**A.0003 选择控制结构**

欢迎回来!

什么叫”控制结构”呢?之前我们看到的程序,都是简单地一行一行地执行下来,没什么跳跃或者重复.然而,有时候,我们需要程序能够实现一些重复性的工作(即循环[loop]),比如循环式地读取每一行数据,依次处理,再循环式地输出到屏幕或者文件.又有时候,我们需要选择性的执行某些代码,比如:如果发现数据是不完整的,就报错,不进行处理;如果发现数据是完整的,就进行处理.

控制结构,指的就是选择,循环等语句.这次我们来讲讲如何实现选择结构.

选择结构,有这几种常见写法:

第一种:

if(某条件1)

{

执行某些操作1;

}

第二种:

if(某条件1)

{

执行某些操作1;

}

else

{

执行某操作2;

}

第三种:

if(某条件1)

{

某操作1;

}

elsif(某条件2)

{

某操作2;

}

else

{

某操作3;

}

我们从第一种入手吧.第二三种最后作为推论,很快就能理解.

第一种方框里的代码的意义是:如果条件1满足,那么就执行后面花括号{}里面的语句们;如果条件1不满足,则跳过{}部分,继续执行{}后面的代码.

条件怎么写呢?条件一般就是一些相等或者不等关系.比如:

$x==1

或者:

$x>5

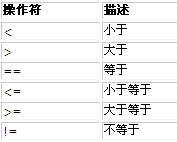
或者:

$x<=4

或者

$x!=76

这里面用到的运算符叫做逻辑运算符



需要注意的是,如果要判断两个数是否相等,要使用==,而不是=.一个等号”=”的意义是赋值,右边的值赋给左边;而两个等号”==”的意义是判断二者是否相等.

我举个第一种选择结构的例子:

$x=<>;

chomp($x);

if($x>=0)

{

$x=$x\*\*0.5;

print ("$x\n");

}

这个程序的意思是,从键盘读入一个x,如果它大于等于0,则计算它的平方根并输出.如果它不满足大于等于0,就什么都没做(程序跳到{}之后,也就是没了).

if语句的分号位置注意一下.if(条件)这句后面没有分号.执行的操作每句后面才有分号.

写条件句的时候,是经常会用到逻辑的”与””或”操作的,与用&&表示,或用||表示.例如:

$x=<>;

chomp($x);

if($x>=0&&$x<=1000)

{

$x=$x\*\*0.5;

print ("$x\n");

}

这样就变成了求[0,1000]区间内的数的平方根.超出此区间,什么都不做.&&,||可以配合括号实现复杂的逻辑关系.

现在我们回过头来看第二种if语法:

if(某条件1)

{

执行某些操作1;

}

else

{

执行某操作2;

}

现在就很好理解这个意思了,就是如果con1满足,就执行op1;如果con1不满足,就执行op2.

再看第三种if语法:

if(某条件1)

{

某操作1;

}

elsif(某条件2)

{

某操作2;

}

else

{

某操作3;

}

这个的意思就是:如果con1满足,执行op1;如果con1不满足但是满足con2,执行op2;如果以上都不满足,执行op3.单从口语语言的角度来理解这些if,else也是一致的.

现在来说明这一点.这里的这个操作[operation, op]是非常宽泛的,可以是一些基本的程序语句,也可以是复杂的if-else语句(即自身),也可以是以后学到的各种,包括循环结构.总之满足相应的条件就会执行对应的语句. 例如:if语句是可以嵌套的,就是可以写成这样子:

if(condition1)

{

operation1;

if(con2)

{

op2;

}

op3;

}

应该能理解吧.同时这也说明了为什么要保持一个好的缩进对齐结构,免得看花眼了.

上面讲的杂七杂八的,不知道读者能懂么.接下来,我们用一个经典例子来说说if语句应该怎么思考规划,怎么写.例子之后就是本次的有爱练习题.

现在我们想做一个闰年判决器.功能是:输入一个正整数年份,输出它是否是闰年.闰年的判断规则(亦有别的等效描述):如果一个年份不是100的整数倍,它能被4整除就是闰年,不能被4整除不是闰年;如果一个年份是100的整数倍,它被400整除就是闰年,不能被400整除就不是闰年.

这个程序怎么一步一步写出来呢?

首先:

$x=<>;

chomp($x);

if($x%100!=0)

{

}

else

{

} #先判断是否被100整除,然后分别处理.%是求余运算.运算符在第二课有讲

思路应该如上,对吧!然后:

$x=<>;

chomp($x);

$y=0; #0-not runnian 1-runnian

if($x%100!=0)

{

if($x%4==0){$y=1;}

}

else

{

if($x%400==0){$y=1;}

}

我们用y来记录判断结果,默认值0(非闰年).

最后,打印出来:

$x=<>;

chomp($x);

$y=0; #0-not runnian 1-runnian

if($x%100!=0)

{

if($x%4==0){$y=1;}

}

else

{

if($x%400==0){$y=1;}

}

if($y==1) {print ("yes\n");}

else {print ("no\n");}

有没有看懂呢?有时候我的op非常短小,我就把{}写到if同一行了.

如果概念和例子都能看懂,那就做做今天的练习题试试吧.

圣诞节前,网上流传一个程序员给女友做的小程序,这个图特别大,我贴到下一页了.

先忽略评论部分,我来抽象描述一下这个程序的功能.为了方便,我们把里面语言改为英文.为了符合我们现有的知识,我把功能也稍作改动.

程序开始,”what’s your name?”这个问题

标准答案”tangcheng”.如果答错,提示”wrong answer!”,然后程序结束(没有第二次机会).如果答对,问下一个问题.”

第二个问题,”are you beautiful?”

标准答案”yes”.如果答错,提示”wrong answer!”,然后程序结束.如果答对,则显示一些节日快乐的信息,或者随便什么图形吧.比如我们就显示”Merry Xmax, my honey!” 这里程序结束.

这里说一下,进行字符串比较的时候,这么写:

if ($x eq "tangcheng")

即用eq代替==

我写了一下,这是几张效果图:

