项目6 利用数组完善项目中数据的处理

目标 TARGET

知识目标

- 掌握一维数组的定义、初始化和引用方法
- 掌握二维数组的定义、初始化和引用方法熟悉Visual
- 理解字符数组与字符串的区别，掌握它们的使用方法

目标 TARGET

技能目标

- 能够利用一维数组知识解决批量数据问题，比如存储、排序、插入等
- 能够利用二维数组处理类似行列式的问题
- 能够使用字符数组处理字符串

目录

任务6.1多个学生一门课程成绩的总成绩和平均成绩

任务描述

在学生成绩管理系统中，已知6名同学《C语言程序设计》课程的期末成绩分别为85、80、90、93、78、69，请问这6名同学《C语言程序设计》课程的总成绩是多少分，平均成绩是多少分？请编制一个程序。

任务分析

传统方案为：
1）定义6个float变量。
2）统计求和，并求出平均值。
从算法分析中，我们可以看到传统的方案，当人数变多，定义的变量个数也随之变多，使用不灵活，累加的变量个数丢失时容易出现误差。为了满足大数量数据的计算需求，在C语言程序设计中提出了数组的概念。

Part 1

一维数组的定义及其应用

◎一维数组的定义

◎一维数组的引用

◎一维数组的初始化

◎一维数组的应用

6.1.1 一维数组的定义

数组：在程序设计中，一组具有相同数据类型的变量集合称为数组。

一维数组：是只有一个下标的数组，它用来表示一组具有相同类型的数据。

在C语言中，一维数组的定义方式如下所示：

类型说明符 数组名[常量表达式];

- 类型说明符表示数组中所有元素的数据类型；
- 数组名就是这个数组型变量的名称；
- 常量表达式指一维数组中元素的个数，即数组长度。

6.1.2 一维数组的引用

数组定义后，就可以引用数组中的任意一个元素了，引用形式如下：

<数组名>[<下标表达式>]；

其中，

“下标表达式”表示数组中的某一个元素的顺序序号，

序号范围是“0-[数组长度-1]”。

6.1.3 一维数组的初始化

完成数组的定义后，只是对数组中的元素开辟了一块内存空间。
这时，如果想使用数组操作数据，还需要对数组进行初始化。

一维数组初始化的一般格式如下：

<类型说明符> <数组名>[常量表达式] = {初值表};

一维数组初始化的常见的方式有三种，具体如下：

1）直接对数组中的所有元素赋值，例如：int i[6]={1,2,3,4,5,6};

2）只对数组中的一部分元素赋值，例如：int i[6]={1,2,3};

3）对数组全部元素赋值但不指定长度，例如：int i[]={1,2,3,4}。

6.1.4 一维数组的应用

例 6-1

案例：

在学生成绩管理系统中，已知6名同学《C语言程序设计》课程的期末成绩分别为85、80、90、93、78、69，请问这6名同学《C语言程序设计》课程的总成绩是多少分，平均成绩是多少分？请编制一个程序。

6.1.4 一维数组的应用

例 6-1

算法：

1.定义一维数组result[6]、求和变量sum、求平均值变量ave、变量i。

2.循环输入6名同学《C语言程序设计》的期末成绩。

3.循环输出6名同学《C语言程序设计》的期末成绩，观察成绩在数组中的存储。

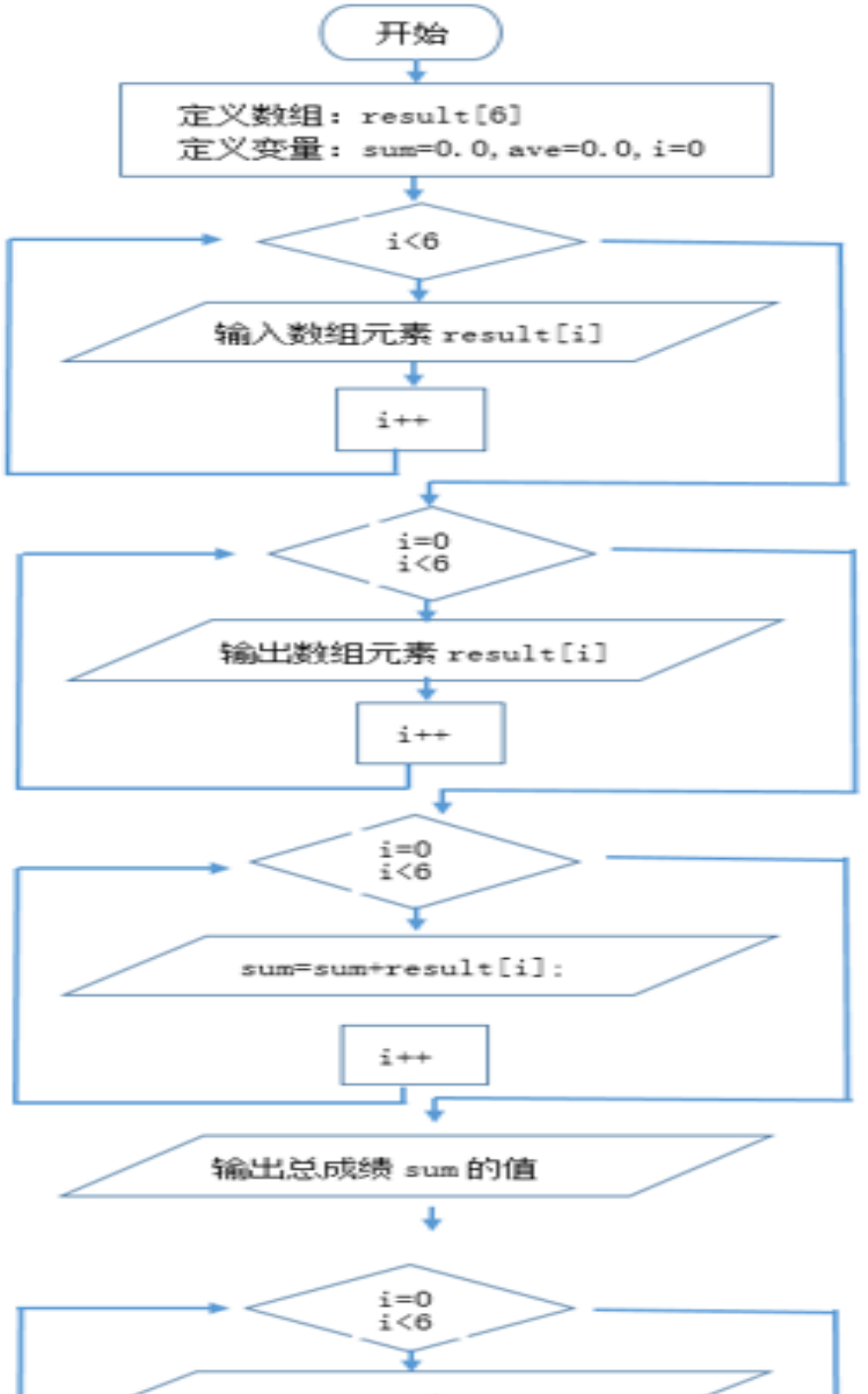
4.求总成绩，存入变量sum中，并输出变量sum的值。

5.求平均成绩，存入变量ave中，并输出变量ave的值。

6.1.4 一维数组的应用

例 6-1

流程图：



6.1.4 一维数组的应用

例 6-1

程序代码：

```
#include"stdio.h"
void main()
{    //定义一个数组
      float result[6],sum=0.0,ave=0.0;
       int i;
       printf("请输入已知的6名同学《C语言程序设计》课程的期末成绩
:\n");
       for(i=0;i<6;i++)
          {
                          scanf("%f",&result[i]);
              }
     //输出，并观察6名同学的成绩在以上数组中的存储
       printf("\n== == == == == == == ==\n");
       printf("\n分别输出6名同学的成绩：\n");
       for(i=0;i<6;i++)
          {
                          printf("result[%d]=%.2f\n",i,result[i]);
              }
     //求总成绩
       printf("\n== == == == == == == ==\n");
```

6.1.4 一维数组的应用

例 6-1

程序执行结果：

```
请输入已知的6名同学《C语言程序设计》课程的期末成绩:
85 80 90 93 78 69

==   ==   == ==   ==   ==   ==   ==

分别输出6名同学的成绩:
result[0]=85.00
result[1]=80.00
result[2]=90.00
result[3]=93.00
result[4]=78.00
result[5]=69.00

==   ==   == ==   ==   ==   ==   ==
6名同学的总成绩为: sum=495.00

==   ==   == ==   ==   ==   ==   ==
6名同学的平均成绩为: ave=82.50
Press any key to continue
```

说明：

（1）%.2f用在格式控制输入输出中，表示保留两位小数。

（2）字符数组的输入输出都是通过循环实现的。

注意：

（1）数组的下标是用方括号括起来，不是圆括号，数组的下标是从0开始的，不是从1开始。

（2）数组的命名同变量的命名规则相同。

任务6.2 多个学生多门课程成绩的总成绩和平均成绩

任务描述

一个学习小组有4个人，每个人有相同三门课程的考试成绩。求组内每个人的总成绩、全组各科的总成绩和各科的平均成绩。请编制一个程序。

任务分析

1）定义一个四行三列的数组，每一行用于存放一名学生的三科成绩。

2）通过遍历数组中的每一个元素，求组内每个人的总成绩、全组各科的总成绩和各科的平均成绩。

Part 2

二维数组的定义及其应用

- 二维数组的定义
- 二维数组的引用
- 二维数组的初始化
- 二维数组的应用

6.2.1 二维数组的定义

二维数组的定义格式为：

<类型说明符> <数组名>[<常量表达式1>] [<常量表达式2>];

- 其中，
- “常量表达式1”表示数组的行数；
- “常量表达式2”表示数组的列数。
- 例如， int b[4][5];

6.2.2 二维数组的引用

1、二维数组的引用方式同一维数组的引用方式一样，也是通过数组名和下标的方式来引用数组元素，引用形式为：

<数组名>[<行下标>][<列下标>]

2、二维数组中元素的存储和排列顺序：
整体按行从上到下存储，行内又按从左到右存放，因此二维数组元素的个数等于行数与列数的乘积。

3、二维数组的行下标和列下标都是从0开始编号。
例如：int b[4][5];
定义名为b的整型二维数组，数组分为4行5列，共20个元素，元素的排列顺序为：

b[0][0] b[0][1] b[0][2] b[0][3] b[0][4]
b[1][0] b[1][1] b[1][2] b[1][3] b[1][4]
b[2][0] b[2][1] b[2][2] b[2][3] b[2][4]
b[3][0] b[3][1] b[3][2] b[3][3] b[3][4]

6.2.3 二维数组的初始化

完成二维数组的定义后，需要对二维数组进行初始化，二维数组初始化的一般格式如下：

<类型说明符><数组名>[<常量表达式1>] [<常量表达式2>] = {初值表

};

初始化二维数组的方式有四种，具体如下：

1）按行给二维数组赋初值

例如：int a[2][3] = {{1,2,3},{4,5,6}};

2）将所有的数组元素按顺序写在一个大括号内

例如：int a[2][3] = {1,2,3,4,5,6};

3）对部分数组元素赋初值

例如：int b[3][4] = {{1},{4,3},{2,1,2}};

4）对全部数组元素置初值，则第一个下标可省略

例如：int a[ ][3] = {1,2,3,4,5,6};

6.2.4 二维数组的应用

例 6-2

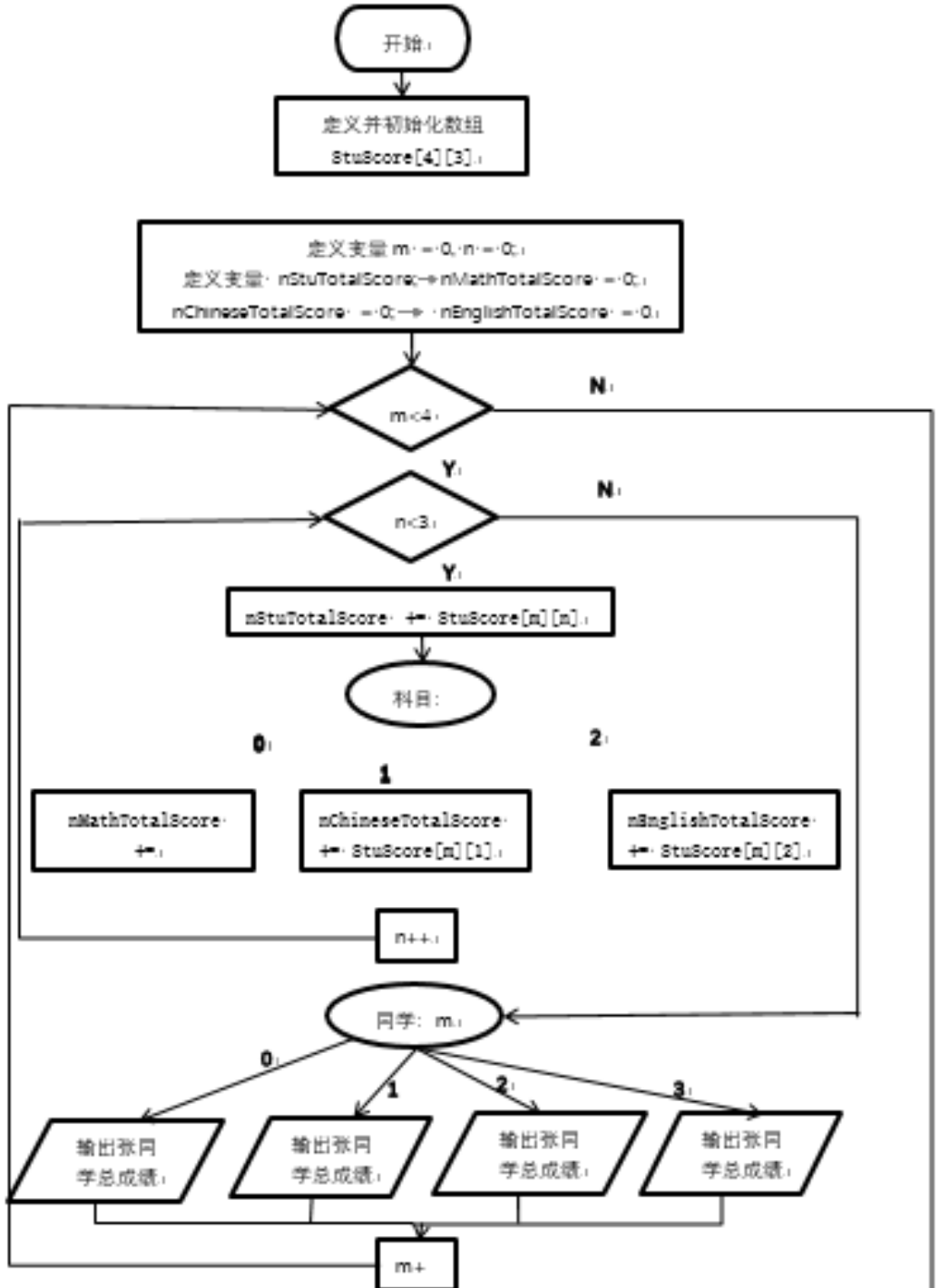
案例：

一个学习小组有4个人，每个人有相同三门课程的考试成绩。请编制一个程序求组内每个人的总成绩、全组各科的总成绩和各科的平均成绩。

6.2.4 二维数组的应用

例 6-2

流程图：



6.2.4 二维数组的应用

例 6-2

程序代码：

```
#include "stdio.h"
void main(int argc)
{
//定义一个四行三列数组存放张同学、王同学、李同学、赵同学四个人成绩
int StuScore[4][3] = {{ 88, 70, 90 },{ 80, 80, 60 },{ 89, 60, 85 },{ 70, 80, 80 }}
int m = 0, n = 0;
int nStuTotalScore;
int nMathTotalScore = 0;
int nChineseTotalScore = 0;
int nEnglishTotalScore = 0;
printf("个人总成绩：\n");
for (m = 0; m < 4; m++)
{
     nStuTotalScore = 0;
     for (n = 0; n < 3; n++)
   {
     nStuTotalScore += StuScore[m][n];
     switch (n)
       {
            case 0: nMathTotalScore += StuScore[m][0];   break;
            case 1: nChineseTotalScore += StuScore[m][1];  break;
            case 2: nEnglishTotalScore += StuScore[m][2];
        }
    }
     switch (m)
      {
            case 0: printf("张同学：%d\n", nStuTotalScore);  break;
            case 1: printf("王同学：%d\n", nStuTotalScore);  break;
            case 2:printf("李同学：%d\n", nStuTotalScore);   break;
            case 3:printf("赵同学：%d\n", nStuTotalScore);   break;
       }
}

 printf("小组数学总分：%d          小组数学平均分：%.2f\n",nMathTotalScore, (double)nMathTotalScore / 4);
 printf("小组语文总分：%d          小组语文平均分：%.2f\n",nChineseTotalScore, (double)nChineseTotalScore /
 printf("小组英语总分：%d          小组英语平均分：%.2f\n",nEnglishTotalScore, (double)nEnglishTotalScore / 4
}
```

6.2.4 二维数组的应用

例 6-2

程序执行结果：

```
个人总成绩:
张同学: 248
王同学: 220
李同学: 234
赵同学: 230
小组数学总分: 327        小组数学平均分: 81.75
小组语文总分: 290        小组语文平均分: 72.50
小组英语总分: 315        小组英语平均分: 78.75
Press any key to continue
```

说明：

（1）其中，定了StuScore[4][3]四行三列数组存放学生成绩，每一行存放一名学生的三科成绩。

（2）通过遍历列下标获取每个小组不同学科的总分，通过遍历行下标获取每个小组成员的总分，最后将小组不同学科的总分和平均分输出。

（3）第一个switch分支case的0、1、2分别表示的是数学、语文、英语成绩。第二个switch分支case的0、1、2、3分别表示的是张同学、王同学、李同学、赵同学四个人的总成绩。

注意：

（1）数组的下标是用方括号括起来，不是圆括号，数组的下标是从0开始的，不是从1开始。

（2）数组的命名同变量的命名规则相同。

任务6.3 使用数组处理字符串

任务描述

已知两个数组str1和str2分别存放字符串“Hello”和“Word”，编写程序复制第1个字符串，并连接字符串1和字符串2，输出连接后的字符串及其长度。

任务分析

（1）定义3个字符数组str1、str2和str3，保证数组长度大于字符串的长度,将“Hello”存入数组str1中、将“Word”存入数组str2中。
（2）通过字符串复制函数将“Hello”复制到数组str3中。
（3）通过字符串连接函数和求字符串长度的函数分别求出连接后的字符串及其长度。

Part 3

使用数组处理字符串

- ◎字符数组的定义、引用和初始化
- ◎字符串
- ◎字符串处理函数
- ◎字符数组的应用

6.3.1 字符数组的定义、引用和初始化

一、定义：字符数组就是数据元素为字符类型的数组，用来存放字符。

字符数组的定义格式为：

char 数组名[常量表达式]；