11.7-结构与联合

代码示例1：（结构体）

#include<iostream>

using namespace std;

/\* 该程序可实现统计一家三口的姓名、年龄、身高、体重和月收入的功能

结构体就像数组一样，可以将多个元素存储在一个单元中，但它有远优于数组的特点，在该程序中得以体现。

\*/

struct statistics

{ //结构体的名称为statistics，这相当于用户自定义了一个新型“大变量的类型”——每个“大变量”里有5个不同或相同类型的普通变量

char name[10];

int age;

float hght; //height

double wght; //weight

double inc; //income

}; //注意结构体末尾有";",点在大括号以外

/\*

至此不难看出，相比于数组，结构体有一个很大的优势——数组虽然也能存储多个元素，但元素的类型必须相同，

比如无法用同一个数组同时存储int型和char型的两个变量，而结构体可以做到这一点

\*/

int main()

{

statistics Mom=

{ "Alice",

37,

1.63,

51.2,

5432.1

};

statistics Dad=

{ "Tommy", 39, 1.83, 80.0, 12000.0};

double a;

cin>>a;

statistics Child=

{ "Jerry", 13, 1.71, 75.6, a};

cout<<Dad.name<<" "<<Mom.hght<<" "<<Child.age<<endl; //数据访问

double total=Dad.inc+Mom.inc+Child.inc; //求总收入

cout<<total<<endl;

return 0;

}

代码示例2：（结构体嵌套）

#include<iostream>

using namespace std;

struct worktime

{

int h; //hours

int min; //minutes

}day[3];

struct person\_worktime

{

char name[10];

worktime t;

};

int main()

{ cout<<"请输入第一天工作的时长:"<<endl;

cin>>day[1].h>>day[1].min;

cout<<"请输入第二天工作的时长:"<<endl;

cin>>day[2].h>>day[2].min;

worktime total;

total.min=(day[1].min+day[2].min)%60;

total.h=day[1].h+day[2].h+(day[1].min+day[2].min)/60;

cout<<"Totally"<<" "<<total.h<<" hours"<<" "<<total.min<<" minutes"<<endl;

//直接给结构类型数组赋值的方法：

worktime a[2]= { {1,20},{2,35} };

cout<<a[0].h<<" "<<a[1].min<<endl;

person\_worktime p={"aoaoao",{2,35}}; //里面这个大括号也可以去掉

cout<<p.name<<"的工作时长为:"<<p.t.h<<"小时"<<p.t.min<<"分钟"<<endl;

return 0;

}

代码示例3：（联合体——共同体）

#include<iostream>

using namespace std;

union exp

{

int a;

int b;

};

int main()

{

union exp m;

m.a=5;

cout<<m.a<<" "<<&m.a<<endl;

cout<<m.b<<" "<<&m.b<<endl;

cin>>m.b;

cout<<m.a<<" "<<&m.a<<endl;

cout<<m.b<<" "<<&m.b<<endl;

//所以说，管union的叫共用体还真是贴切——完全就是共用一个内存首地址，并且各种变量名都可以同时使用，操作也是共同生效。

return 0;

}