

程序样例：五子棋游戏程序代码

说明：下面是张晋珩同学编写的程序。从认真态度、自学精神、创新意识和代码能力都使得我感动、值得我学习。我征求其意见，共享共勉。

//老师。之前我传给您的版本没有注释，且平局判断过于复杂冗长，因此我对其进行了完善，并进行了注释，还望您重新验收、评定。下面是我对程序的总体说明：

//首先，考虑到用户的游戏体验，游戏先询问了玩家的名字，并将其储存下来，以在以后称呼对应的玩家；然后，询问 player1 的棋型，通过输入 12 代表黑白，并且如果输入的不是 1 或 2，则会报错并提示玩家重新输入。然后进行简短的游戏说明，就开始绘制棋盘开始游戏。在游戏过程中，玩家通过输入棋子坐标的方式下棋，输入的 x、y 对应意思是“x 行 y 列”。若输入的坐标为小数，则通过 floor 函数将其转化为不超过它的最大整数。若转化后的整数仍不属于 0~14，则会报错，提示玩家重新输入。每一步下完棋，交给对手下时，都会为后者提示上一步对手下的位置。每次下完棋都通过系统 cls 命令清屏、重新绘制棋盘，棋盘的绘制是通过记录二维数组的每个元素的值（0、1、2），来判断该在此处画什么形状。至于胜负判断，我们只需要写出四个判断函数，分别是横连线、竖连线、一三象限斜连线、二四象限斜连线。判断胜负无果后再判断平局。这样，我们就可以判断输赢，然后进行结尾的输出，也就是宣布获胜的一方（或平局），然后说“欢迎使用，再见”之类的话。需要注意的是，题目中要求记录双方对弈的点数，我选择用静态类成员变量，理由很简单，因为游戏中有两个玩家类，但是我们要想让两个玩家操作的时候这个计数变量均自增，因此 static 变量就是一个优选。最后输出这一数据的值即可。为了契合我们所学的内容，析构函数执行时进行了语句说明，体现了三个对象的析构过程。这些就是我五子棋程序的要点。

//具体的注释在附件中的程序源代码。

//请用 VS2012 等对中文支持较好的编译器编译验收，谢谢。

```
#include<iostream>
#include<cstring>
#include<cstdlib>
#include<cmath>
using namespace std;

int board[15][15]={0}; //
定义一个二维数组，来储存棋盘上各点的棋子情况

char type[11]; //
全局变量，代表 Player 1 棋型
```

```

bool ping()
    //判断棋盘是否已下满
{
    bool p=true;
    for(int i=0;i<15;i++)
    {
        for(int j=0;j<15;j++)
        {
            //遍历棋盘
            if(board[i][j]==0)
            {
                p=false;
                break;
            }
        }
    }
    return p;
}

```

```

class ChessBoard
{
    private:
        int x;
        int y;
        //棋子位置
        int size;
        //
        棋盘大小
    public:

        void show();
        //
        展示棋子位置
        void setx(int a);
        void sety(int a);
        //
        设定棋子位置
        ChessBoard(int sizee=15,int xx=0,int yy=0):size(sizee),x(xx),y(yy){} //构造函数
        ~ChessBoard()
        //析构函数
        {
            cout<<"Destructor called."<<endl;
            //
            调用语句
        }
};

```

```

void ChessBoard::show()

```

```

        //显示对手的上一步
    {
        cout<<"Your opponent's last step:"<<endl<<"Location x is: "<<x<<endl<<"Location y
is: "<<y<<endl;
    }

void ChessBoard::setx(int a)
{
    x=a;
}

void ChessBoard::sety(int a)
{
    y=a;
}

class player //
建立玩家类
{
    private:
        char name[21];
        //玩家姓名
        enum chesstype
        //枚举变量，定义棋型
        {
            black=1,white
        };
        float px;
        float py;
    public:
        static int numb; //静态成员，记录对弈点数
        void setchess(int n); //玩家 n（1 或 2）下棋
        int givex(); //访问 x 的端口
        int givey(); //访问 y 的端口
        void give_name(int n); //赋予名字
        void put_name(); //输出名字
        void pick_type(); //选择棋型
        void show_type(); //展示棋型
        ~player() //player 类的析构函数
        {
            cout<<"Destructor called."<<endl;
        }
};

```

```

void player::give_name(int n)
{
    printf("Player %d,please enter your name (no longer than 20 letters):",n);
    gets(name);
    cout<<name<<" ,welcome!"<<endl;
}

void player::show_type()
{
    printf("So,%s,your chess type is:",name);
    puts(type);
}

void player::pick_type()
{
    cout<<"\n"<<name<<" ,which colour would you want to play?\nOne thing you need to
know is that black plays first.\nPlease enter '1' for black or '2' for white:";
    int a;
    cin>>a;
    switch(a)
    {
        case 1:strcpy(type,"black");break;
        case 2:strcpy(type,"white");break;
        default:cout<<"Input error!"<<endl;pick_type();break; //防止恶意输入，提示
Input error
    }
}

```

```

void player::setchess(int n) //程序中最重要函数之一：下棋函数
{
    printf("%s,please set chess. Enter the x&y location of it.Input example: 7 7 \n",name);
    cin>>px>>py;
    int a,b;
    a=floor(px);
    b=floor(py); //防止恶意输入小数
    bool loclegal1=false;
    bool loclegal2=false;
    bool loclegal=true; //三个布尔型变量，判断坐标的合法性
    for(int i=0;i<15;i++)
    {
        if(floor(px)==i) loclegal1=true;
        if(floor(py)==i) loclegal2=true; //遍历判断法
    }
    if((loclegal1==false)||loclegal2==false||board[a][b]!=0) loclegal=false; //数据越界
}

```

或下在已有棋子的位置上均会报错

```
    if(!oclegal==false)
    {
        cout<<"Input error!"<<endl;
        setchess(n);
    }
    if(oclegal)
    {
        board[a][b]=n;           //如果输入正确，就为二维数组赋值，并增加棋子数
        numb++;
    }
}

int player::givex()
{
    return px;
}

int player::givey()
{
    return py;
}

void player::put_name()
{
    cout<<name;
}

bool win_xie_1(int n)           //判断有无二四象限斜连线
{
    bool b=false;
    for(int i=0;i<11;i++)
    {
        for(int j=0;j<11;j++)           //双层循环遍历棋盘
        {

            if(board[i][j]==n&&board[i+1][j+1]==n&&board[i+2][j+2]==n&&board[i+3][j+3]==n&
&board[i+4][j+4]==n)
            {
                b=true;
                break;
            }
        }
        if(b==true) break;
    }
}
```

```

    }
    return b;
}

```

```

bool win_xie_2(int n)                                //判断有无一三象限斜连线
{
    bool b=false;
    for(int i=0;i<11;i++)
    {
        for(int j=4;j<15;j++)
        {
            if(board[i][j]==n&&board[i+1][j-1]==n&&board[i+2][j-2]==n&&board[i+3][j-3]==n&&board[i+4][j-4]==n)
            {
                b=true;
                break;
            }
        }
        if(b==true) break;
    }
    return b;
}

```

```

bool win_shu(int n)                                  //判断有无竖连线
{
    bool b=false;
    for(int i=0;i<11;i++)
    {
        for(int j=0;j<15;j++)
        {
            if(board[i][j]==n&&board[i+1][j]==n&&board[i+2][j]==n&&board[i+3][j]==n&&board[i+4][j]==n)
            {
                b=true;
                break;
            }
        }
        if(b==true) break;
    }
    return b;
}

```

```

bool win_heng(int n)                                //判断有无横连线

```

```
{
    bool b=false;
    for(int i=0;i<15;i++)
    {
        for(int j=0;j<11;j++)
        {

            if(board[i][j]==n&&board[i][j+1]==n&&board[i][j+2]==n&&board[i][j+3]==n&&board[
i][j+4]==n)
                {
                    b=true;
                    break;
                }
        }
        if(b==true) break;
    }
    return b;
}
```

```
void drawboard_black() //当玩家 1 选择黑子时，棋盘的绘制函数
{
    char sign[15][15][4];
    for(int i=0;i<15;i++)
    {
        for(int j=0;j<15;j++)
        {
            if(board[i][j]==0)
                strcpy(sign[i][j]," \0");
            if(board[i][j]==1)
                strcpy(sign[i][j],"●\0");
            if(board[i][j]==2)
                strcpy(sign[i][j],"◎\0");           //根据二维数组元素的值绘制对应位置
        }
    }
    printf("\nEnter location to set chess.\n");
    printf("          0   1   2   3   4   5   6   7   8   9   10  11  12  13  14\n");
    printf("┌───────────┴───────────┐\n");
    for(int i=0,j=0;i<14;i++,j++)
    {
```



```

        printf("
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----| \n");
    }
    printf("
14 | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %3s | %
d\n",sign[14][0],sign[14][1],sign[14][2],sign[14][3],sign[14][4],sign[14][5],sign[14][6],sign[14][7
],sign[14][8],sign[14][9],sign[14][10],sign[14][11],sign[14][12],sign[14][13],sign[14][14],14);
    printf("
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----| \n");
    printf("          0  1  2  3  4  5  6  7  8  9  10  11  12  13  14
\n");
}

```

```

void claim()                                //有关游戏的部分说明
{
    printf("\nAttention:\n\nl.The vertical direction location,called 'hang' in Chinese,is
considered x.\n\nl.If you input decimal data as location,the program will replace it with the
largest integer smaller than it.\nExample:'Location x is 14.3' is equal to 'Location x is 14'.\n");
    cout<<"\nGet ready for your fierce battle!\n"<<endl;
}

```

```

int player::numb=0;                        //类外初始化静态变量

```

```

int main(){                                //开始主游戏程序
    ChessBoard Chess;
    player P1,P2;                          //创建三个对象
    int a[3]={0};
    bool pingju=false;                    //判断是否平局的变量
    P1.give_name(1);
    P2.give_name(2);                      //让玩家输入他们的名字
    P1.pick_type();
    P1.show_type();                      //棋型的选择与显示
    printf("Thus,");
    P2.put_name();
    printf(",you play the other colour of chess.\n\n");
    claim();                             //细枝末节的提示
    if(strcmp(type,"black")==0)           //如果玩家一执黑，主程序主体
    {
        do{
            drawboard_black();            //绘制棋盘
            P1.setchess(1);
            Chess.setx(P1.givex());

```

```

Chess.sety(P1.givey());          //下棋
if(win_xie_1(1)||win_shu(1)||win_heng(1)||win_xie_2(1))
{
    a[1]=1;
    system("cls");
    drawboard_black();
    break;
}                                //判断胜负
if(ping())
{
    pingju=true;
    break;
}
system("cls");                   //清屏
Chess.show();                   //提示上一步对手的走向
drawboard_black();
P2.setchess(2);
Chess.setx(P2.givex());
Chess.sety(P2.givey());
if(win_xie_1(2)||win_shu(2)||win_heng(2)||win_xie_2(2))
{
    a[2]=1;
    system("cls");
    drawboard_black();
    break;
}
if(ping())
{
    pingju=true;
    break;
}
system("cls");
Chess.show();
}while(1);                      //循环结束的唯一条件是分出胜负或平局
}
if(strcmp(type,"white")==0)     //如果玩家一执白，主程序主体
{
    do{
        drawboard_white();
        P2.setchess(2);
        Chess.setx(P2.givex());
        Chess.sety(P2.givey());
        if(win_xie_1(2)||win_shu(2)||win_heng(2)||win_xie_2(2))
        {

```

```

        a[2]=1;
        system("cls");
        drawboard_white();
        break;
    }
    if(ping())
    {
        pingju=true;
        break;
    }
    system("cls");
    Chess.show();
    drawboard_white();
    P1.setchess(1);
    Chess.setx(P1.givex());
    Chess.sety(P1.givey());
    if(win_xie_1(1)||win_shu(1)||win_heng(1)||win_xie_2(1))
    {
        a[1]=1;
        system("cls");
        drawboard_white();
        break;
    }
    if(ping())
    {
        pingju=true;
        break;
    }
    system("cls");
    Chess.show();
}while(1);
}
if(a[1]==1)                //如果玩家 1 胜利的情况
{
    P1.put_name();
    printf(" wins!\nThere are totally %d chesses played.\nThank you for playing my
game.Goodbye.\n",player::numb);
}
if(a[2]==1)                //如果玩家 2 胜利的情况
{
    P2.put_name();
    printf(" wins!\nThere are totally %d chesses played.\nThank you for playing my
game.Goodbye.\n",player::numb);
}

```

```
    if(pingju) printf("You two reaches a peace situation!\nThere are totally %d chesses  
played.\nThank you for playing my game.Goodbye.\n",player::numb);  
    return 0;  
}
```