

结构体

- 结构体和类是C++中两种自定义的数据类型，用来组织数据的行为。
- 结构体和类都允许创建用户定义的数据类型，使得代码更具模块化和可扩展性。
- 两者都支持对现实世界中的实体进行抽象建模，即将对象的属性（数据）和行为（方法）结合起来形成一个完整的实体。如将一个学生抽象成一种数据类型，学生有年龄，姓名，分数这些属性。

1.结构体

1.1结构体的定义

```
// struct 是关键字
struct 结构类型名
{
    成员表; //可以有多个成员
};
```

例如构建一个学生成绩有关的数据类型

```
struct Student{
    char *name; //姓名
    int age; //年龄
    int school_id; //学号
};
/*
```

注意：各个变量都不能在结构体内初始化，因为结构体声明只是创建一个新的数据类型，还不存在这种类型的变量实体。

```
*/
```

1.2先定义结构体再定义结构体变量

```
struct Person
{
    char name[11];
    int age;
    bool gender;
    char address[101];
    char id[19];
};
```

```
Person a[10], b;
```

1.3定义结构体的同时定义结构体变量

```
// struct 是关键字
struct 结构类型名
{
    成员表; //可以有多个成员
}结构体变量表;
```

例如

```

struct Person
{
    char name[11];
    int age;
    bool gender;
    char address[101];
    char id[19];
}Person a[10], b; // 定义a数组变量, b变量

```

1.4 结构体定义的注意事项

- 定义结构体变量时，结构体变量名和结构体类型名不能相同
- 定义结构体的时候，系统不分配实际内存
- 定义结构体变量的时候，系统才分配内存

1.5 结构体和数组

数组：存储一组相同类型数据项的变量

结构体：用户自定义的数据类型，允许存储不同类型的数据项

数组和结构体相结合，可以写出功能很强大的程序

1.6 结构体变量的输入输出以及赋值

```

#include <iostream>
#include <string>
using namespace std;

// 定义结构体
struct Person {
    string name;
    int chinese;
    int math;
    int total;
};

int main() {
    // 创建结构体变量
    Person person1;

    cin >> person1.name >> person1.chinese >> person1.math;
    person1.total = person1.chinese + person1.math;

    // 输出结构体变量的属性值
    cout << "\n您输入的信息是：" << endl;
    cout << "姓名：" << person1.name << endl;
    cout << "语文：" << person1.chinese << endl;
    cout << "数学：" << person1.math << endl;
    cout << "总分：" << person1.total << endl;
    return 0;
}

```

1.7 结构体数组变量的输入输出以及赋值

```
#include <iostream>
#include <string>
using namespace std;

// 定义结构体
struct Person {
    string name;
    int age;
    string occupation;
};

int main() {
    const int arraySize = 3; // 定义数组大小
    Person people[arraySize]; // 定义结构体数组

    // 输入结构体数组的属性值
    for (int i = 0; i < arraySize; ++i) {
        cout << "请输入第 " << i + 1 << " 个人的信息: " << endl;

        cout << "姓名: ";
        cin >> people[i].name;

        cout << "年龄: ";
        cin >> people[i].age;

        cout << "职业: ";
        cin >> people[i].occupation;

        cout << endl;
    }

    // 输出结构体数组的属性值
    cout << "您输入的信息是: " << endl;
    for (int i = 0; i < arraySize; ++i) {
        cout << "第 " << i + 1 << " 个人的信息: " << endl;
        cout << "姓名: " << people[i].name << endl;
        cout << "年龄: " << people[i].age << endl;
        cout << "职业: " << people[i].occupation << endl << endl;
    }

    // 对结构体数组的赋值操作
    Person person2 = {"Alice", 25, "Engineer"};
    people[1] = person2;

    // 输出更新后的信息
    cout << "更新后的信息: " << endl;
    for (int i = 0; i < arraySize; ++i) {
        cout << "第 " << i + 1 << " 个人的信息: " << endl;
        cout << "姓名: " << people[i].name << endl;
        cout << "年龄: " << people[i].age << endl;
        cout << "职业: " << people[i].occupation << endl << endl;
    }

    return 0;
}
```

1.8 结构体嵌套

```
#include <iostream>
#include <string>
using namespace std;

// 定义日期的结构体
struct Date {
    int day;
    int month;
    int year;
};

// 定义记录出生日期的结构体，包含了日期结构体作为成员
struct Person {
    string name;
    Date birthdate; // 使用日期结构体作为成员
};

int main() {
    // 创建Person结构体变量
    Person person1;

    // 赋值日期结构体的属性值
    person1.birthdate.day = 15;
    person1.birthdate.month = 8;
    person1.birthdate.year = 1990;

    // 赋值Person结构体的其他属性值
    person1.name = "小明";

    // 输出赋值后的信息
    cout << "姓名: " << person1.name << endl;
    cout << "出生日期: " << person1.birthdate.year << "年"
        << person1.birthdate.month << "月" << person1.birthdate.day << "日" <<
endl;

    return 0;
}
```

1.9 结构体变量的初始化, 直接赋值, 整体交换

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

// 定义结构体
struct Person {
    string name;
    int chinese;
    int math;
    int total;
} students[2]={
    {"xiaozi",100,80},
    {"xiaoming",89,77},
}; //结构体变量的初始化

int main() {
    // 创建结构体变量
    student_3 = {"xiaofei",88 , 89};

    students[0] = student_3; //结构体变量的直接赋值
    swap(students_3, students[1]); //结构体变量的整体交换
    return 0;
}
```

1.10 结构体变量的排序

排序库函数中, 是将数组

```

#include <iostream>
#include <string>
#include<algorithm>
using namespace std;

// 定义结构体
struct Student {
    string name;
    int chinese;
    int math
    int total;
};
Student a[100];
int main()
{
    int n;
    cin >> n;
    for(int i =0;i<n;i++)
    {
        cin >> a[n].name >> a[n].chinese >> a[n].math >> a[n].total;
    }
    sort(a, a+n, cmp);
    for(int i =0;i<n;i++)
    {
        cout << a[n].name << a[n].chinese << a[n].math << a[n].total << endl;
    }
    return 0;
}

bool cmp(Student s1, Student s2)
{
    return s1.total > s2.total;
}

```

<algorithm> 库中的 sort() 函数说明，在 C++ 的 sort() 函数中，可以提供一个自定义的比较函数（或者比较谓词），以便对容器中的元素进行排序。这个比较函数被称为 cmp 参数。

思考：如何修改上述的函数使得按照数学成绩的顺序进行排序呢？