中国人民大学
RENMIN UNIVERSITY OF CHINA

C程序设计

第6讲 结构体

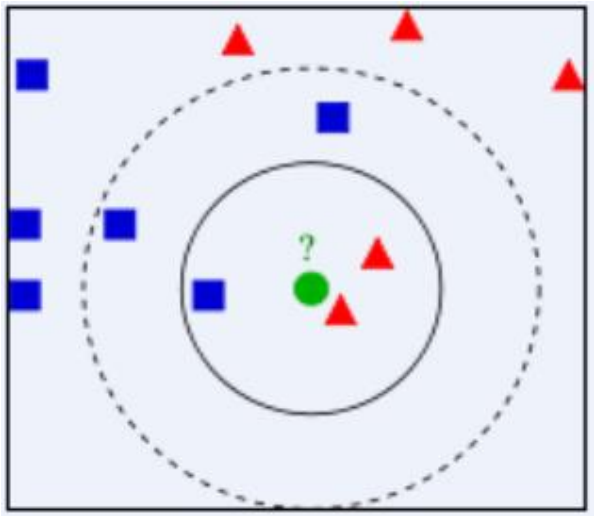
余力

buaayuli@ruc.edu.cn

期中题型分析

- **序列最小差**（数值计算、基本送分）
- **三解形**（枚举、排序、思路算法）
- **社交距离**（数值计算、分类求和）
- **年级姓名排序**（统计排序、字符串）
- **K近邻法**（统计排序、统计输出）

从 #273 K近邻算法 讲起



【输入样例】	【输出样例】
3 4	1
3 5 4 5 3 4 5	
3 1 1 10 1 0 10	
3 5 6 6 5 8 9	
5 5	

$$D_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$$

1）如果待分类样本到两个点的距离 D 相同，那么它跟类别编号小的那个样本更相近。

2）如果最相似第 k+1、k+2、……、k+j 样本到待分类样本的距离等于最相似的第 k 个样本到待分类样本的距离，那么确定待分类样本的类别是需要考虑这 j 个样本。

3）与待分类样本最相似的 k 个样本中无法找到一个唯一的大多数所属的类别，则认为相同的大多数类别中类别编号小的那个类别为待分类样本所属的类别。例如 k 取 4 时，如果发现 k 个最相似的样本有 2 个来自类别 1，另外 2 个来自类别 2，则将待分类样本划分为类别 1。

4）样本总数小于 k 时，考虑所有样本。

```
int n, m, k, P[10000][4] = {0}, x0, y0, count = 0,
int temp, num, ClassNum[100] = {0}, ClassID = 0;
scanf("%d%d", &m, &k);
for (int i = 0; i < m; i++) {
    scanf("%d", &n);
    for (int j = 0; j < n; j++) {
        scanf("%d%d", &P[count][1], &P[count][2]);
        P[count][0] = i;
        count++;    }
}
```

```
scanf("%d%d", &x0, &y0);
for (int i = 0; i < count; i++)
    P[i][3] = (P[i][1] - x0) * (P[i][1] - x0) + (P[i][2] - y0) * (P[i][2] - y0);
```

```
if (k>=count)  num = count;
else    for (num=k; num<count; num++)
            if (P[num][3] > P[k-1][3]) break;
for (int i = 0; i < num; i++)
    ClassNum[P[i][0]]++; //统计每个节点所属类别
for (int i=0, MaxNum=0; i<m; i++)
    if (ClassNum[i] > MaxNum) {
        MaxNum = ClassNum[i];
        ClassID = i + 1;    }
printf("%d", ClassID);
```

传统风格

```
for (int i = 1; i < count; i++)
    for (int j = i - 1; j >= 0; j--)
        if ((P[j][3] > P[j + 1][3]) || (P[j][3] == P[j + 1][3] && P[j][0] > P[j + 1][0])) {
            temp = P[j][0]; P[j][0] = P[j + 1][0]; P[j + 1][0] = temp;
            temp = P[j][3]; P[j][3] = P[j + 1][3]; P[j + 1][3] = temp;    }
```

```
struct point P[10000], temp;
int n, m, k, x0, y0, count = 0, num, ClassNum[100] = {0}, MaxClassID = 0;
scanf("%d%d", &m, &k);
for (int i = 0; i < m; i++) {
    scanf("%d", &n);
    for (int j = 0; j < n; j++) {
        scanf("%d%d", &P[count].x, &P[count].y);
        P[count].classID = i; //存储类别
        count++;    }
}
```

结构体风格

```
scanf("%d%d", &x0, &y0);
for (int i = 0; i < count; i++)
    P[i].dis = (P[i].x - x0) * (P[i].x - x0) + (P[i].y - y0) * (P[i].y - y0);
```

```
for (int i = 1; i < count; i++)
    for (int j = i - 1; j >= 0; j--)
        if (P[j].dis > P[j+1].dis || (P[j].dis == P[j + 1].dis && P[j].classID > P[j+1].classID)) {
            temp = P[j];  P[j] = P[j + 1];  P[j + 1] = temp;    }
```

```
if (k >= count) num = count;
else for (num = k; num < count; num++)
        if (P[num].dis > P[k - 1].dis)  break;
for (int i = 0; i < num; i++)
    ClassNum[P[i].classID]++;
for (int i = 0, MaxNum = 0; i < m; i++)
    if (ClassNum[i] > MaxNum) {
        MaxNum = ClassNum[i];
        MaxClassID = i + 1;    }
printf("%d", MaxClassID);
```

```
struct point {
    int classID;
    int x;
    int y;
    int dis;  };
```

核心区别1 - 表述直观

- int P[10000][4];

  P[i][0] → P[i].classID

  P[i][1] → P[i].x

  P[i][2] → P[i].y

  P[i][3] → P[i].dis

```
struct point {
    int classID;
    int x;
    int y;
    int dis; };
```

- int clsssID[10000], x[10000], y[10000], dis[10000];

  clsssID[i] → P[i].classID

  x[i] → P[i].x

  y[i] → P[i].y

  dis[i] → P[i].dis

核心区别2 – 比较省事

```
for (int i = 1; i < count; i++)
    for (int j = i - 1; j >= 0; j--)
        if ((P[j][3] > P[j + 1][3]) || (P[j][3] == P[j + 1][3] && P[j][0] > P[j + 1][0])) {
            temp = P[j][0]; P[j][0] = P[j + 1][0]; P[j + 1][0] = temp;
            temp = P[j][3]; P[j][3] = P[j + 1][3]; P[j + 1][3] = temp;
        }
```

多方面都要交换

```
for (int i = 1; i < count; i++)
    for (int j = i - 1; j >= 0; j--)
        if (P[j].dis>P[j+1].dis || (P[j].dis==P[j + 1].dis && P[j].classID > P[j+1].classID)) {
            temp = P[j]; P[j] = P[j + 1]; P[j + 1] = temp;
        }
```

一次性整体交换

内容

6.1 为什么结构体

6.2 结构体定义

6.3 结构体数组

中国人民大学
RENMIN UNIVERSITY OF CHINA

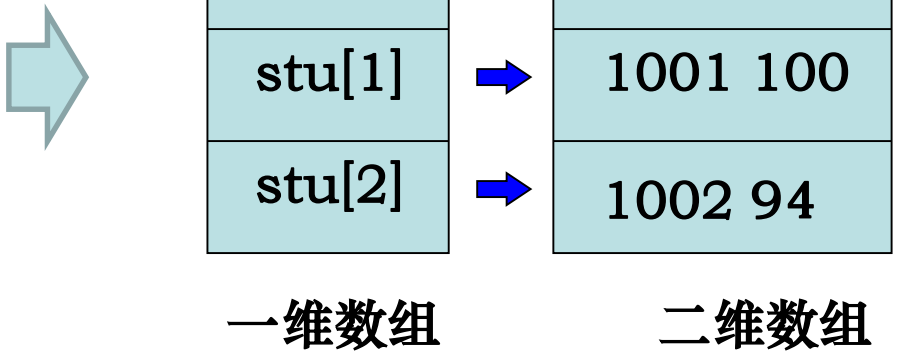
1. 为什么结构体

变量概念拓展：从数据 到 对象

核心问题

- 如何表示一个对象（学生）？
  - 表示为二维数组，例：stu[1000][2]
  - 每一行表示一名学生
  - 第一列表示学号
  - 第二列表示平均成绩

```
样例一
5
1000 95
1001 100
1002 94
1006 95
1007 100
```



为什么要使用结构体

- 考虑下面的实际问题
  - 将一组学生按照出生日期进行排序
- 待排序的“学生”不是基本数据类型！
- 学生信息包括
  - 姓名：汉语拼音，最多20个字符
  - 性别：M / F
  - 生日：19841107（年月日）
  - 身高：1.74（m）
  - 体重：51.5（kg）
- 使用现有学的知识完成此任务？
  - 定义5个数组，在排序算法中同时操作这5个数组

#308 猪场分配

```
for (i = 0, Farm_Count = 0; i < Farm_Total; i++) {
    for (j = 0; j < 5; j++)
        scanf("%d", &data[i][j]);
    if (data[i][1] == -1) continue;
    ID[Farm_Count] = data[i][0];
    rongliang[Farm_Count] = data[i][2];
    base_cost[Farm_Count] = data[i][3];
    each_cost[Farm_Count] = data[i][4];
    Farm_Count++;//记录可用场子数
}
```

```
int ID[3001];
int rongliang[3000];
int base_cost[300];
int each_cost[300];
```

```
struct pig {
    int ID;
    int rongliang;
    int base_cost;
    int each_cost;
} PigFarm[3000];
```

中国人民大学
RENMIN UNIVERSITY OF CHINA

2. 结构体定义与使用

结构体类型的定义

- 结构体：表示一组类型可能不同的数据

```
struct Student {
    char  name[20];          //姓名
    char sex;                //性别
    unsigned  long birthday; //生日
    float height;            //身高
    float weight;            //体重
};
```

```
struct farm {
            int num;
            int state;
            int max_cap;
            int basic_cost;
            int pig_cost;
};
```

```
struct Stu {
            int id;
            int kemu;
            double aver;
};
```

定义结构体类型的格式

```
struct 结构体名
{
    类型名 1   成员名 1；
    类型名 2   成员名 2；
      . . .
    类型名 n   成员名 n；
};
```

struct 是结构体类型的标志，结构体名student是编程者自己选定

编程习惯：首字母大写！

大括号所括起来的5条语句是结构体中5个成员的定义。

结构体定义之后一定要跟一个“；”

结构体变量的定义与引用 (1)

```
#include <stdio.h>

struct  Student                    //定义结构
{
    char  name[20];                //姓名
    char sex;                      //性别
    unsigned  long birthday;    //生日
    float height;                  //身高
    float weight;                  //体重
 };
```

结构体变量的定义与引用 (2)

```
int main()      {                                   // 主函数
  struct Student my;  // 结构体变量printf(  “输入自己的数据
    \n”  ); // 提示信息
    printf( "姓名（汉语拼音）\n" );  // 显示提示
    printf( "性别：M/F\n" );
    printf( "生日（年月日）\n" );
    printf( "身高（米）\n" );
    printf( "体重（kg）\n\n" );
```

结构体变量的定义与引用 (3)

```
// 依次输入个人信息
  scanf( "%s", my.name);
  scanf( "%c", &my.sex );
  scanf( "%d", &birthday );
  scanf( "%f", &my.height );
  scanf( "%f", &my.weight );
```

结构体变量的定义与引用 (4)

```
    // 依次输出个人信息
    printf( "%s\n", my.name );
    printf( "%c\n", my.sex );
    printf( "%d\n", my.birthday );
    printf( "%f\n", my.height );
    printf( "%f\n", my.weight );
    return 0;
}
```

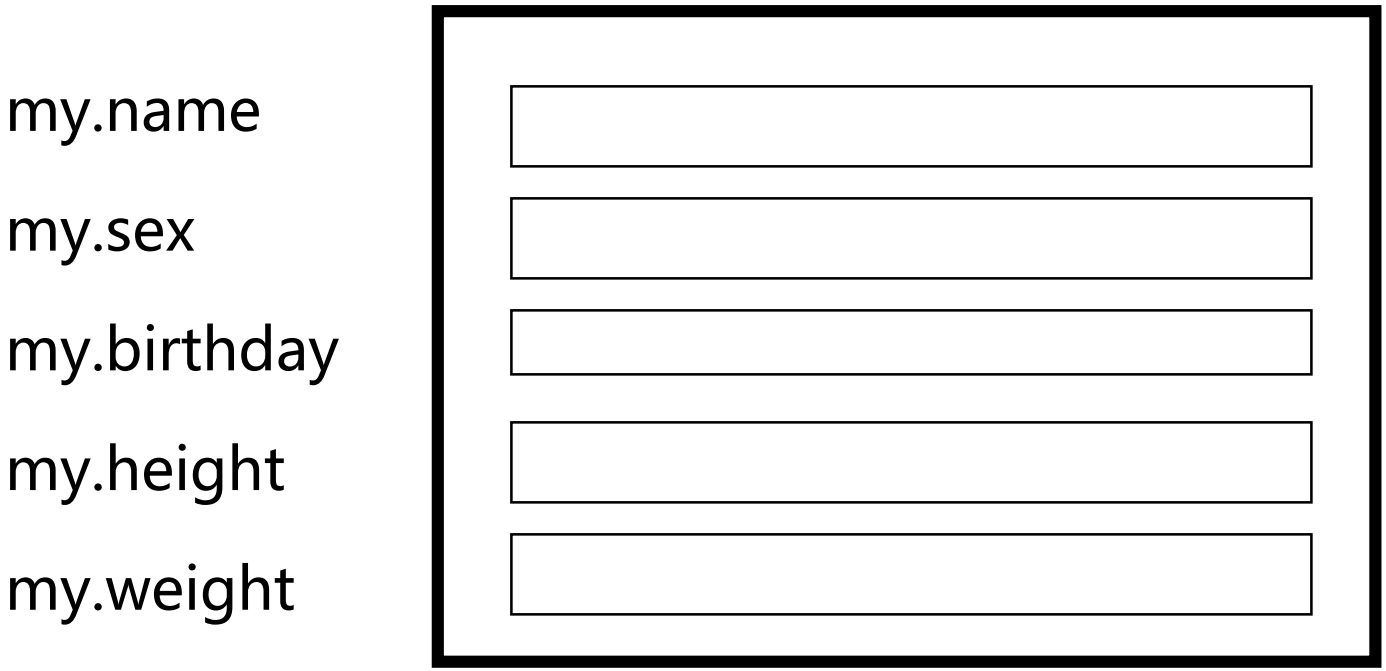
说明

- 定义类型不会分配内存空间

```
struct Student    //此处为结构类型定义
 {
     char  name[20];                  //姓名
     char sex;                        //性别
     unsigned  long birthday;         //生日
     float height;                    //身高
     float weight;                    //体重
 };
```

struct Student my;

变量 my （my为变量的符号地址）



& my（变量的内存地址）

结构体变量的初始化

- 方法一：定义和初始化同时完成

```
struct  Person
{
    char  name[10];
    unsigned  long  birthday;
    char  placeofbirth[20];
} per = { “Li ming”, 19821209, “Beijing”};
```

Person 是结构体类型
per 是结构体变量

```
per.name = "Liming";
per.birthday = 19821209;
per.placeofbirth = "Beijing";
```

结构体变量的初始化

- 方法二：分开完成

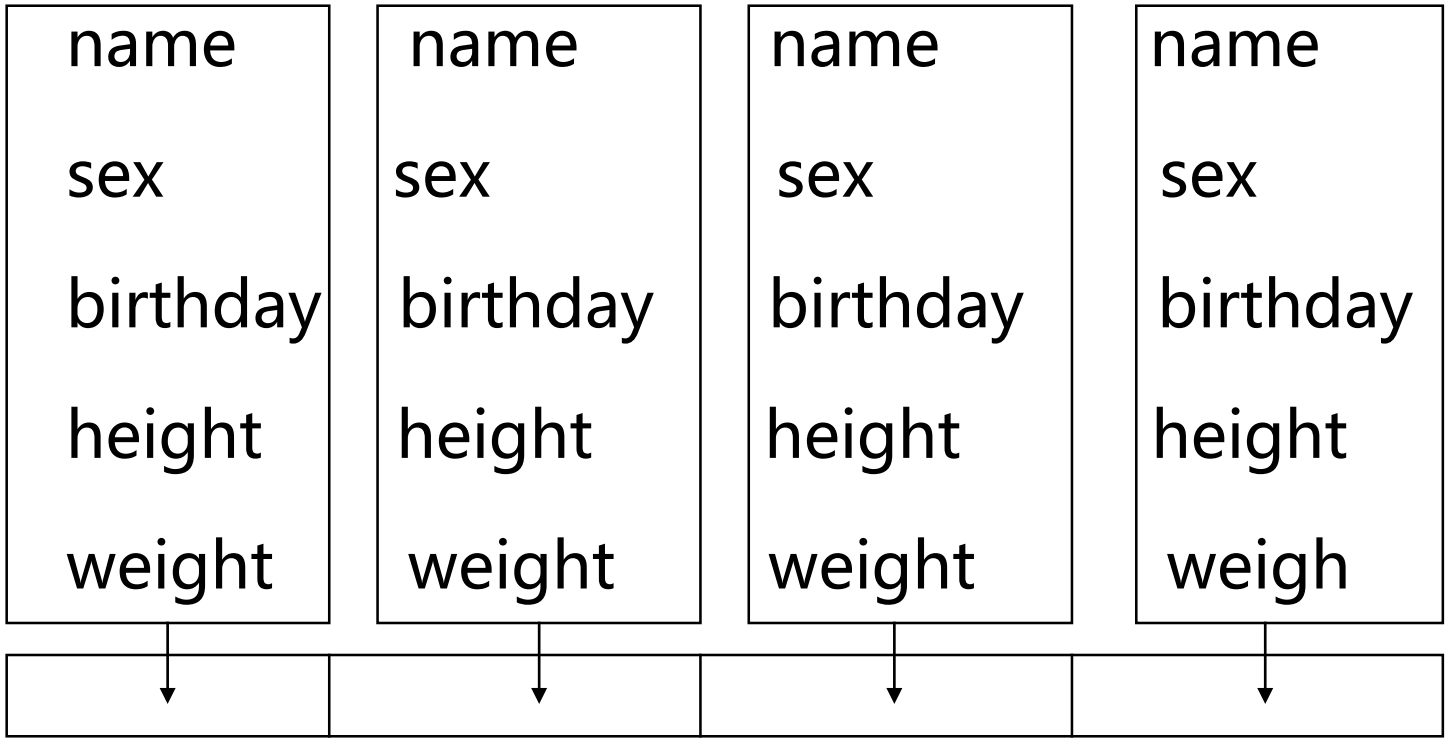
```
struct  Person
{
    char  name[10];
    unsigned  long  birthday;
    char  piaceofbirth[20];
 };

struct Person per = { “Li ming” , 19821209,  “Beijing” };
```

3. 结构体数组

结构数组

- 结构也可以构成数组，即每个数组元素是一个结构；
- 当然，要求这一类数组的全部元素都应该是同一类结构；
- 例如：**同宿舍4名同学的数据，构成一个有4个**元素的结构数组。

```
struct Student  STUs[ 4 ] = {
        { “Li li”,  ‘M’, 19840318, 1.82, 65.0 },
        { “Mi mi”,  ‘M’, 19830918, 1.75, 58.0 },
        { “He lei”,  ‘M’, 19841209, 1.83, 67.1 },
        { “Ge li”,  ‘M’, 19840101, 1.70, 59.0 }
};
```


name
sex
birthday
height
weight
name
sex
birthday
height
weight
name
sex
birthday
height
weight
name
sex
birthday
height
weigh
下标0
下标1
下标2
下标3
STUs

STUs[0]	“Li li”, ‘M’, 19840318, 1.82, 65.0
STUs[1]	“Mi mi”, ‘M’, 19830918, 1.75, 58.0
STUs[2]	“He lei”, ‘M’, 19841209, 1.83, 67.1
STUs[3]	“Ge li”, ‘M’, 19840101, 1.70, 59.0

```
#include <stdio.h>
struct  Student  //名为student的结构类型
{
    char  name[20];              //姓名
    char sex;                    //性别
    unsigned  long birthday;   //生日
    float height;                //身高
    float weight;                //体重
 };
struct Student  STUs[4] = {//定义全局结构数组/初始化
      {“Li li”, ‘M’, 19840318, 1.82, 65.0 },
      {“Mi mi”, ‘M’, 19830918, 1.75, 58.0 },
      {“He lei”, ‘M’, 19841209, 1.83, 67.1 },
      {“Ge li”, ‘M’, 19840101, 1.70, 59.0 }
};
```

```
int  main() {
   struct Student q;      //结构变量q，用于交换时保存中间结果
   int i = 0 ; int j =0;

   for ( j = 0; j < 4-1; j++ )
      for ( i = 0; i < 4-1– j ; i++ )
          if ( STUs[ i ].birthday >  STUs[i+1].birthday )
                {    q= STUs[ i ];     //交换，结构变量赋值
                     STUs[ i ]=STUs[ i+1];
                     STUs[ i+1]= q;     }

   for( i = 0; i < 4; i = i + 1 ) {    //输出排序后的结果
          printf( "%s\n", STUs[i].name );
          printf( "%c\n", STUs[i].sex );
          printf( "%d\n", STUs[i].birthday );
          printf( "%f\n", STUs[i].height );
          printf( "%f\n", STUs[i].weight );        }

   return 0;
}
```

Chp06_学生生日排序.cpp

#195 平均成绩排序（结构体）

```
struct Stu
{
    int id;
    int kemu;
    double aver;
};

int main(){
    int n,i,j,k,sum,lin1,lin2,z=0,x,lo;
    struct Stu stu[10000];
    struct Stu tmp;
    scanf("%d",&n);
```

```
int main(){
    int n,i,j,k,sum,lin1,lin2,z=0,x,lo;
    struct Stu stu[10000];
    struct Stu tmp;
    scanf("%d",&n);
```

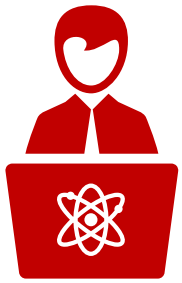
```
    // 输入 n 位学生信息
    for(i=0;i<n;i++)
        {sum=0;
        scanf("%d %d",&lin1,&lin2);
        if(lin2<2)
            for(int u=0;u<lin2;u++)
                scanf("%d",&lin1);
        else //输入学生的多门课成绩并求好平均数
            {stu[z].id=lin1; stu[z].kemu=lin2;
            for(j=0;j<stu[z].kemu;j++)
                {scanf("%d",&x);
                sum=sum+x;
                }
            stu[z].aver=sum*1.0/stu[z].kemu;
            z++; //重新计算保存有 2 门以上课程的学生
            }
    }
```

#195 平均成绩排序

```
    if(z==0){
        printf("NO");
        return 0;
    }
    else{

        for(i=0;i<z-1;i++){
            for(j=0;j<z-1;j++)
                if((fabs(stu[j].aver-stu[j+1].aver)<1e-7&&stu[j].id>stu[j+1].id)||stu[j].aver<stu[j+1].aver)
                    {tmp=stu[j];stu[j]=stu[j+1];stu[j+1]=tmp;}
            }
        for(i=0;i<z;i++){
            printf("%d %.2lf\n",stu[i].id,int(stu[i].aver*100)/100.0);
        }
    return 0;}
```

Chp06_平均成绩排序(#195)

中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！