一维数组

Int Array[20] //定义20长度的数组

Int Array[]={1,2,3,4} //定义并赋值

Array[1] =0 //赋值，数组范围是从0开始

Sum=array[1]\*2 //数组参与运算

Cin>>array[1] //输入数组元素

数组下标不能越界

Count <<array //数组不能整体赋值

Char a[]=”holle” //字符串，有\0

Char a[]={“h”,”o”,”l”,”l”,”e”} //字符串数组，没有\0

Cin>>array //一维数组输入

Cout<<Array //一维数组输出

二维数组

Int scor[1][2]

//静态定义

Const int n=1,m=3;

Double poin[n][m];

//行（横） 列（竖）

Int a[2][2]={1,2,3,4}

Int a[][2]={1,2,3,4}

Int a[2][2]={{1,2},{1,2}}

结构体

反映一个特质

定义

Struct <结构体名称>

{

成员1

成员2

};

声明结构体变量

1 struct <结构体名称> <变量名称>

Struct Data student;

1. 直接在定义的时候声明

Struct Data //这里可以省略data

{

Int a,b;

}student;

初始化赋值

Struct Data student={“a”,”b”}

Student.a=1; //已经声明结构体类型

结构体数组

Struct data student[30]

Student[1].a=”s”;

枚举

Enum <枚举类型名><枚举常量表>

Enum week[‘a’,’b’] //a=0,b=1

Enum week[a=1,b=100]

函数调用数组

Fun(int a[],int b);

Int a[]={1,2,3,4};

Fun(a,b)

二维数组形参列不能省略

Fun(int a[][10],int b)

结构体变量做参数

//声明

Struct student{

Int a;

}

//定义函数

Void fun(student a){

..

}

//min调用

Student b={1} //结构体赋值

Fun(b)//这里的b与结构体student的a对应

函数重载

一个函数名相同的函数，可以处理不同的数据类型的数据，如int double，参数个数不同

系统会根据不同点选择不同的函数

全局变量

在int main（）外定义

：：a在{}中使用全局变量，如果在局部中有同名的变量

Extern 全局变量声明

使用全局变量会降低通用性

变量存储内省

堆区 存放动态分配的数据

栈区 存放局部数据，局部变量

全局数据区 全局数据和静态数据，全局变量

程序代码区 程序的各个函数代码

变量存储类型

Auto自动类

register寄存器类 仅局部变量

static静态类

extern外部类