

構造体の配列

身体検査データを、構造体の配列として実現したのが、List 2-13 のプログラムです。

chap02/physical.c

List 2-13

// 身体検査データ用構造体の配列

```
#include <stdio.h>
```

```
#define VMAX 21 // 視力の最大値2.1×10
```

```
/*--- 身体検査データ型 ---*/
```

```
typedef struct {  
    char name[20]; // 氏名  
    int height; // 身長  
    double vision; // 視力  
} PhysCheck;
```

```
/*--- 身長の平均値を求める ---*/
```

```
double ave_height(const PhysCheck dat[], int n)  
{  
    double sum = 0;  
    for (int i = 0; i < n; i++)  
        sum += dat[i].height;  
    return sum / n;  
}
```

```
/*--- 視力の分布を求める ---*/
```

```
void dist_vision(const PhysCheck dat[], int n, int dist[])  
{  
    for (int i = 0; i < VMAX; i++)  
        dist[i] = 0;  
    for (int i = 0; i < n; i++)  
        if (dat[i].vision >= 0.0 && dat[i].vision <= VMAX / 10.0)  
            dist[(int)(dat[i].vision * 10)]++;  
}
```

```
int main(void)
```

```
{  
    PhysCheck x[] = {  
        {"AKASAKA Tadao", 162, 0.3},  
        {"KATOH Tomiaki", 173, 0.7},  
        {"SAITOH Syouji", 175, 2.0},  
        {"TAKEDA Shinya", 171, 1.5},  
        {"NAGAHAMA Masaki", 168, 0.4},  
        {"HAMADA Tetsuaki", 174, 1.2},  
        {"MATSUTOMI Akio", 169, 0.8},  
    };
```

```
    int nx = sizeof(x) / sizeof(x[0]);
```

// 人数

```
    int vdist[VMAX];
```

// 視力の分布

```
    puts("■□■ 身体検査一覧表 ■□■");  
    puts(" 氏名          身長 視力 ");
```

```
    puts("-----");
```

```
    for (int i = 0; i < nx; i++)
```

```
        printf("%-18s%4d%5.1f\n", x[i].name, x[i].height, x[i].vision);
```

```
    printf("\n平均身長: %5.1fcm\n", ave_height(x, nx));
```

```
    dist_vision(x, nx, vdist);
```

// 視力の分布を求める

```
    printf("\n視力の分布\n");
```

```
    for (int i = 0; i < VMAX; i++)
```

```
        printf("%3.1f ~: %2d人\n", i / 10.0, vdist[i]);
```

```
    return 0;
```

実行結果

■□■ 身体検査一覧表 ■□■
氏名 身長 視力

AKASAKA Tadao	162	0.3
KATOH Tomiaki	173	0.7
SAITOH Syouji	175	2.0
TAKEDA Shinya	171	1.5
NAGAHAMA Masaki	168	0.4
HAMADA Tetsuaki	174	1.2
MATSUTOMI Akio	169	0.8

平均身長: 170.3cm

視力の分布

0.0 ~: 0人

0.1 ~: 0人

0.2 ~: 0人

0.3 ~: 1人

0.4 ~: 1人

... 以下省略 ...

このプログラムは、身体検査データの一覧表を表示し、さらに、平均身長と視力の分布を表示します。

本プログラムの冒頭で宣言・定義している **PhysCheck** は、氏名（文字列）、身長（**int** 型）、視力（**double** 型）をまとめた構造体です。

▶ 本プログラムでは、構造体の宣言と **typedef** 宣言をまとめて行っています（すなわち、構造体の宣言をするとともに **typedef** 名を与えています。なお、タグ名は与えていません）。

身体検査データを格納するのが、**PhysCheck** 型の配列 **x** です。各要素に対して、氏名・身長・視力のデータが初期化子として与えられています。

前節で学習した配列は、要素型や構成要素型が基本型でした（基本型とは、文字型・整数型・浮動小数点型の総称です）。配列の要素や構成要素は、基本型だけでなく、列挙型や構造体なども許されることが分かります。

▶ 配列 **x** の要素数は、式 $\text{sizeof}(x) / \text{sizeof}(x[0])$ で求めています。この求め方は、p.44 で学習しました。

プログラムでは、二つの関数が定義されています。

関数 **ave_height** は、身体検査データの配列を受け取って、身長の平均値を実数値で求める関数です。

関数 **dist_vision** は、視力の分布を求める関数です。分布の格納先は、第3引数 **dist** です。視力の分布は 0.1 刻みで求めます。

▶ 本プログラムは、視力の最大値が 2.1 であるという前提で作られています。

演習 2-11

List 2-13 のプログラムの視力の分布の表示を、右のようなグラフで出力するように書きかえたプログラムを作成せよ（注：右に示すグラフは一例であって、List 2-13 のプログラムの視力の分布とは一致しない）。記号文字 '*' を人数分だけ繰り返し表示すること。

```
0.1 ~ : *
0.2 ~ : ***
0.3 ~ : *
... 以下省略 ...
```

演習 2-12

日付を表す構造体が右のように与えられているとして、次の関数を作成せよ。

- **y** 年 **m** 月 **d** 日を表す構造体を返却する関数 **DateOf**
`Date DateOf(int y, int m, int d);`
- 日付 **x** の **n** 日後の日付を返す関数 **After**
`Date After(Date x, int n);`
- 日付 **x** の **n** 日前の日付を返す関数 **Before**
`Date Before(Date x, int n);`

この他にも、いろいろな関数を設計して作成すること。

```
typedef struct {
    int y;    // 西暦年
    int m;    // 月 (1~12)
    int d;    // 日 (1~31)
} Date;
```