ。 |

<u>10.1 マクロ</u> 10.2 マクロプログラミング

11 データベース

この章のポイント

――最もよく使われるアプリケーション――

例えば、大学における学生データベースを考えてみよう。学生に関する情報は、 教務課、学生課、経理課、・・・等で必要となる。これらの部署が独立にデータを保 持していては、例えば同じ住所変更を複数の部署で行わなければならないという不 便さがあるだけでなく、一方で住所変更がされたのに他方ではまだ変更されていな いなど、データの整合性がとれなくなることも予想される。

データベース管理システム(DBMS、DataBase Management System)は、データを 一元的に管理し、様々な利用者(部署)の様々な要求(質問)に対して、保持データ を加工して応答するもので、コンピュータの事務的アプリケーションとしては、表計 算とともに最も一般的なシステムである。

データベースにおける主要な問題には、既存データベースに対して、望みの情報 を得るためにどのような要求を出せばよいかという「問い合わせ(クエリー)の問題」 と、効率よく、データを保存し予想される問い合わせに答えるためには、データをど のような構造で保存するかという「設計の問題」とがある。

この章の構成

11.1 データベースとは:サンプルデータベースの閲覧

<u>11.2 SQLによる問い合わせ</u>

<u>11.3 SQL演習</u>

11.4 データベースの設計

11. データベース

11.1 サンプルデータベースの閲覧 <u>11.2 SQLによる問い合わせ</u> <u>11.3 SQL演習</u> <u>11.4 データベースの設計</u> 11.1 サンプルデータベースの閲覧

現在の多くのデータベースソフトは、データを2次元のテーブル(表)の形で保存する関係 データベースである。テーブルの最上行には、各列の項目名が表示され、テーブルの2行 目以下の各行が、テーブルに保存されているデータ(項目ごとの値の組)で、これをレコード と呼ぶ。

実習1

Accessのサンプルデータベース North Wind 上で作業できるように、

- フォルダ C:¥Program Files¥Microsoft Office¥Office11 ¥Samples を フォルダごとZドライブにコピーせよ。以前Excelの時に行な った場合はやらなくてよい。
- Z:¥Samples¥Northwind.mdb をAccessで開け。
 (うまく行かない場合、上記ファイルをデスクトップにコピーしてから開け。)
- 3. このファイルを開きますか? には「開く」と答える。
- 4. 安全でない式をブロックしますか? には「いいえ」と答える。
- 5. 再度のセキュリティ警告にも「開く」と答える。
- Northwindを開いたら、最初のメッセージおよびメインスイッ チボードを閉じる
- 続いてオブジェクト欄の「テーブル」タブをクリックし、現れる 各種のテーブルを覗いてみよ。

まずは**運送会社テーブルを表示させて**みよう。テーブルの最上段に表示 された「運送コード」、「運送会社」、「電話番号」を、項目名(フィールド名)と いう。次に**運送会社テーブルのデザインを表示させて**みよう。テーブルの 各項目には、その項目に適するデータの型(値の集合)が定まっており、こ れがテーブルの構造(デザイン)を定める。

実習2

Northwind中の運送会社テーブルや社員テーブルなどのデザインを表示させ、現れる型の種類とその型の値の例をあげよ。

例. テキスト型:アカネコ、トマト 数値型:101、203

関係データベースは、通常、共通の項目を持つ複数のテーブルから構成される。ツー ルメニューのリレーションシップウィンドウを開くと、共通の項目を持つ各テーブルの

項目一覧表とその関連が見られるので、確認してみよう。

11.1 サンプルデータベースの閲覧 <u>11.2 SQLによる問い合わせ</u> <u>11.3 SQL演習</u> <u>11.4 データベースの</u> 設計 ー橋大学 情報基礎

11. データベース

<u>11.1 サンプルデータベースの閲覧</u> 11.2 SQLによる問い合わせ <u>11.3 SQL演習</u> <u>11.4 データベースの設計</u> 11.2 SQLによる問い合わせ

1. テーブルの基本操作

データベースに対する問い合わせに対し、DBMS(データベース管理システム)は、 保存しているテーブルを操作して、目的にかなったテーブルを作成する。テーブルの 基本操作には、条件にあったレコード(テーブルの行)を取り出す「選択」、表示させ る項目(テーブルの列)を選ぶ「射影」、複数のテーブルを組み合わせる「直積」、が ある。

DBMSに問い合わせを行う際には、表に対する操作を直接記述するのではなく、 SQL (Structured Query Language)のSELECT文を用いる。SQLでは、SELECT文だ けでなく、新しいテーブルを作成(CREATE TABLE)したり、テーブルへのデータの追 加(INSERT INTO)、削除(DELETE)も行えるが、ここではSELECT文だけ学習する。

2. SELECT文の基本構文

SELECT文の基本構文

SELECT 項目リスト FROM テーブルリスト WHERE レコードの抽出条件;

と指定すると、FROMのテーブルリストにあるテーブルを組み合わせ(直積)て、抽出 条件をみたすレコードを選択し、SELECTで指定された項目だけを表示(射影)する。

POINT

注. 以下の実習では、全角文字と半角文字の区別に注意せよ。特に全角の空 白文字は空白(半角)とはみなされないので注意が必要である。

3. 例題

3.1:射影操作

社員テーブルの、氏名と部署名の項目のみ表示。 SELECT 氏名, 部署名 FROM 社員

クエリーを作成するには、「クエリー」タブをクリックして、「新規作成」→現れたウィンドウで【OK】→「テーブルの表示ウィンドウ」で【閉じる】。次に、メインウィンドウの左上の「ビュー」スピードボタンの▼を押してSQLビューに切り替え、現れたエディタ上で 直接入力(編集)せよ。

3.2:選択操作

社員テーブルから部署名="営業開発"であるレコードを選択し、全フィールド (*)を表示。 SELECT * FROM 社員 WHERE 部署名="営業開発" 3.3:直積操作

社員テーブルと受注テーブルの直積を作成。 SELECT * FROM 社員, 受注

直積では、各テーブルから取り出したレコードを並べて新たなレコードが作成される。しかし、あらゆる組み合わせについて新たなレコードが作成されるため、 通常無意味なレコードが大量に作成される。上の例では、社員Aのレコードと 別の社員Bの受注レコードを組み合わせたレコードは無意味であろう。したが って、

SELECT * FROM 社員, 受注

WHERE 社員、社員コード=受注、社員コード

のように、社員テーブルと受注テーブルで共通する「社員コード」項目が一致しているレコードのみを取り出す。このように、複数のテーブルの直積を取り、共通する項目の値が一致しているレコードのみ選択する操作を**結合**という。 なお、複数のテーブルで同じ名前の項目がある場合に、それらを区別するために、テーブル名と項目名をピリオド(..)でつないで社員、社員コード、受注、社員コードのように表す。

注意! 受注テーブルの「社員」フィールドは、実際には「社員コード」フィールドであり(デザインで確認せよ)、社員テーブルから氏名を引いて表示している (LOOK UPという)。したがって、受注テーブルの社員フィールドを参照したい 場合、受注.社員コード としなければならない。

<u>11.1 サンプルデータベースの閲覧</u> 11.2 SQLによる問い合わせ <u>11.3 SQL演習</u> <u>11.4 データベースの設計</u> ー橋大学 情報基礎

11. データベース

<u>11.1 サンプルデータベースの閲覧</u> <u>11.2 SQLによる問い合わせ</u> 11.3 SQL演習 <u>11.4 データベースの設計</u> 11.3 SQL演習

次の演習4~8までを行え。

実習4

例1のクエリーを入力して、問い合わせ行なえ。問い合わせをするには、 左上の「ビュー」スピードボタンの▼を押してデータシートビューに切り替え よ。

実習5. レコードの整列

(1)以下の例では、商品テーブルのレコードを商品名の昇順に並べ替えて、 商品名と単価の欄だけをまデオス

商品名と単価の欄だけを表示する。

SELECT 商品名, 単価 FROM 商品 ORDER BY 商品名 ASC;

このクエリーを作成し、問い合わせを行なえ。 (2)(1)を修正して、単価の降順(DESC)に整列するクエリを書け。 (3)さらに降順(昇順)に並べて異なる TOP 10 だけ表示させたければ、

SELECT DISTINCT TOP 10 商品名, 単価 FROM 商品 ORDER BY 単価 DESC;

のようにする。このクエリを作成し、問い合わせを行なえ。

実習6. テーブルの組み合わせ

2つ以上のテーブルを結合させてそこからデータを取り出すには、まず、 FROM句のテーブルリストに組み合わせたいテーブルを並べて書く。 (1)まず、テーブルの結合の仕組みをみるために、以下の<u>SQL</u>を実行せ よ。

SELECT * FROM 運送会社, 商品区分;

この例から分かるように、テーブルの結合では運送会社のレコード(3個)と 商品区分テーブルのレコード(8個)を組み合わせた24個のレコードが得ら れる。(ただしこの例は二つのテーブルに関連性がないのであまり意味が ない) (2)通常は、関連する(あるいは同じ)項目を持つテーブル同士で、その項 目が同じ値になるレコードを抽出する。例えば、商品テーブル中の商品に、 それが属する商品区分の情報(商品区分テーブルにある説明と図)を付加 するには、以下のようにすればよい。

SELECT 商品:*, 商品区分:説明, 商品区分.図 FROM 商品, 商品区分 WHERE 商品.区分コード=商品区分.区分コード;

このSQLを実行せよ。

実習7.集計関数の利用

集計関数(SUMやCOUNT)を利用して、集計結果を表示する項目をつくる こともできる。その際、GROUP BY を使って集計するレコードのグループ化 を行なうと、指定された項目の値ごとにグループにまとめられ、集計操作が 適用される。以下のSQLは「1997年売上高クエリー」で作成されるテーブ ルから、「商品区分名」ごとに「商品別売上高」の和(Sum)をとったものを、 「区分別売上高」として表示させるものである。

SELECT 区分名, Sum(商品別売上高) AS 区分別売上高 FROM [1997年 売上高] GROUP BY 区分名;

上のSQLを実行せよ。

一般にはこれらの例よりさらに複雑なSQLが可能で、以下のような形をしている。

SELECT [レコード選択方法の指定] 〈表示するフィールドや集 計関数〉 FROM 〈データを抽出するテーブル〉 [WHERE 〈レコードの抽出条件〉] [GROUP BY 〈グループ化に使われるフィールド〉] [HAVING 〈グループ化したレコードの表示条件〉] [ORDER BY 〈降順・昇順表示の指定〉] 注: [と]で括った部分は省略可能である。

ここでWHEREの条件では = 以外に <, >, <=, >=, <> も使える。

実習8 NorthWindデータベースにおいて、次の操作を行うクエリーを作成せよ。

- 1. 神奈川県にある仕入先の一覧表を求める(例1を参考にせよ)SQL。
- 2. 社員ごとにを受注件数を カウント(COUNT)した表を求める(実習7 を参考にせよ)SQL。
- 受注件数の多いTOP 3 の社員の表を求めるSQL。
 (ヒント: "2"で作成したクエリーが作る表のTOP 3 を取り出すクエリ を作れ。)

注意! "2"で受注テーブルの「社員」フィールドは、実際には「社員コー

ド」フィールドであり(デザインで確認せよ)、社員テーブルから氏名を引い て表示している。したがって、社員フィールドを参照したい場合、例えば SELECT 社員コード や GROUP BY 社員コード のようにしなければならな い。

課題

実習8で作成したSQL文と問い合わせの結果得られた表をWORD文書にまとめ、 Webclassから提出せよ。

<u>11.1 サンプルデータベースの閲覧</u> <u>11.2 SQLによる問い合わせ</u> 11.3 演習と課題 <u>11.4 データベースの設計</u> ー橋大学 情報基礎

11. データベース

<u>11.1 サンプルデータベースの閲覧</u> <u>11.2 SQLによる問い合わせ</u> <u>11.3 SQL演習</u> 11.4 データベースの設計 11.1 簡単なデータベースの設計

1.素朴なテーブル

蔵書リストを例に、データベースを設計するときに考慮すべき問題について考えて みよう。以下は、蔵書に関連する項目(フィールド)をとりあえず並べて作ったテーブ ルである。

| 蔵 | 書リストテーブル | | |
|---|-------------|------------|--------------------------|
| | フィールド名 | データ型 | 説明 |
| 鍵 | 書籍コード | テキスト型 | ISBN、分類番号 |
| | タイトル | テキスト型 | 本のタイトル |
| | 著者 | テキスト型 | 著者のリスト |
| | 購入日 | 日付/時刻 型 | 購入年月日 |
| | 購入価格 | 通貨型 | |
| | 備考 | メモ型 | 読後の感想、備考 |
| | 種類 | テキスト型 | 文庫、新書、B5ハードカバー、・・・ |
| | 出版社名 | テキスト型 | |
| | 出版社種別 | テキスト型 | 総合、絵本、歴史、文学、自然科 学、••• |
| | 出版社電話番 号 | テキスト型 | |
| | 出版社住所 | テキスト型 | |

テーブルは、そのフィールドの値からレコードが一意に定まるような、キー(鍵)フィ ールドを持たなければならない。上の例では、書籍コードをキーフィールドとしてい る。 ところで、この蔵書リストテーブルには欠陥がある。それは、出版社情報を同じテー

ところで、この蔵書リストナーフルには欠陥がある。それは、出版社情報を同じナー ブルに組み込んでいるので、

1. A 社の本を20冊持っていれば、A 社のデータが 20回重複する

2. 重複して入力されたデータの間に不整合が生じる可能性がある

といった不都合があるためである。

2. テーブルの分割

ー般的にいうと、テーブルのフィールド集合は、その中にサブキー(出版社名)とサブ キーから定まるフィールドの真部分集合(出版社名、出版社種別、出版社住所、出 版社電話番号)を含んでいてはならない。このようなフィールドの部分集合があれ ば、それらを別のテーブルとして独立させ、元のテーブルにはサブキーのみを残す べきである。

したがって、次のような書籍と出版社の2つのテーブルに分割すべきである。

| 書籍 | 唐テ ーブル | | |
|----|---------------|--------|--------------------|
| | フィールド名 | データ型 | 説明 |
| 鍵 | ISBN | テキスト型 | |
| | タイトル | テキスト型 | 本のタイトル |
| | 著者 | テキスト型 | 著者のリスト |
| | 購入日 | 日付/時刻型 | 購入年月日 |
| | 購入価格 | 通貨型 | |
| | 備考 | メモ型 | 読後の感想、備考 |
| | 種類 | テキスト型 | 文庫、新書、B5ハードカバー、・・・ |
| | 出版社名 | テキスト型 | |

| 出 | 版社テーブル | | |
|---|-------------|-----------|--------------------------|
| | フィールド名 | データ型 | 説明 |
| 鍵 | 出版社名 | テキスト 型 | |
| | 出版社種別 | テキスト 型 | 総合、絵本、歴史、文学、自然科 学、••• |
| | 出版社電話番 号 | テキスト 型 | |
| | 出版社住所 | テキスト 型 | |

3. 実習と課題

2で作成したテーブルでは、同じ書籍を重複して持っていなければよいが、場合によっては、同じ書籍を買うこともあるだろう。その場合、ISBNはもはや鍵フィールドならないので、例えばオートナンバー型の行番号を値とする鍵フィールドを導入する必要があり、さらに、書籍テーブルを分割する必要があるだろう。

実習

上の議論を参考に、蔵書リストあるいは自由なテーマでデータベースを 設計せよ。ただし、データベースは複数のテーブルから構成されるもので あること。 また、蔵書リストを選んだ場合、新たに購入書店の情報も付け加えるよう にせよ。

課題

実習で設計したデータベースに適当なデータ(レコード)を入力して作成したデ

ータベースファイルをWebClassから提出せよ。

<u>11.1 サンプルデータベースの閲覧</u> <u>11.2 SQLによる問い合わせ</u> <u>11.3 SQL演習</u> 11.4 データベースの設計 ー橋大学 情報基礎 12 ウィンドウアプリケーションの開発

――お絵かきソフトの自作――

簡単なウィンドウアプリケーションの作成を通して、アプリケーションがどのように 作成されるのかをみてゆこう。作成には、NetBeansというアプリケーション作成のた めの統合環境を用いる。NetBeansでは、実際にアプリケーションとして作成される ウィンドウ画面を見ながら、比較的容易にウィンドウアプリケーションを作成できる。

NetBeansのダウンロード

NetBeans はフリーで手に入る。、興味のあるものは最新バージョンを自宅 パソコンに<u>ここから</u>ダウンロードするとよい。大学のNetBeansとはバージョ ンが異なるが問題はないだろう。

本節で作成するアプリケーション

作成するアプリケーションは、まずボタンのクリックによってアプリケーションのウィンドウ(フォームという)の色を変えるものを作成し、それをさら に発展させて簡単なお絵かきソフトを作る。以下の手順にしたがって、ア プリケーションを作成すれば、驚くほど簡単にアプリケーションが作成でき ることがわかるだろう。

アプリケーションサンプル:うまく表示されないときはページを再表示(更新)せよ。



<u>12.1 NetBeans の起動</u>

<u>12.2 外観の設計</u>

<u>12.3 動作の記述</u>

<u>12.4 実習と課題</u>

ー橋大学 情報リテラシー

12. アプリケーション開発

12.1 NetBeansの起動 <u>12.2 外観の設計</u> <u>12.3 動作の記述</u> <u>12.4 実習と課題</u> **12.1 NetBeansの起動とアプリケーション開発の準備**

本節では、NetBeansを起動して、ウィンドウアプリケーション開発の準備を行う。

実習1. ウィンドウアプリケーションの作成準備
【スタート】→【プログラム】→【NetBeans5.5】を起動する
1. プロジェクトの作成

1)【ファイル】→【新規プロジェクト】で、

カテゴリ=一般、プロジェクト=Javaアプリケーションを指定して、【次へ】
2)プロジェクト名を英数字で適宜指定し、

プロジェクトの場所を授業用フォルダに指定し、

主クラスを作成のチェックを外して【完了】

2. ウィンドウズアプリの作成

1)【ファイル】→【新規ファイル】で新規ファイルウィザード

を開く 2)JavaGUIフォーム →JFrameフォーム を選択して【次へ】 3)クラス名を指定して【完了】

この結果、デザインエディタが現れ、コンポーネントパレットから部品を配置 したり、プロパティを設定して、アプリケーションの外観を設計することがで きる。

- なお、Webページ上で動作するアプリケーションを作成する場合は2.の2)で JavaGUIフォーム →JAppletフォーム
- を選び、ウィンドウを使わない文字ベースの入出力によるプログラムを作成する場合は

Javaクラス→Java主クラス を選択する。

12.1 NetBeansの起動 12.2 外観の設計 12.3 動作の記述 12.4 実習と課題

12. アプリケーション開発

<u>12.1 NetBeansの起動</u> 12.2 外観の設計 <u>12.3 動作の記述</u> <u>12.4 実習と課題</u> **12.2 外観の設計**

本節ではアプリケーションに必要な部品を、コンポーネントパレットから選んで、フォームに配置し、それらのプロパティを設定する。

1. 部品の配置

コンポーネントパレット上で目的の部品をクリックしてから、フォーム上の置きたい場所 をクリックする。

実習2:部品の配置

お絵描きアプリケーションを作成するために、以下の部品をフォーム上の適当な 位置に配置しよう。

- 1. JPanel (お絵描きのキャンバス用)
- 2. JButton (キャンバスの下塗りを行う)
- 3. JColorChooser (色の指定)

配置した部品には、jPanel1やjButton1のように、番号つきの名前がつく。また配 置した部品は、

| 移動 | コンポーネントをドラッグ |
|-----------|---|
| サイズ変 更 | コンポーネントをクリックして 口をド ラッグ |
| 削除 | コンポーネントをクリックしてDelet eキー |

が行えるので、大きさや場所などを調整するとよい。

2. プロパティの設定

フレーム(JFrame)およびフレーム上の配置した部品に対し、位置やサイズ以外にも 種々のプロパティを設定できる。

実習3:プロパティ設定

実習手順3:

- 1. jPanel1をクリックする(インスペクタ上で反転表示されることを確認せよ)。
- 2. プロパティタグのbackground欄をクリックする。
- 3. 現れた[…]ボタンをクリックし、エディタの中から適当な色を選ぶ。
- 4. jButton1をクリックする(インスペクタ上で反転表示される)。
- 5. プロパティタグの title欄をクリックし、右側の欄に「背景色」と入力する。

一般に、プロパティの設定方法には、

- 直接入力
- […]クリックで開くプロパティエディタで設定

の2通りがある。

3. アプリケーションの実行

作成したアプリケーションを実行するには、ソースエディタ上で ・マウスの右ボタンクリック→ファイルの実行 を選ぶか ・シフトキーとF&キーを同時におす(Shift+F6) ただし、この時点では、まだ何も動作を記述していないので、(ウィンドウ右上のの 最小化、最大化、閉じるボタンを除いて)何も反応しない。 ソースファイルは、実行と同時に自動的に、プロジェクトフォルダー中のsrcフォル ダに、保存される。実際に保存されていることを確認してみよ。

12.1 NetBeansの起動 12.2 外観の設計 12.3 動作の記述 12.4 実習と課題

12. アプリケーション開発

<u>12.1 NetBeansの起動</u> <u>12.2 外観の設計</u> 12.3 動作の記述 <u>12.4 実習と課題</u> **12.3 動作の記述**

前節で、アプリケーションの外観を整えることはできたが、まだ何の動作もしない。本節では、ウィ アプリケーションの動作を記述する方法を学ぶ。ウィンドウアプリケーションは、通常ユーザからの タンのクリック、マウス操作、キー入力等)を待機していて、これらの操作に応答するかたちで動作 のようなユーザ操作を含む種々の出来事をイベントと言い、イベントに対して応答メソッドを関連付 によって、動作を実現する。

実習2:イベントメソッドの記述1

以下、【下塗り】ボタンのクリックで、パネルの色をカラーチューザーで選択している色に変更 するようにする。

- 1. jButton1をクリックする(プロパティウィンドウのタイトルを確認せよ)。
- 2. イベントタグをクリックし、actionPerformed(イベント最上段)の右の空欄をクリックした ち、[Enter]キーを押す。
- 現れたソースエディタで、jButton1ActionPerformedメソッドの中身(下図の赤字下線部 だけを入力する。
 記述した命令の意味は、「jColochooser1のColorプロパティの値(色)を取得(get)し、j Panel1のbackgroundプロパティの値(色)に設定(set)する」ということである。
 少し見にくいが、最後の;(セミコロン)に注意せよ。

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
 jPanel1.setBackground(jColorChooser1.getColor());
}

ここでは、メソッドの外枠をNetBeansに用意させ、中身だけを入力する点が重要である。自分でこのメソッド全体を入力しても、(jButton1のイベントと連携されないので)正しく動かない

実習3:イベントメソッドの記述2

以下、マウスのドラッグ操作でフレーム(パネル)上に絵を描く機能を追加しよう。そのためには、 のウィンドウアプリケーションにおける描画の仕組みを理解する必要がある。

ウィンドウアプリケーションでは、ウィンドウの移動、最小化、最大化などに応じて、ウィンドウの 再描画しなければならないが、その際、フレームや部品等を描画するメソッドの他に、paintメソット ばれる約束になっており、必要ならばそこに描画命令を記述する。

1. 以下のpaintメソッドでは、位置(x,y)に半径rの円を描く。これをソース中の適当な位置(例え メソッドの手前)に記入せよ。

int x=0, y=0, r=5;//位置(x,y)と半径r

public void paint(java.awt.Graphics g) {

g.fillOval(x-r, y-r, 2*r, 2*r);//位置(x,y)に横径=縦径=rの(楕) 円を塗りつぶす

}

- 2. JFrameをクリックする(プロパティウィンドウのタイトルを確認せよ)。
- 3. イベントタグをクリックし、mouseDraggedイベントの右の空欄をクリックしたのち、[Enter]キー す。
- 4. 現れたソースエディタで、formMouseDraggedメソッドの中身(下図の赤字下線部)を入力す

private void formMouseDragged(java.awt.event.MouseEvent evt) {

```
<mark>x=evt.getX(); y=evt.getY();</mark> //(x,y)にマウス座標を代入
repaint(); //paint()で再描画
```

記述した命令の意味は、マウス位置のX座標を取得(get)し整数(int)型変数xに代入し、Y座標: (get)し整数(int)型変数yに代入し、そのうえで、再描画(repaint)を行う。これによって、Frame上 スをドラッグするたびにpaintメソッドが呼ばれ、(x,y)に代入されたマウス位置に円が描画される。 繰り返しになるが、メソッドをNetBeansに用意させ、中身だけを入力する点が重要である。

12.1 NetBeansの起動 12.2 外観の設計 12.3 動作の記述 12.4 実習と課題

12. アプリケーション開発

<u>12.1 NetBeansの起動</u> <u>12.2 外観の設計</u> <u>12.3 動作の記述</u> 12.4 実習と課題 **12.4 実習と課題**

このように、NetBeansにおけるアプリケーション開発は、例えばプラモデルを組み立てるように、

- 1. 部品(コンポーネント)を配置し
- 2. 部品のプロパティを設定し、
- 3. イベントに関連付けた応答メソッドをプログラムする

ことによってなされる。このようなプログラム手法を、オブジェクト指向(上記の1,2)お よびイベントドリブン(上記3)プログラミングという。

1. 実習

実習1:保存プロジェクトとファイルを開く

NetBeans を起動し、前回保存したプロジェクトとファイルを開こう。このときま ずプロジェクトを開くことに注意せよ。

- 1. NetBeans を起動する。
- 2. ファイルメニュー→「プロジェクトを開く」
- 現れた「プロジェクトを開く」ウィンドウで、授業フォルダ中に作成したプロジェクトフォルダを指定
- 4. 【プロジェクトフォルダを開く】ボタンをクリック
- 5. 左上のプロジェクトウィンドウで、ソースパッケージ→デフォルトパッケージ を展開し、 現れたiavaファイルをダブルクリック

実習2:筆の色と太さの設定

1. 筆の色の設定

マウスで描く筆を、黒ではなく、カラーチューザーで設定した色になるようにしよう。それには、paintメソッド中で円を描く前に

g.setColor(jColorChooser1.getColor());

として、グラフィックgが使う色(Color)をjColorChooser1から取得した色に設定すればよい。

2. 筆の太さの設定

JScrollBar部品を使って筆の太さ(円の半径)を変えられるようにしよう。

- JForm上の適当な位置にJScrollBarを配置し、Orientationプロパティを HORIZONTALにする。
- 2. jScrollBar1のadjustmentValueChangedイベントの右側の空欄をクリックして 【Enter】キーを押し、現れたソースエディタで、次のように記述する。

private void jScrollBar1AdjustmentValueChanged(java.awt.event.AdjustmentEvent evt) {
 r=jScrollBar1.getValue();
}

3. 結果の確認 ファイルを実行して、意図通り動くか結果を確認せよ。

実習3:作成者の明示

JFormのTitleプロパティに作成者の学籍番号・氏名を表示せよ。

2. 課題

| 提出 | 出期 艮 | 次回授業開始時まで |
|-------|---------|--|
| | _ | 完成させたプログラム(javaファイル)をWebCLassから提出せよ。 (JFormのTitleに作成者の学籍番号・氏名を表示させることに注意。) |
| 内 | 容 | 興味をもったら、アプリケーションに機能を追加したり(例えばペンの形を 変えるなど)、Javaの参考書にある他のアプリケーションを作ってみよう。山 崎の計算機概論の授業ページでもいろいろなプログラムを紹介している。 |

12.1 NetBeansの起動 12.2 外観の設計 12.3 動作の記述 12.4 実習と課題

13 アルゴリズムの重要性と公開鍵暗号

この章のポイント

問題を解く手順(解法)をアルゴリズムといい、その手順をコンピュータ上で実現したものをプログラムという。ここでは、アルゴリズム(プログラム)の巧拙によって、計算時間やメモリー量等が大きく異なることを紹介する。したがって、問題を解決するための効率のよいアルゴリズムを開発することはコンピュータサイエンスの重要な課題である。

一方、ある種の問題に対して効率のよい解法が存在しないことを示すこともまた、 コンピュータサイエンスの重要な課題であり、例えば暗号システム(暗号系)はそれ を効率よく解く(解読する)アルゴリズムが存在しないことが決定的に重要でる。ここ では、インターネットで広く利用されている公開鍵暗号系について、その仕組みを解 説する。

この章の構成

<u>13.1 アルゴリズムの巧拙</u>

<u>13.2 公開鍵暗号系</u>

13. アルゴリズムと公開鍵暗号

131 アルゴリズムの巧拙 <u>13.2 公開鍵暗号</u> 13.1 アルゴリズムの巧拙

本節では、アルゴリズムの巧拙によって大幅に計算時間が異なる例として、公開鍵暗号系で重要な役割を果たす(nを法とする)ベキ乗計算と、幅広い応用を持つ整列とをとりあげる。

1. 巾乗計算

×のd乗(x^d)の素朴な計算法は、×を順に掛けていく方法であろう。この方法では、d-1回の掛け算が必要だから、その計算にはdに比例する時間がかかる。これはdの桁数が小さいうちはいいが、dの桁数が増えてくると、現実的には計算できないといえるほどの時間がかかることになる。では、どのくらいの時間がかかるか、MOD電卓で試してみよう。ベキ乗の値はすぐに大きくなってしまうので、ここでは、MOD欄に入力された値を法とする(割った余りを求める)演算で計算している。実は、桁数の多いdに対する x^dのmod計算は次節の公開鍵暗号系における暗号化および復号で必要となる。

【巾乗】ボタンではx^dを、xをd-1回掛けて計算しているので、dの桁数が増えると計算時間が桁数の指数関数的に増える。このこと、つまり、桁数が1増えると計算時間が10倍になることを確認しよう。

ここで、仮に d=10,000,000 (=10⁷) 乗に1秒かかるとしよう。もちろん、実行するコンピュータによって多少異なるが、以下の議論に本質的な影響はない。1日は 3,600 × 24=86,400秒なので、1.16日≒10⁵秒と見積もれば、d=10¹⁶乗の計算には10⁹≒1.16 × 10⁴日=1,1600日かかる。つまり、dが16桁程度の大きさのとき、素朴な方法で x^d の計算をすることは現実問題として不可能である。

ー方、x^dの計算には、次のような高速計算法が知られている。高速巾乗計算では、x^dの計算に、dの2進表示と、 x¹, x², x⁴, x⁸, … の値を利用する。例えば15の2進表示は1111なので、x¹⁵= x⁸⁺⁴⁺²⁺¹= x⁸•x⁴•x²•x¹である。このとき、 x², x⁴, x⁸ の計算は2乗計算の繰り返しで掛け算3回、さらに、x⁸•x⁴•x²•x¹の計算が掛け算3回でできる。x²²ならば、22 は2進5桁の10110なので、x²²= x¹⁶⁺⁴⁺²= x¹⁶•x⁴•x²である。このとき、x², x⁴, x⁸, x¹⁶の計算は2乗計算の繰り返しで掛 け算4回、さらに、x¹⁶•x⁴•x²の計算が掛け算2回でできる。

この方法では、一般に x^d の計算に、dが2進表示でk桁のとき、x¹, x², x⁴, … x^{2^{k-1}}の系列を計算するのに掛け算k-1 回、これらを組み合わせてx^d を計算するのに(最大)k-1回の掛け算、都合、2k-2回の掛け算で計算できる。2進表示 の桁数は10表示の桁数のほぼ3.3倍(2¹⁰≒10³より)なので、高速巾乗は10進の桁数 log d に比例する時間(掛け算 回数)で計算できる。MOD電卓の高速ベキ乗ボタンでその計算時間を確認してみよう。

実習1. 以下の2種類の巾乗プログラムを完成させ、いくつかの d の値に対する計算時間の表を作り、WebClassから 提出せよ。

特に【巾乗】では、dの桁数が1増えると計算時間がほぼ10倍になることが分かるようにデータを採ること。

MOD電卓



mを法とする除算は、整数の除算と同様、常に可能だとは限らない。

2. 整列(ソーティング)

整列(ソーティング)は、データを大きさの順に並べ替えることで、表計算やデータベース管理システムにも基本機能 として実装されている。ここでは直観的に理解しやすい選択ソートと、高速整列法として知られるクイックソートについ て、基本的なアイディアを理解するとともにその計算時間を比べてみよう。

選択ソート

整列法比較欄で、【ソートアルゴリズム】のタブをクリックし、【生成】ボタンで乱数列を生成したら、【選択ソート】ボタン を押してみよう。 選択整列法では、

未整列部から最大値を選択して右端に移動させ、残りを未整列部とするということを繰り返す。

ここで選択ソートの計算時間を考えてみよう。未整列部のデータ数が k のとき、最大値を選択するのに k に比例する時間がかかるので、全体のデータ数が n のとき、n+(n-1)+(n-2)+・・・+2+1=n(n+1)/2 に比例する計算時間がかかることになる。これはデータ数が2倍になると計算時間がほぼ4倍になり、データ数が10倍になれば計算時間がほぼ100倍になることを意味する。このことを【時間比較】タブをクリックして確認してみよう。

クイックソート

整列法比較欄で、【ソートアルゴリズム】のタブをクリックし、【生成】ボタンで乱数列を生成したら、【クイックソート】ボタ ンを押してみよう。

クイックソートは、

未整列部の右端の値を基準に、その値以下の未整列部分と、その値、その値以上の未整列部に分割するということを繰り返す。

以下の計算例では、最初、右端の27を基準にして、27以下の未整列部と、27、27以上の未整列部に分割している。 次に27以下の部分を右端の9を基準に9以下の未整列部と、9、9以上の未整列部に分割し、27以上の部分を右端の 29を基準に29以下と、29、29以上に分割している。分割された部分が空(φ)か1データだけになれば整列済みでこれ 以上分割する必要がなく、データ数2以上の未整列部がなくなったところで、整列が終了する。

クイックソートの計算例

ソートアルゴリズム 時間比較 選択ソート 20 生成 クイックソート 29 34 36 4 18 25 39 1 12 3 29 9 30 32 1 19 30 23 17 27 <17 23 19 4 18 25 1 1 12 3 9 27 30 32 39 36 30 34 29 29> <3 1 1 4 9 25 19 23 12 17 18> 27 <29 29 39 36 30 34 30 32> <3 1 1 4 \$\phi > 9 < 17 12 18 19 25 23> 27 29 29 < 30 30 32 34 39 36> 1 3>49<φ 12 17>18<19 23 25> 27 29 29 < φ 30 30> 32 <34 <1 36 39>1 1 3 4 9 12 17 18 19 23 25 27 29 29 30 30 32 34 36 39 整列終了

クイックのソートの計算時間の評価は難しいが、未整列部の分割は未整列部のデータ数に比例する時間でできるので、全体としては分割にかかる時間の総和、すなわち、上の計算例で < と > で囲まれた部分の長さの総和に比例する時間でできる。その値(時間)は、最初のデータによって異なる(試してみよ)が、平均して n × log n に比例する時間であることが知られている。

log n は n の桁数に比例する値であるから、n と比較して非常に小さく、クイックソートの計算時間はデータ数が10倍になるとして、このことを、【時間比較】タブをクリックして確認するとともに、選択ソートと計算時間を比較してみよう。

実習. 2種類の整列法に対し、いくつかの n の値に対する計算時間の表を作り、WebClassから提出せよ。 特に、n が2倍、3倍にに増えたとき、それぞれの整列プログラムで計算時間がほぼ何倍になるかが分かるようにデー タを採ること。

整列法の比較

| ソートアルゴリズム | 時間比較 | | | | |
|-----------|------|-------|---------|----|--|
| | 生成 | 選択ソート | クイックソート | 20 | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

131 アルゴリズムの巧拙 13.2 公開鍵暗号

13. アルゴリズムと公開鍵暗号

<u>13.1 アルゴリズムの巧拙</u> 13.2 公開鍵暗号

13.2 公開鍵暗号の仕組み

インターネットは、オープンなネットワークであり、ネットワークを流れるデータには常に盗聴・改ざんの危険が伴う。したがって、例えばクレジットカード番号のような大事な情報のやりとりは、暗号化して送受信しなければならず、そのような暗号通信を保証するサイトして、https://で始まるサイトがある。

しかし、そのようなサイトとそれにアクセスする(不特定多数の)パソコンとの間で、どうやって暗号化通信を行うことができるのだろうか。この章では、 その原理について学習する。

1. 基本的枠組み

最初に、いくつかの言葉を定義しておこう。暗号化通信において、送信者は平文を暗号化して暗号文を作成し、それを受信者に送信する。受信者は 受信した暗号文を復号して、平文に戻す。

送信者:平文=【暗号化】⇒暗号 ~ 送信 ~ 受信者:暗号=【復号】⇒平文

2. 共通鍵暗号

従来の暗号システムでは、暗号通信を行う当事者二人が、共通の鍵を(他人に知られないように)秘密裏に共有し、暗号化および復号にその共通鍵 を用いる。共通鍵暗号システムでは、共通鍵を他人に知られることが情報の盗聴・改ざんに直結するため、共通鍵をいかにして秘密裡に共有しかつそ の秘密をいかにして守るかが問題であり、最初に述べたようなオープンなネットワークであるインターネットでは、そのための特別な仕組みが必要にな る。

また、共通鍵暗号システムでは、n人が互いに暗号通信を行う際には、送受信者のすべての組みについて異なる共通鍵を用意する必要があるため、 多数の共通鍵が必要になる。

問1.n人が互いに暗号通信を行うのに必要な共通鍵の個数を求めよ。

3. 公開鍵暗号

公開鍵暗号システムは、暗号化に用いる鍵と復号に用いる鍵が異なる暗号システムであり、このような鍵の対の

1) 一方の鍵を知られて(公開して)も、他方の鍵の秘密が保てる

2) どちらの鍵で暗号を作成しても、もう一方の鍵で復号できる

という特徴を持つ。このような鍵の対をどのように作るかは次節で説明するが、ここでは、その利用方法を考えよう。

公開鍵暗号システムでは、自分が使う鍵の対を生成すると、その一方を自分の公開鍵として公開し、他方を自分の秘密鍵として秘匿する。したがって、通信する2者の間で秘密の情報(鍵)を共有する手続きは不要であり、この点がインターネット上での暗号通信に用いられる理由である。

3.1. 暗号通信

受信者AIに暗号文を送信しようとする人は、Aの公開鍵で作った暗号文をAに送信する。それを受け取ったAは、自分の秘密鍵を使って復号する。Aの 公開鍵で作成された暗号文を復号できるのは、対応する秘密鍵をもっているAだけであり、通信の安全性が保たれる。

問2.n人が互いに公開鍵暗号通信を行うのに必要な鍵の対の個数を求めよ。

しかし、復号は、暗号文をもとの平文に戻す操作であるから、暗号化鍵=公開鍵、すなわち暗号化の方法を公開しても、その逆操作である復号鍵= 移密鍵の秘密が保てるのはどうしてであろうか。これについては、3節で議論する。

3.2. 電子署名

公開鍵暗号では、秘密鍵で暗号化した暗号文を公開鍵で復号できるという上記の特徴2)を利用して、電子的な文書に署名を付けることができる。署 名は文書の作成者が本人であることを保証し、文書の内容に責任をもつ働きがある。これを実現するため、文書の作成者Aは、文書をAの秘密鍵で 暗号化したものをAの署名として文書に付ける。

Aの署名を検証しようとするものは、Aの公開鍵でAの署名を復号し、文書と比較する。このとき

- Aの公開鍵で復号できる署名(暗号)を作成できるのは対応する秘密鍵を持つAだけであるから、作成者が確認できる
- 署名や文書の改ざんがあれば、署名を復号したものと文書とが一致しないので、改ざんの有無が検出できる

3.3.インターネットでの暗号通信

インターネットにおける暗号通信では、サイト(サーバー)にアクセスしたパソコン(クライアント)は、まずサーバーの公開鍵を取得する。次にクライアントは、以後の暗号通信のみに用いる共通鍵を作成し、それを公開鍵で暗号化して、サーバーに送信する。これによって、サーバーとクライアントの間で、安全に共通鍵を共有でき、以後、この共通鍵を使って、暗号通信が行われる。一般に、公開鍵より共通鍵の方が高速に暗号化と復号ができるので、このような方法がとられている。

ところで、最初に送られてきた公開鍵およびそのサイトは信用してもいいのだろうか。これを保証するのが認証局で、一般的に認められた認証局の認 証を受けていないサイトの(公開鍵の)場合、その旨の表示が出て注意が促される。

4. RSA暗号の作り方

公開鍵暗号の一つであるRSA暗号の基礎をなすのは、自然数mを法とする演算である。mを法とする演算(加算、減算、乗算)では、0以上m-1以下の整数同士の演算(加算、減算、乗算)をmで割った余りが演算結果になる。

- 特に以下の2点を確かめてみよう。
- ・MOD値が素数pのとき、x^{p-1}≣ 1 (mod p) (フェルマーの小定理)
- ・MOD値が素数 p, q の積 p×q のとき、x^{1+(p-1)(q-1)}≣ x (mod pa)

MOD電卓

| 0 MOD 2 | MOD値を法とする電卓 ・計算結果は上の欄に ・x+y=Min{z z×y=x} - MOD値が素数のとき、xのp-1乗=1 |
|-------------------|---|
| + - × + 巾乗 クリア | MOD値が素数p.qの積pqのとき、 xの1+(p-1)(q-1)乗=x ・ 巾乗計算時間は巾乗数に比例 ・ 高速巾乗計算時間は巾乗数の桁数に比例 |
| 高速巾乗 0 | |

問.以下の値を求めよ。

1) $4 \times 2 \mod 5$ 2) $3 \div 2 \mod 5$ 3) $5 \times 2 \mod 6$ 4) $4 \div 2 \mod 6$ 5) $3 \div 2 \mod 6$ 6) $3^6 \mod 7$ 7) 様々な x に対する x¹⁰ mod 11 8) 様々な x に対する x⁶¹ mod 77 9) 様々な x に対する x¹²¹ mod 77

RSA暗号では、素数 p, q の積 n=p × q を法とする演算で x ≡ x^{1+(p-1)}(mod n) が成り立つことを利用する。このとき、x ≡ x^{1+(p-1)}(q-1) ≡ x¹⁺²(p-1)(q-1)) ≡ x¹⁺³(p-1)(q-1) ≡ ··· (mod n) なので、d×e ≡ 1 (mod (p-1)(q-1)) ならば、(x^d)^e ≡ (x^e)^d ≡ x^{de} ≡ x (mod n) である。このとき、n**を法としたd乗とe乗が互** いに逆関数の関係 にあり、一方を暗号化に、他方を復号に利用できる。

以下のアプレットでRSA暗号を作成してみよう。まず、2つの素数 p, q と適当な数(素数がよい) e を決めておく(本来素数 p, q は非常に大きくしなければ安全でないが、ここでは実験なので、n=p × qが 65536 以上で、それほど大きな値にならないようにすること)。鍵作成ボタンを押し、d×e = 1 (mod (p-1)(q-1)) なる d が求まれば、n と e を公開鍵として公開し、p, q, d は秘密にする。d の計算は (p-1)(q-1)を法とした除算 (1÷e)なので、d が求まらない場合もあり、その場合は別の e を試す。

公開鍵を使って暗号文を送信する場合は、右の欄に平文を入れて【公開鍵暗号化】ボタンを押す。このとき、一文字ごとに、符号(の値)を(nを法と して) e 乗した値が、暗号として左の欄に表示(送信)される。 送信された暗号文を戻すには【秘密鍵復号化】ボタンを押す。このとき、送信された値は(nを法として) d 乗され、もとの文字に戻される。

送信された暗号文を戻すには**【秘密鍵復号化】**ボタンを押す。このとき、送信された値は(nを法として) d 乗され、もとの文字に戻される。 秘密鍵を使って署名する場合は、左の欄に平文(空白以外の文字列)を入れて**【秘密鍵署名】**ボタンをおす。すると、秘密鍵で作成した暗号文に平 文をつけて右の欄に送信される。

【公開鍵検証】ボタンでは、受け取った暗号文を公開鍵で復号化した結果と、受け取った平文とを比較して一致していればOKと表示する。OKの場 合、署名した人は、公開鍵で正しく復号化できる暗号文を作成できる秘密鍵を持っている人、すなわち公開鍵を作成した本人以外にはありえない。 暗号文や平文の一部を変更して、送信途中で送信エラーや改ざんがあったときには、NO!と表示されることを確認せよ。

| 復号 署名付送信 暗号化送信 署名検証 313 317 331 337 347 349 353 359 3 397 401 409 419 421 431 433 439 4 467 479 487 491 499 503 509 521 3 569 571 577 587 593 599 601 607 643 647 653 659 661 673 677 683 733 739 743 751 757 761 769 773 3 823 827 829 839 853 857 859 863 3 | 8p 素数q 秘密鍵d 331 457 0 | 鍵作成 クリア | 65536 公開鍵e 31 | 参考:3桁の素数 101 103 107 109 11 167 173 179 181 19 239 241 251 257 26 | 13 127 131 137 139 31 193 197 199 211 33 269 271 277 281 |
|---|--------------------------|--------------------------------------|---------------|--|---|
| | 復号 署名付送 | \$信 暗号化送信 | 署名検証 | 313 317 331 337 34 397 401 409 419 42 467 479 487 491 49 569 571 577 587 59 643 647 653 659 66 733 739 743 751 75 823 827 829 839 85 | 1 439 353 359 364 21 431 433 439 443 39 503 509 521 523 30 509 601 607 613 31 673 677 683 691 57 761 769 773 787 53 857 859 863 877 |

最後に、RSA暗号の安全性に考えてみよう。RSA暗号を破るには、素朴には、次のような方法が考えられる。

1) 公開鍵を使って作成した暗号(の値)を、nを法として1乗、2乗、3乗、…して元に戻るか否かを調べる。当然、d 乗まで調べた時点で元に戻るから、 秘密鍵 d をいつかは知ることができる。これは**秘密鍵dを虱潰しに探す**ことに対応し、d に比例する時間(d の桁数の指数時間) がかかる。これは言い 換えると、y ≡ x⁹ (mod n) (e, n は公開されているので y は誰でも計算できる)対し、x, y から x ≡ y^d (mod n) を満たす d を求めることに相当する。

2) nを素因数分解して、p, q を求め、(p-1)(q-1)を法とする(1÷e)を計算してdを求める。素因数分解を求める素朴な方法は、小さい方の素数から順に 虱潰しに割ってみる方法で、これも、p あるいは q (までの素数の個数)に比例する時間 がかかる。

1)の方法でRSA暗号を破るのは、アルゴリズムの節で述べたように、例えば d が10進16桁程度の大きさのとき、現実的でない。1,1600日後に、暗号 文の内容がわかったとしてもほとんど無意味だろう。また1,1600日では不安(1,1600台のパソコンを並列に動かせば1日でできるかもしれない)なのであ れば、dの桁数を増やせばよい。(実際上の安全性のためには、nの値を10進で300~1000桁程度にすることが推奨されている。)

同様に、2)の方法も、素数 p, qを十分大きくとれば、nの素因数分解を現実的な時間内に行うことはできない。

ところで、x^dの計算は復号計算であるから、その計算に x を d 回掛ける素朴な方法しかなければ、復号にとんでもない時間がかかってしまうことに なり、RSA暗号は機能しない。幸いdの値がわかっていれば、アルゴリズムの節で述べたように d 乗を高速に行う方法が知られている。

すなわち、**復号は、秘密鍵 d を知っている本人は高速実行できるが、秘密鍵を知らない第3者は、素朴な方法では現実的な時間内に実行できない**。 現在のところRSA暗号を現実的な時間内に破る方法は知られていない。1)について言えば、与えられた x, y, n, に対し、x ≡ y^d (mod n)を解く効率 的な方法は知られていない。しかし、何か巧妙な方法が、まだ見つかっていないだけで、今後見つかる可能性は今のところのではない。そのような方法 は存在せずそれゆえRSA暗号は安全であると広く信じられてはいるが、そのことの数学的に厳密な証明は得られておらず、P≠NP問題というクレイ数 学研究所から100万ドルの賞金がかけられている有名な未解決問題の一つに関わる問題である。

コラム. インターネット上でのコイントス

遠く離れた二人A、Bが、インターネット(通信回線でもよい)を介して、信頼できる第三者なしに、コイントスを行うことを考えよう。Aが コインを投げ、それを隠して、Bにコインの裏表を当てさせ、当たればBの勝ち、外れればAの勝ちという「賭け」である。 A、Bが共に信頼する公正な第三者Cがいれば、Aがコインを投げた結果(裏表)をCに預け、Bはその予想をCに伝えればよい。CはA から伝えられたコインの裏表とBからの予想を、その結果とともに発表すれば、A、B共に納得できるだろう。 しかし、AはCとBが結託して自分の結果(裏表)を知ってからBが予想立てているのではないかという疑念を、Cへの信頼なしに拭うこ
とはできないし、Bも、CがAと結託して、自分の予想を知ってからAがコインの裏表を変えているのではないかと疑念を、Cへの信頼なしに拭うことはできない。では、信頼できる第三者がいない状況で、コイントスが可能だろうか。実は、公開鍵を使えば、次の手順でこれが可能になる。

1. Aによるコイン投げ

A は、乱数 r を発生させて B に送り、r に対する A の署名 (r を A の秘密鍵で暗号化した値)s が偶数か否か尋ねる。

- 2. Bによる予想
- B は、s が偶数か否かを予想して答える。B は A の秘密鍵を知らないので、s の値をあらかじめ知ることはできない。
- 3. 結果の確認

<u>13.1 アルゴリズムの巧拙</u> 13.2 公開鍵暗号

一橋大学 情報基礎