8.3-switch语句

语法：

switch（表达式）

{

case 结果1：执行语句; break; //解释为什么加break

case 结果2：执行语句; break;

...

default：执行语句; break;

}

比如是结果2，是从结果2开始执行下面结果3、4、5...和default对应的全部代码，所以要用break跳出

缺点：判定的表达式的结果只能是整型或、字符型或枚举型，其余不支持，且不能是一个范围

优点：处理一分为多的情况时，代码长度和执行速度均有优势

示例代码1：

问题情景：

输入一个学生的分数（分数都是整数，100分满分），设定80-100分评定为优，60-79分评定为合格，0-59之间评定为不合格，若输入的数据不在0-100之间，则输出错误提示

int s;

cin>>s;

switch(s/10)

{

case 10:

cout<<"Good"<<endl;

break;

case 9:

cout<<"Good"<<endl;

break;

case 8:

cout<<"Good"<<endl; //8、9、10可合并

break;

case 7:

cout<<"OK"<<endl;

break;

case 6:

cout<<"OK"<<endl;

break;

case 5:

cout<<"Bad"<<endl;

break;

case 4:

cout<<"Bad"<<endl;

break;

case 3:

cout<<"Bad"<<endl;

break;

case 2:

cout<<"Bad"<<endl;

break;

case 1:

cout<<"Bad"<<endl;

break;

case 0:

cout<<"Bad"<<endl;

break;

default:

cout<<"ERROR"<<endl; //输入负数的时候有问题，加if-else修改

}

示例代码2：

问题情景：输入两个浮点数，再根据输入的运算符完成加减乘除四则运算

double a,b;

char ch;

cout<<"请输入两个浮点数："<<endl;

cin>>a>>b;

cout<<"请选择你要完成的运算类型："<<endl;

cout<<"+ - \* /"<<endl;

cin>>ch;

switch(ch)

{

case('+'):

cout<<"a+b="<<a+b<<endl;

break;

case('-'):

cout<<"a-b="<<a-b<<endl;

break;

case('\*'):

cout<<"a\*b="<<a\*b<<endl;

break;

case('/'):

cout<<"a/b="<<a/b<<endl;

break;

default:

cout<<"ERROR"<<endl;

break;

}