

# 项目用户菜单设计

模块二介绍了“学生成绩管理系统”的数据定义和运算符相关知识,本模块开始搭建项目框架,完成项目的两级菜单的设计。项目中包含主菜单和子菜单(管理员、学生用户)。系统运行首先进入主菜单,通过选择进入相应用户的子菜单,再选择进入各个菜单功能。由于菜单的选择应该具有重现性和循环性,因此,本模块的主要任务就是完成主菜单和子菜单的循环显示和选择,为下一模块的学生成绩管理功能提供调用界面。

## 【工作任务】

- (1) 任务 3-1: 主菜单显示。
- (2) 任务 3-2: 主菜单选择。
- (3) 任务 3-3: 子菜单选择。
- (4) 任务 3-4: 菜单循环显示。

## 【学习目标】

- (1) 掌握程序控制的顺序、选择和循环三大结构,并能够熟练画出算法流程图。
- (2) 掌握顺序结构的输入输出语句。
- (3) 掌握分支结构的 if 语句和 switch 语句。
- (4) 掌握循环结构的 while 语句和 do while 语句。

## 任务 3-1: 主菜单显示

### 任务描述与分析

周老师将班级所有同学划分为五个项目组,要求每个项目组完成“学生成绩管理系统”的主菜单的显示,具体实现效果如图 3-1 所示。系统运行时,首先进入主菜单,主菜单有 3 个选项,分别为: 1——管理员、2——学生、0——退出。

要完成这个任务,周老师要给项目组的同学们分析一下需要掌握哪些知识。

首先,程序开发前要了解算法流程图和基本的程序控制结构。在进行程序设计之前,要将解决这

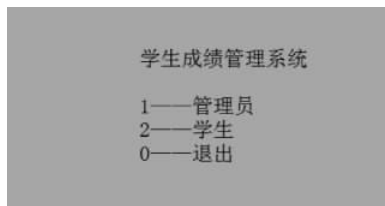


图 3-1 主菜单

个任务的程序结构理清,并将算法描述出来才能进行编码。

其次,本任务需要用到 C 语言中的格式输出语句。

## 相关知识与技能

### 3-1-1 算法和程序结构

程序设计就是面对一个需解决的实际问题,设计适合于计算机的算法,并利用程序设计语言(如 C 语言)写出算法成为程序、运行程序,此问题得以解决。而程序是解决特定问题所需要的语句集合。算法是为了解决某个特定问题而采取的确定有效的步骤。算法的描述可以通过自然语言法、伪代码法、流程图表示法、高级语言表示法。下面介绍一下传统流程图的算法描述方法。

传统流程图符号如图 3-2 所示。



图 3-2 传统流程图符号及功能

基本的程序结构有以下 3 种。

- (1) 顺序结构。语句顺序逐条执行,不发生流程转移,如图 3-3 所示。
- (2) 选择结构。选择结构流程图如图 3-4 所示。



图 3-3 顺序结构流程图

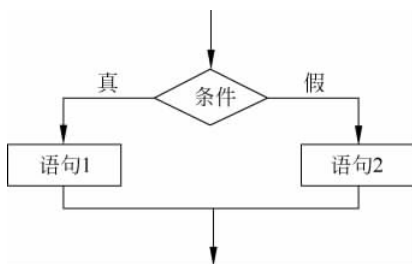


图 3-4 选择结构流程图

- (3) 循环结构。循环结构分为当型循环和直到型循环两种,分别如图 3-5、图 3-6 所示。理论已经证明,利用这三种程序结构可以解决任意问题。

计算机程序解决问题的算法与人们日常解决相同问题的算法相比较:程序设计算法基于日常逻辑,因此计算机擅长大量的重复计算,引入了循环/判断等结构,程序设计算法往往更简洁、清晰、独特。

### 3-1-2 格式化输出语句

#### 1. 格式化输出函数 printf

printf 函数的作用是向系统指定的隐含输出设备输出若干数据。

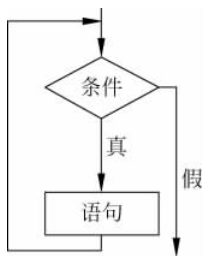


图 3-5 循环结构流程图——当型循环

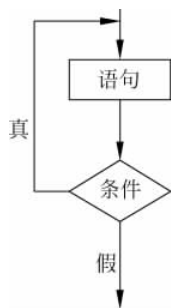


图 3-6 循环结构流程图——直到型循环

printf 函数的语法格式如下。

```
printf("格式控制字符串", 输出项列表);
```

其中,输出项列表可以是常量、变量、表达式、函数调用等;格式控制字符串一般包含两部分,即格式控制符和其余字符。格式控制符是以“%”开头的字符串,控制输出数据的类型和格式。其余字符指原样输出的提示字符串,有几个“%”,就有几个输出项。

那么,“printf(“x= %d,y= %d\n”,x,y);”语句中哪些是格式控制符? 哪些是提示字符串? 会输出什么?

输出用格式控制字符串为“%格式字符”。表 3-1 列出了常用的格式字符。

表 3-1 格式字符

格式字符	说 明
%d,i	输出带符号的十进制数(正数不带符号)
%u	输出无符号的十进制数
%o	输出无符号的八进制数(不输出前导符 0)
%x,X	输出无符号的十六进制数(不输出前导符 0 x)
%c	以字符形式输出 1 个字符
%s	输出 1 个字符串(到第 1 个 '\0' 为止)
%f	输出小数形式的实数(隐含输出 6 位小数)
%e,E	输出指数形式的实数(隐含输出 6 位小数)
%g,G	输出%f和%e中宽度较短的,不输出无意义的 0
%p	输出指针地址
%%	输出%

附加的输出用格式字符串为“%附加格式字符格式字符”,如表 3-2 所示。

## 2. 字符输出函数

字符输出函数为 putchar(ch),可以向终端输出 1 个字符,与 printf 的 %c 格式输出无区别。

例如

```
putchar('y');
putchar('\n');
```

```
putchar(ch);
putchar('0xa');
```

表 3-2 附加格式字符

附加字符	说 明
+	输出的数字总带十号或一号
-	输出的数据在所在域中左对齐
l	输出长整型
m	输出数据的最小宽度
.n	输出数据中小数点后的位数
#	使输出的 8 或 16 进制数带前导 0 或 0x

### 3. 字符串输出函数

字符串输出函数为 puts(字符串常量/字符串地址),可以将字符串内容输出,直至遇到'\0',并且自动换行。

例如,语句

```
printf(" %s", "I am a good student");
```

可以输出字符串"I am a good student",但不会自动换行。语句

```
puts("I am a good student");
```

也可以输出字符串"I am a good student",但会换行。

## 3-1-3 空语句和复合语句

### 1. 空语句

空语句只有一个“;”,语句为空,不执行任何操作,但在构成程序结构或调试阶段还是很有用的。

### 2. 复合语句

多于 1 条的语句用{}括起来,称为复合语句。复合语句在语法上等同于 1 条语句,凡是单个语句出现的地方,都可以出现复合语句,大大增强了程序的处理能力。在复合语句内部可以包含任何数据结构定义和其他语句,在其内部定义的变量只在此复合语句内起作用。

### 3. 注释

注释:为了使编码人员和其他读者更好地理解程序,在程序中写的注解。

//: 用于单行注释;

/\* ... \*/: 用于多行注释或块注释。

注释的内容是不进行编译和执行的,因此注释有两个作用:对程序进行注解;屏蔽不需执行的代码。





入管理员子菜单,当选择 2 时进入学生子菜单,选择 0 时退出系统。

要完成这个任务,周老师要给项目组的同学们分析一下需要掌握哪些知识。

要进行主菜单的选择,首先需要用户从键盘输入数字进行选择,系统接收到输入,然后根据接收到的数字进行判断,再进行跳转。

因此,本任务需要用到 C 语言中的格式输入语句和判断分支语句。

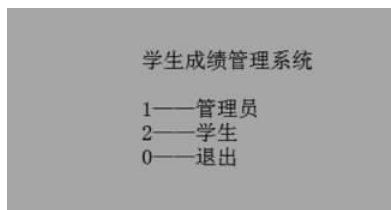


图 3-7 主菜单选择

## 相关知识与技能

### 3-2-1 格式化输入语句

#### 1. 格式化输入函数 scanf

格式化输入函数可以在系统指定的隐含输入设备上输入数据到变量。

scanf 函数的语法格式如下。

```
scanf("格式控制字符串",变量地址 1,变量地址 2,...);
```

有几个%,就有几个&,如“scanf("%d%d",&x,&y);”。

格式控制字符串为“%格式字符”,如表 3-3 所示。

表 3-3 格式控制字符

格式字符	说 明
%d	输入带符号的十进制数(遇空格、回车结束)
%o	输入带符号的八进制数(遇空格、回车结束)
%x	输入带符号的十六进制数(遇空格、回车结束)
%c	输入 1 个字符(遇字符结束)
%s	输入 1 个字符串(遇回车、空格、制表符结束)
%f	输入小数形式的实数(遇空格、回车结束)
%e	输入指数形式的实数(遇空格、回车结束)

scanf 函数的执行过程如下。

(1) 执行到 scanf 语句时,程序停下来,等待用户的输入。

(2) 输入 1 个变量时,请按照以上结束方式结束输入。

(3) 当需要同时输入多个变量时,有以下两种情况。

① scanf 的格式字符串中有分隔符,必须严格按次序输入数值和相应分隔符。

例如,执行语句“scanf("%d,%d",&x,&y);”时,必须输入“3,4”。

② scanf 的格式字符串没有分隔符,可以用空格、跳格、回车符等分隔多个数值。

例如,执行语句“scanf("%d%d",&x,&y);”时,可以输入“3 4”,或者“3<HT> 4”,或者“3<CR>4”。

**【例 3-1】** 设有以下变量,从键盘为其输入值。

```
int a,b;
float x;
char ch1,ch2,stuName[20];
scanf("%d%d%f%c%c%s",&a,&b,&x,&ch1,&ch2,stuName);
```

如果在键盘上这样录入:“3 4 1.2 A B rabby<Enter>”,则各变量的值是: a=3, b=4, x=1.2, ch1=' ', ch2='A', stuName="b"。

如果在键盘上这样录入:“3 4 1.2ABrabby<Enter>”,则各变量的值是: a=3, b=4, x=1.2, ch1='A', ch2='B', stuName="rabby"。

## 2. 字符输入函数

字符输入函数为 c=getchar(),用于从键盘上输入 1 个字符(包括空格等),按 Enter 键确认,函数的返回值就是该字符。

例如:

```
char a,b;
b=getchar();
scanf("%c",&a);
```

这两个输入语句的区别是:前一个语句输入 1 个字符后,需要按 Enter 键才能接收到字符;后一个语句只要输入任何字符,马上就被接收了。也就是说,getchar()与 scanf 的%c 格式字符的用法是有区别的,使用 getchar 函数输入数据需要按 Enter 键确认输入,而使用 scanf 函数则接收当前字符。

## 3. 字符串输入函数

字符串输入函数为 gets(字符串地址),接收从键盘输入的字符串,按 Enter 键结束。

例如,遇到 gets 函数时,如果输入“I am a good student<CR>”,则得到的字符串是“I am a good student”,若使用 scanf 函数的%s 格式符接收此字符串时,由于回车、空格、制表符都是分隔符,因此得到的字符串是“I”。

## 3-2-2 if 语句

用 if 语句可以构成分支结构。它根据给定的条件进行判断,以决定执行某个分支程序段。C 语言的 if 语句有以下三种基本形式。

### 1. 第一种形式为基本形式:if

```
if(表达式)
{
    语句
}
```

语义:如果表达式的值为真,则执行其后的语句,否则不执行该语句,流程如图 3-8 所示。

说明:花括号内的语句如果为单条语句,花括号可以省略,

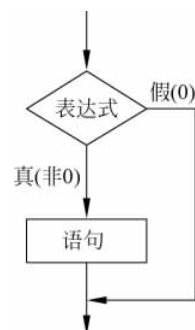


图 3-8 if 语句流程图

否则不能省略。

**【例 3-2】** 比较两个数值的大小,用 if 语句的第一种形式。

```
main()
{
    int a, b, max;
    printf("\n input two numbers:   ");
    scanf("% d % d", &a , &b);
    max = a;
    if(max < b)
        max = b;
    printf("max = % d", max);
}
```

2. 第二种形式为: if-else

```
if(表达式)
{
    语句 1
}
else
{
    语句 2
}
```

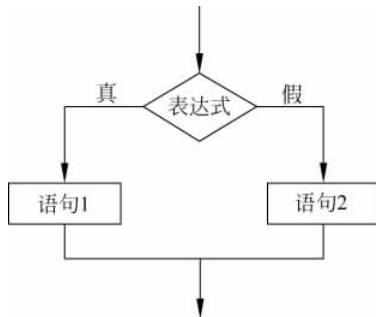


图 3-9 if-else 语句流程图

语义: 如果表达式的值为真,则执行语句 1,否则执行语句 2,流程如图 3-9 所示。

**【例 3-3】** 比较两个数值的大小,用 if 语句的第二种形式重构代码。

```
main()
{
    int a, b, max;
    printf("input two numbers:");
    scanf("% d % d", &a, &b);
    if(a > b)
        max = a ;
    else
        max = b;
    printf("max = % d", max);
}
```

3. 第三种形式为: if-else-if

```
if(表达式 1)
{
    语句 1;
}
```

```

else if(表达式 2)
{
    语句 2;
}
...
else if(表达式 n - 1)
{
    语句 n - 1;
}
else
{
    语句 n;
}

```

语义：依次判断表达式的值，当出现某个值为真时，则执行其对应的语句；然后跳到整个 if 语句之外继续执行程序，如果所有的表达式均为假，则执行语句  $n$ ；然后继续执行后续程序，过程如图 3-10 所示。

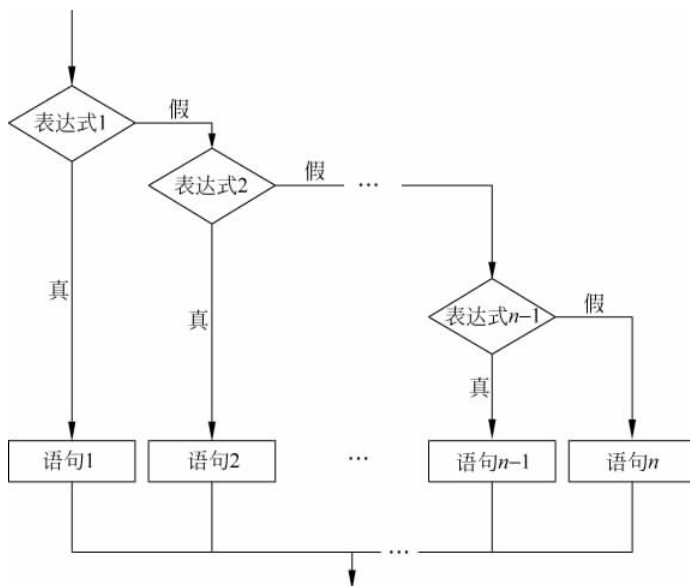


图 3-10 if-else-if 语句流程图

**【例 3-4】** 判别键盘输入字符的类别。

```

main()
{
    char c;
    printf("input a character:   ");
    c = getchar();
    if(c < 32)
        printf("This is a control character\n");
}

```