

Java五子棋作业

代码主要分为两个 .java 文件，一共有两个类 ChessBoard 和 Player ，
main 函数写在 Player.java 文件中，可以直接从 main 开始看。

1 代码

ChessBoard.java :

```
1  public class ChessBoard {
2      /*
3       * 这个棋盘是15路的，因为文本难以打印出交点上的棋子，
4       * 所以将以横15纵15的方格形式输出
5       */
6
7      public char[][] chessBoard=new char[15][15];
8
9      public static int WHITE=1;
10     public static int BLACK=0;
11
12     public ChessBoard(){
13         for(int i=0;i<15;i++)
14             for(int j=0;j<15;j++)
15                 chessBoard[i][j]='.';
16     }
17
18     public void printBoard(){
19         char[] a=
20         {'1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
21
22         System.out.print("  ");
23         for(int i=0;i<15;i++)
24             System.out.print(a[i]+" "); //输出列号
25         System.out.print("\n");
26
27         for(int i=0;i<15;i++){
28             System.out.print(a[i]+" "); //输出行号
29             for(int j=0;j<15;j++)
30                 System.out.print(chessBoard[i][j]+" ");
31             System.out.print("\n");
32         }
33     }
```

```

33
34     public void putChess(int x, int y, int color){
35         //0: black
36         //1: white
37         char chess=(color==WHITE)?'O':'X';
38         chessBoard[x-1][y-1]=chess;
39
40         if(appearWinner(x,y,chess))
41             success(color);
42     }
43
44     public boolean occupied(int x,int y){
45         return (chessBoard[x-1][y-1]!='. ');
46     }
47
48     public boolean appearWinner(int x,int y,char chess){
49         /*
50          * para-----
51          * x,y: 当前棋子的坐标
52          * color: 当前执子方的颜色，值可能是 'X' || 'O'
53          *
54          * return-----
55          * 是否在任意一个方向上连成五子
56          */
57
58         //只要这次落子引起了任何一个方向上的五子连珠的形成，就直接这一方
59         success
60         return checkLR(x,y,chess)
61             || checkUD(x,y,chess)
62             || checkLURD(x,y,chess)
63             || checkRULD(x,y,chess);
64     }
65
66     public boolean checkLR(int x,int y,char color){
67         // -
68         //连起来一共有多长，1是当前的落子
69         int sum=1;
70         //从当前棋子出发向左
71         for(int i=1;i<5;i++){
72             int xx=x-1 - i,yy=y-1;
73             //直到这个位置的棋子和当前落子颜色不一样停止。
74             if((xx<0) || (xx>14) || yy<0 || yy>14)
75                 break;
76             if (chessBoard[xx][yy] != color)
77                 break;
78             sum++;
79         }
80         //从当前棋子出发向右
81         for(int i=1;i<5;i++){

```

```

81         int xx=x-1 + i,yy=y-1;
82         if((xx<0)||(xx>14)||yy<0||yy>14)
83             break;
84         if (chessBoard[xx][yy] ≠ color)
85             break;
86         sum++;
87     }
88     return sum == 5;
89 }
90 public boolean checkUD(int x,int y,char color){
91     // |
92     int sum=1;
93     for(int i=1;i<5;i++){
94         int xx=x-1 ,yy=y-1-i;
95         if((xx<0)||(xx>14)||yy<0||yy>14)
96             break;
97         if (chessBoard[xx][yy] ≠ color)
98             break;
99         sum++;
100     }
101     for(int i=1;i<5;i++){
102         int xx=x-1 ,yy=y-1+i;
103         if((xx<0)||(xx>14)||yy<0||yy>14)
104             break;
105         if (chessBoard[xx][yy] ≠ color)
106             break;
107         sum++;
108     }
109     return sum == 5;
110 }
111 public boolean checkLURD(int x,int y,char color){
112     // \
113     int sum=1;
114     for(int i=1;i<5;i++){
115         int xx=x-1-i ,yy=y-1+i;
116         if((xx<0)||(xx>14)||yy<0||yy>14)
117             break;
118         if (chessBoard[xx][yy] ≠ color)
119             break;
120         sum++;
121     }
122     for(int i=1;i<5;i++){
123         int xx=x-1+i ,yy=y-1-i;
124         if((xx<0)||(xx>14)||yy<0||yy>14)
125             break;
126         if (chessBoard[xx][yy] ≠ color)
127             break;
128         sum++;
129     }

```

```

130         return sum = 5;
131     }
132     public boolean checkRULD(int x,int y,char color){
133         // /
134         int sum=1;
135         for(int i=1;i<5;i++){
136             int xx=x-1+i ,yy=y-1+i;
137             if((xx<0) || (xx>14) || yy<0 || yy>14)
138                 break;
139             if (chessBoard[xx][yy] != color)
140                 break;
141             sum++;
142         }
143         for(int i=1;i<5;i++){
144             int xx=x-1-i ,yy=y-1-i;
145             if((xx<0) || (xx>14) || yy<0 || yy>14)
146                 break;
147             if (chessBoard[xx][yy] != color)
148                 break;
149             sum++;
150         }
151         return sum = 5;
152     }
153
154     public void success(int color){
155         //某方成功之后输出成功的信息并且退出程序
156         String tmp=(color==WHITE)?"WHITE wins!":"BLACK wins!";
157         printBoard();
158         System.out.println(tmp);
159         System.exit(0);
160     }
161
162 }

```

Player.java :

```

1  import java.util.Scanner;
2
3  public class Player {
4
5      int color;
6      public static int WHITE=1;
7      public static int BLACK=0;
8
9      public Player(int c){
10         color=c;
11     }
12     public void play(ChessBoard cb){

```

```

13         String tmp=(color==WHITE)?"P2's turn, you control
white.":"P1's turn, you control black.";
14         System.out.println(tmp);
15         System.out.println("Input the position you want to place
your chess, e.g. \"1 2\":");
16
17         Scanner sc=new Scanner(System.in);
18         String str=null;
19         str=sc.nextLine();
20         char x1 = str.charAt(0);
21         char y1 = str.charAt(2);
22         //从输入获取真实坐标
23         int x=chToInt(x1);
24         int y=chToInt(y1);
25
26         //把输入限制在(1~F)之间，而且输入的坐标不可以是被占用的，如果不对就接
        着输入。
27         while(!((x1>'0'&&x1≤'9' || x1≥'A'&&x1≤'F')&&
        (y1>'0'&&y1≤'9' || y1≥'A'&&y1≤'F'))
28             ||cb.occupied(x,y)){
29             System.out.println("please give right input!");
30             str=sc.nextLine();
31             x1 = str.charAt(0);
32             y1 = str.charAt(2);
33             x=chToInt(x1);
34             y=chToInt(y1);
35         }
36
37         cb.putChess(x,y,color);
38     }
39
40     public static void main(String[] args){
41         Player p1=new Player(BLACK);
42         Player p2=new Player(WHITE);
43         ChessBoard cb=new ChessBoard();
44         cb.printBoard();
45
46         //黑子先行，轮流下棋。
47         //15*15的棋盘，最多下225个棋子，先下一个黑子，然后112回合
48         p1.play(cb);
49         cb.printBoard();
50         for(int i=0;i<112;i++){
51             p2.play(cb);
52             cb.printBoard();
53             p1.play(cb);
54             cb.printBoard();
55         }
56
57         //112回合过后如果还没有胜利者，输出平局（否则会在中间就退出程序）

```

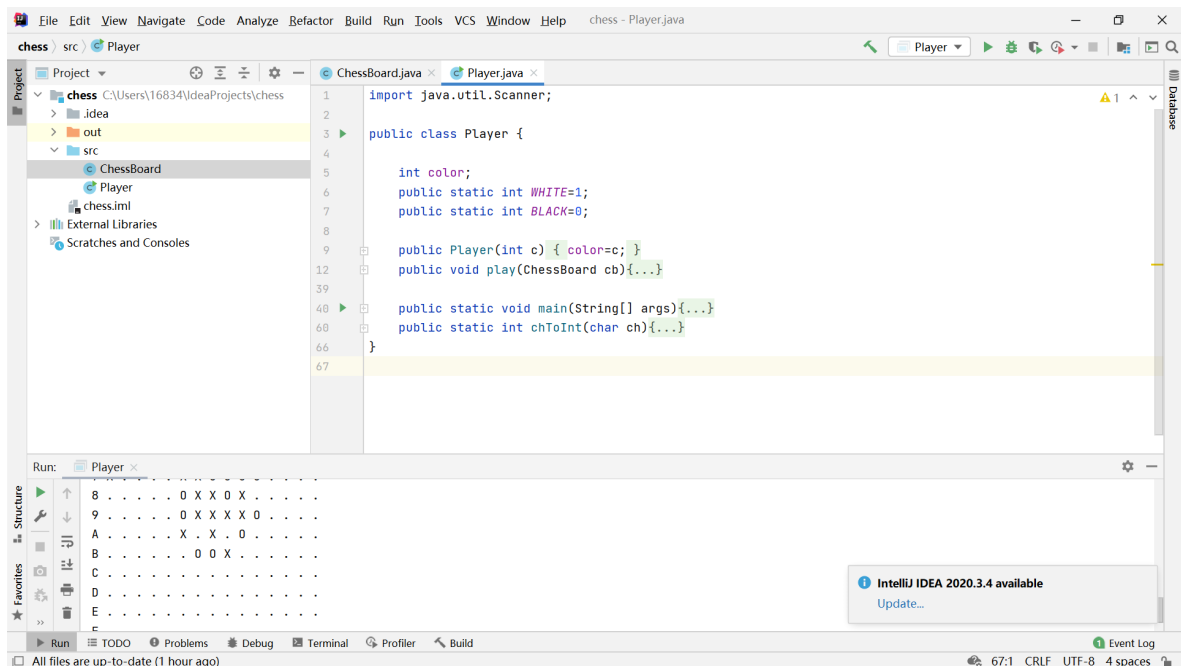
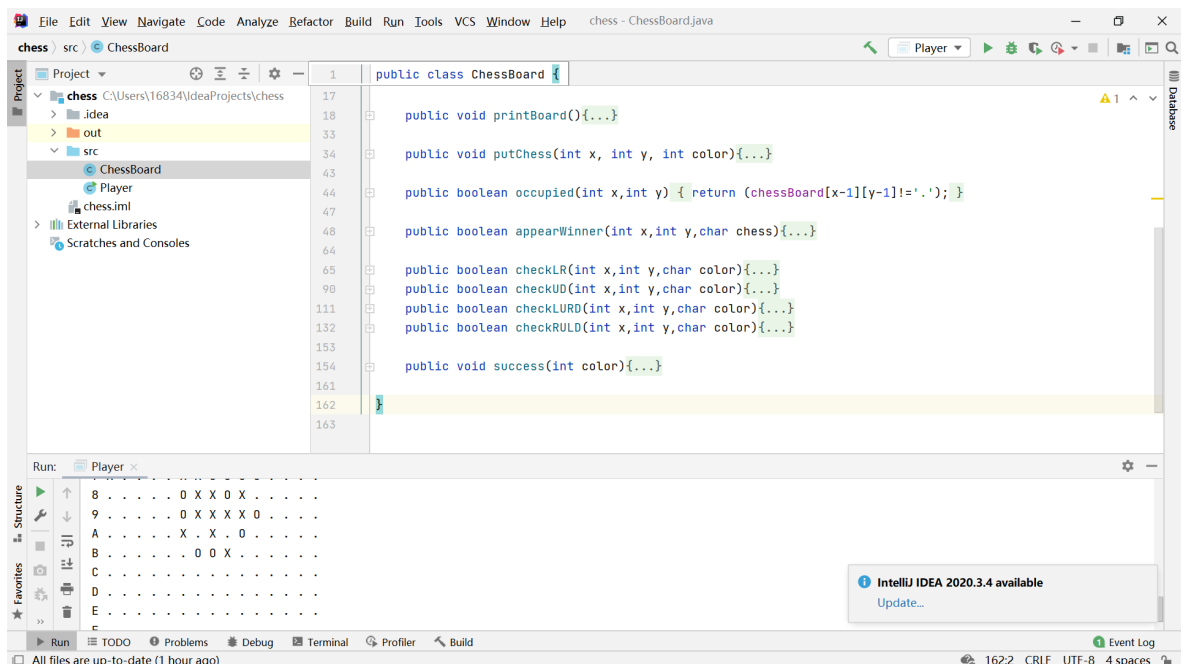
```

58         System.out.println("The result of this game is a draw.");
59     }
60     public static int chToInt(char ch){
61         if(ch>='1'&&ch<='9')
62             return ch-'0';
63         else
64             return ch-55;
65     }
66 }

```

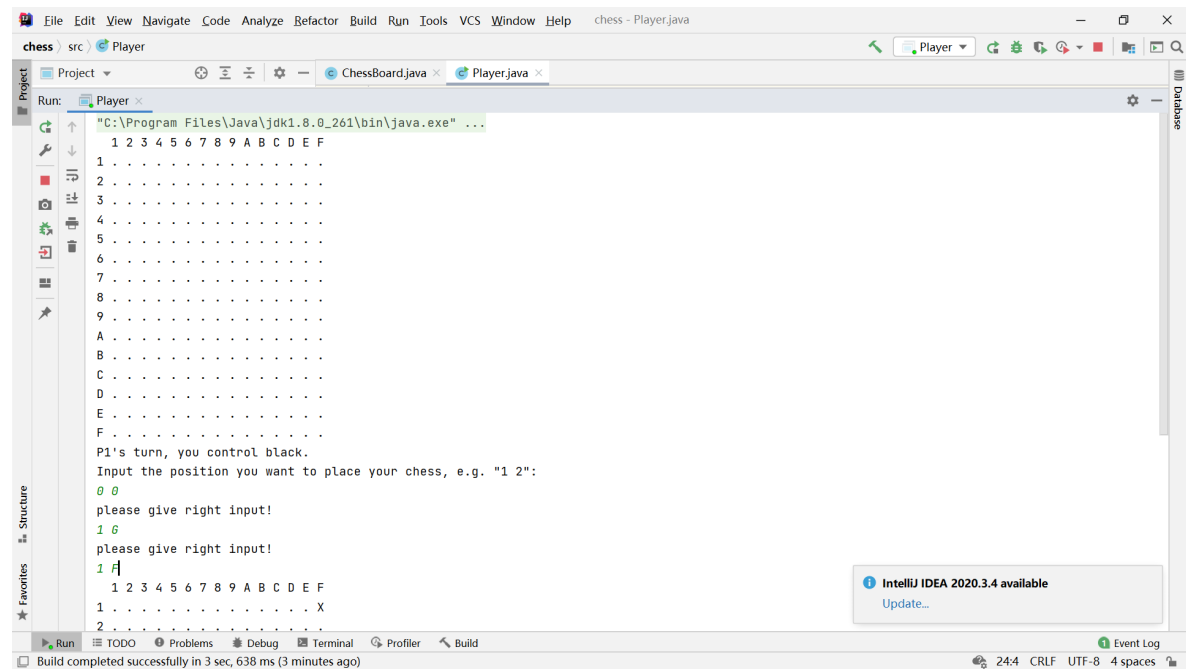
2 部分代码屏幕截图

因为当小项目写的，idea比较顺手（虽然这个程序确实太小了）。



3 结果展示

处理越界展示：



```
chess - Player.java
chess - src - Player
Run: Player
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
 1 2 3 4 5 6 7 8 9 A B C D E F
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
A . . . . . . . . . .
B . . . . . . . . . .
C . . . . . . . . . .
D . . . . . . . . . .
E . . . . . . . . . .
F . . . . . . . . . .
P1's turn, you control black.
Input the position you want to place your chess, e.g. "1 2":
0 0
please give right input!
1 6
please give right input!
1 1
 1 2 3 4 5 6 7 8 9 A B C D E F
1 . . . . . . . . . . X
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
A . . . . . . . . . .
B . . . . . . . . . .
C . . . . . . . . . .
D . . . . . . . . . .
E . . . . . . . . . .
F . . . . . . . . . .
P2's turn, you control white.
Input the position you want to place your chess, e.g. "1 2":
1 F
please give right input!
|
```

处理试图在已经有棋子的地方下棋展示：

```
 1 2 3 4 5 6 7 8 9 A B C D E F
1 . . . . . . . . . . X
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
A . . . . . . . . . .
B . . . . . . . . . .
C . . . . . . . . . .
D . . . . . . . . . .
E . . . . . . . . . .
F . . . . . . . . . .
P2's turn, you control white.
Input the position you want to place your chess, e.g. "1 2":
1 F
please give right input!
|
```

另外因为只读入两个char（以空格为分界）作为输入造成的奇妙bug（或许不算bug）：

```
P2's turn, you control white.
Input the position you want to place your chess, e.g. "1 2":
1 F
please give right input!
7 100
  1 2 3 4 5 6 7 8 9 A B C D E F
1 . . . . . . . . . . . . . . X
2 . . . . . . . . . . . . . .
3 . . . . . . . . . . . . . .
4 . . . . . . . . . . . . . .
5 . . . . . . . . . . . . . .
6 . . . . . . . . . . . . . .
7 0 . . . . . . . . . . . . . .
8 . . . . . . . . . . . . . .
9 . . . . . . . . . . . . . .
A . . . . . . . . . . . . . .
B . . . . . . . . . . . . . .
C . . . . . . . . . . . . . .
D . . . . . . . . . . . . . .
E . . . . . . . . . . . . . .
F . . . . . . . . . . . . . .
P1's turn, you control black.
Input the position you want to place your chess, e.g. "1 2":
```

一方赢棋展示（四种赢法） - | \ / :

```
P2's turn, you control white.
Input the position you want to place your chess, e.g. "1 2":
1 5
  1 2 3 4 5 6 7 8 9 A B C D E F
1 0 0 0 0 0 . . . . . . . . .
2 . . . . . . . . . . . . . .
3 . . . . . . . . . . . . . .
4 . . . . X . . . . . . . . .
5 . . . . X . . . . . . . . .
6 . . . . . . . . . . . . . .
7 . . . . . . . X . . . . . .
8 . . . . . . . X . . . . . .
9 . . . . . . . . . . . . . .
A . . . . . . . . X . . . . .
B . . . . . . . . . . . . . .
C . . . . . . . . . . . . . .
D . . . . . . . . . . . . . .
E . . . . . . . . . . . . . .
F . . . . . . . . . . . . . .
WHITE wins!

Process finished with exit code 0
|
```


P1's turn, you control black.

Input the position you want to place your chess, e.g. "1 2":

5 F

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	X
2	X
3	.	.	.	0	X
4	0	X
5	X
6
7	0	0
8
9
A
B
C
D
E
F

BLACK wins!

Process finished with exit code 0

|

P1's turn, you control black.

Input the position you want to place your chess, e.g. "1 2":

7 8

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	0
2	0
3	.	.	.	X
4	X
5	X
6	X
7	X
8
9
A	0
B
C
D
E	0
F

BLACK wins!

Process finished with exit code 0

|

P1's turn, you control black.
Input the position you want to place your chess, e.g. "1 2":

```
3 8
  1 2 3 4 5 6 7 8 9 A B C D E F
1 . 0 . . . . . . . . . 0 .
2 . . . . . . . . . . .
3 . . . . . . X . . . . .
4 . . . . . X . . . . .
5 . . . . . X . . . . .
6 . . . . X . . . . .
7 . . . X . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
A . . . . . . . 0 . . .
B . . . . . . . . . .
C . . . . . . . . . .
D . . . . . . . . . .
E . . . . . . . . . .
F . . 0 . . . . . . . .
BLACK wins!
```

Process finished with exit code 0

平局由于需要太多输入（棋盘是15*15的，需要225个输入），我没有精力，就没有写。