

第9回 配列(array)型の変数

(添字付きの変数)

LiterateX

今日の内容

1. 配列の必要性
2. 配列の宣言
3. 配列変数のイメージ
4. 配列変数を使用した例
5. 多次元配列変数
6. 多次元配列変数を使用した例
7. データのソーティング
8. 今日の練習問題

LiterateX

1. 配列の必要性

多数のデータ処理

- (要求) 10,000人分の成績データもらって、合計点の大きい順に並べ替えて出力したい。
- (方法) ひとまず、全部のデータを変数に入れておく。
- (問題) しかし、`var d1, d2, d3, ..., d10000: integer;`のように、10,000個も変数を書けるか？
- (解決策) 配列を使おう!!
`var d: array[1..10000] of integer;`
 と書くだけで`d[1]~d[10000]`まで、10,000個の変数を使うことができる。

LiterateX

2. 配列の宣言

- 一般の変数を宣言する場所で配列も宣言する。

■ 宣言の一般形

配列名: `array[下限..上限] of 型;`

- 配列名は変数名と同様、アルファベット・数字・記号からなる。
 - 最初の文字はアルファベットであること。
 - 記号は一部しか使用できない。
- 下限と上限で添字の範囲を指定する。
 - 下限は1以外の数でもよい。
例えば、-5..100 のように書いてもよい。
 - 下限と上限は列挙型(整数、文字など)の値でなければならない。
- 型は構成要素の型を表す。

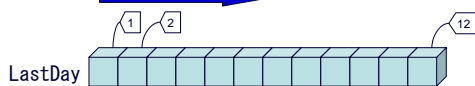
LiterateX

3. 配列変数のイメージ

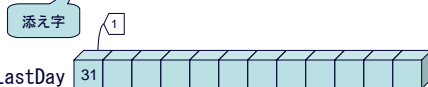
`LastDay : array[1..12] of integer;`

12個の箱が並んでいると考える。

先頭から順番に1、2、3、...12と添え字が付く。



`LastDay[1] := 31;` ← 代入文



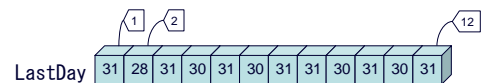
LiterateX

配列変数へのデータの代入と参照

```

program LastDayOfMonth:
{$APPTYPE CONSOLE}
uses SysUtils;
var LastDay : array[1..12] of integer;
    month : integer;
begin
    LastDay[1]:=31; LastDay[2]:=28; LastDay[3]:=31;
    LastDay[4]:=30; LastDay[5]:=31; LastDay[6]:=30;
    LastDay[7]:=31; LastDay[8]:=31; LastDay[9]:=30;
    LastDay[10]:=31; LastDay[11]:=30; LastDay[12]:=31;

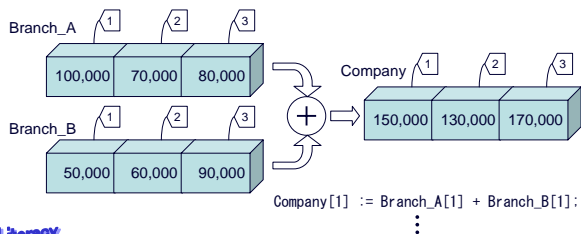
    write('月の日数を答えます。何月を知りたい?:');
    readln(month);
    writeln(month, '月は', LastDay[month], '日です。');
    readln
end.
    
```



LiterateX

4. 配列を使用した例

- ある会社に2つの支店 (A支店とB支店) があるとする。
- 各支店で1月～3月の各月の売上額が下の表のようなデータとしてある場合、この会社の各月の売上額を計算する。



プログラム例

```

program sales;
{$apptype console}
uses SysUtils;
var Branch_A, Branch_B : array[1..3] of integer;
    Company : array[1..3] of integer;
    month : integer;
begin
    Branch_A[1]:=100000; Branch_A[2]:=70000;
    Branch_A[3]:=80000;
    Branch_B[1]:=50000; Branch_B[2]:=60000;
    Branch_B[3]:=90000;
    for month := 1 to 3 do [各月で計算]
        Company[month] := Branch_A[month] + Branch_B[month];
    writeln('この会社の売上額');
    for month := 1 to 3 do [各月で売上額を表示]
        writeln(month, '月は', Company[month], '円');
    readln
end.
    
```

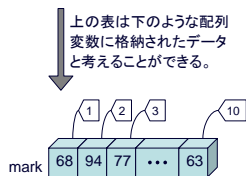
配列を使用した例(偏差値)

- テストの成績データがあったとして各学生の偏差値を計算するプログラムを作成してみよう。

学生番号	1	2	3	4	5	6	7	8	9	10
成績	68	94	77	52	73	85	79	71	86	63

出力例

学生番号	成績	偏差値
1	68	??
2	94	??
3	77	??
⋮	⋮	⋮
10	63	??



偏差値の計算

- 偏差値は下に示す式で求まる。

$$\text{学生}i\text{の偏差値} = \frac{(\text{学生}i\text{の得点} - \text{平均}) \times 10}{\text{標準偏差}} + 50$$

- 平均と標準偏差を求める。

$$\text{平均} = \frac{\text{個々の得点の総和}}{\text{学生数}} = \frac{d_1 + d_2 + d_3 + \dots + d_{10}}{10} = \frac{1}{10} \sum_{i=1}^{10} d_i$$

$$\text{標準偏差} = \sqrt{\frac{(\text{個々の得点} - \text{平均})^2 \text{の総和}}{\text{学生数}}} = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (d_i - m)^2}$$

- 総和を求める計算

- 繰り返し文を使ってプログラム
 - 右のプログラム片で計算できる。
- ループ終了後、変数waの値が総和

プログラム片

```

wa := 0;
for i := 1 to 10 do
    wa := wa + d[i];
    
```

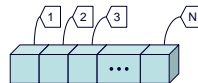
プログラム例

```

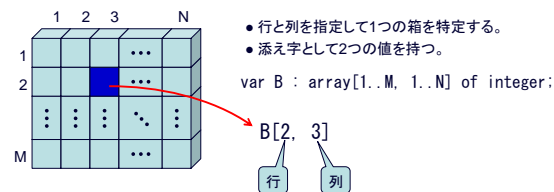
program Hensachi;
{$APPTYPE CONSOLE}
uses SysUtils;
const N = 10;
var wa, y : integer;
    m, sigma, wa2 : real;
    i : integer;
begin
    wa := 0;
    for i := 1 to N do wa := wa + mark[i]; [総和を求める]
    m := wa / N;
    wa2 := 0;
    for i := 1 to N do wa2 := wa2 + sqr(mark[i] - m);
    sigma := sqrt(wa2 / N);
    writeln('学生番号 成績 偏差値');
    for i := 1 to N do begin [各学生の偏差値を計算し、表示する]
        y := round(10 * (mark[i] - m) / sigma + 50);
        writeln(i:5, ' ', mark[i]:3, ' ', y:3);
    end;
end;
readln
end.
    
```

5. 多次元配列変数

- 1次元配列: 一列(直線的)に並べた変数



- 2次元配列: 平面的に並べた変数



多次元配列変数の数学との対応

意味を与えた添字付きの変数を使う計算

例 $c_{ij} := a_{ij} + b_{ij}$ (行列の和の計算)

Pascal	数学
1次元配列 a[i]	数列 a_i
2次元配列 b[i, j]	行列 b_{ij}
3次元配列 c[i, j, k]	???? c_{ijk}
⋮	

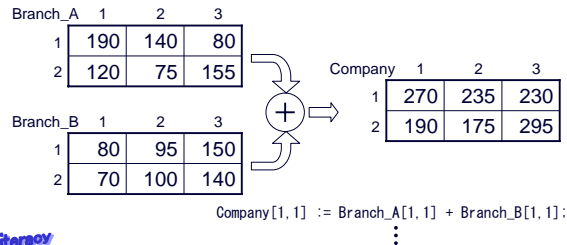
Pascalの配列では、何次元配列でも作ることができる。

例) var b : array[1..n, 1..m] of integer;
配列bは、n×m 行列



6. 多次元配列を使用した例

- ある会社に2つの支店(A支店とB支店)があるとする。
- 各支店で1月～3月の各月の商品1、商品2の売上数が下の表のようなデータとしてある場合、この会社の各月の各商品の売上数を計算する。



```

program Sales2;
{$APPTYPE CONSOLE}
uses SysUtils;
var Branch_A, Branch_B : array[1..2, 1..3] of integer;
    Company : array[1..2, 1..3] of integer;
    good, month : integer;
begin
    Branch_A[1, 1]:=190; Branch_A[1, 2]:=140; Branch_A[1, 3]:= 80;
    Branch_A[2, 1]:=120; Branch_A[2, 2]:= 75; Branch_A[2, 3]:=155;
    Branch_B[1, 1]:= 80; Branch_B[1, 2]:= 95; Branch_B[1, 3]:=150;
    Branch_B[2, 1]:= 70; Branch_B[2, 2]:=100; Branch_B[2, 3]:=140;
    for good := 1 to 2 do [各商品で処理]
        for month := 1 to 3 do [各月で処理]
            Company[good, month] :=
                Branch_A[good, month] + Branch_B[good, month];
    writeln('この会社の売上数');
    for good := 1 to 2 do begin [商品ごとに1行で表示]
        write('商品 ', good, ':');
        for month := 1 to 3 do
            write(' ', Company[good, month]:4, '個');
        writeln
    end;
    readln
end.
    
```



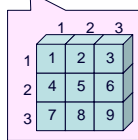
3行3列行列の和の計算

■ 数学の行列計算

- 3行3列の行列の和を求める問題
- 下のような計算

- 行列の i 行 j 列の要素の和を計算する

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} + \begin{pmatrix} 9 & 6 & 3 \\ 8 & 5 & 2 \\ 7 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 10 & 8 & 6 \\ 12 & 10 & 8 \\ 14 & 12 & 10 \end{pmatrix}$$



3行3列行列の和の計算

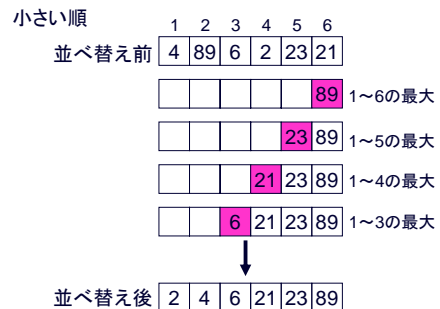
```

program mat0;
{$APPTYPE CONSOLE}
uses SysUtils;
var a, b, c : array[1..3, 1..3] of real;
    i, j : integer;
begin
    writeln('行列A=?'); [行列Aの成分を入力]
    for i := 1 to 3 do
        for j := 1 to 3 do read(a[i, j]);
    writeln('行列B=?'); [行列Bの成分を入力]
    for i := 1 to 3 do
        for j := 1 to 3 do read(b[i, j]);
    for i := 1 to 3 do [和C=A+Bを計算]
        for j := 1 to 3 do c[i, j]:=a[i, j]+b[i, j];
    writeln('A + B ='); [和C=A+Bの成分を出力]
    for i := 1 to 3 do begin
        for j := 1 to 3 do write(c[i, j]);
        writeln
    end;
    readln
end.
    
```



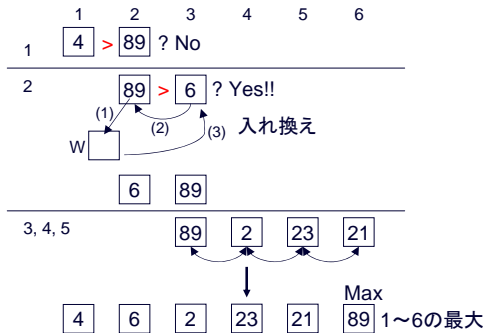
7. データのソーティング

並べ替えイメージ(その1)



並べ替えイメージ(その2)

隣り合った数を比べ、前の数が大きければ入れ替える。



プログラム例

```

program bubble_sort;
($APPTYPE CONSOLE)
uses SysUtils;
const N = 6; // 定数Nを宣言
const d: array[1..N] of integer = (4, 89, 6, 2, 23, 21);
var i, j, w: integer;
sd: array[1..N] of integer;
begin
    [並べ替えるために、データを配列sdにコピーする。]
    for i := 1 to N do sd[i] := d[i];
    for i := 1 to N do
        for j := 1 to N - i do
            if sd[j] > sd[j+1] then begin
                w := sd[j];
                sd[j] := sd[j+1];
                sd[j+1] := w;
            end;
        end;
    writeln('Sorted data:');
    for i := 1 to N do write('sd[' , i, ':', ' ' '); writeln;
    for i := 1 to N do write(sd[i], ' '); writeln;
    readln;
end.
    
```

8. 今日の練習問題

Level	問題
C	多次元配列の例題の会社において、3ヶ月間の全売上数を求めるプログラムを作成せよ。
C	月と日を入力すると、その年の1月1日からの経過日数を表示するプログラムを作成せよ(閏年は考えない)。
B	配列内の変化の様子を観察できるようにソートの例題プログラムを変更せよ。
B+	n個のデータもらって、各データの値だけ「*」を横に並べる棒グラフを描くプログラムを作成せよ。 (参考: テキストp.57の例5.4.2)
A	上の棒グラフを縦棒のグラフにせよ。
A	3×3行列の固有値を求めるプログラムを作成せよ。