```
//  ============================================================
//   【名    称】オセロゲーム Othello Version 0.52
//   【著 作 権】Copyright(C) koyuki
//   【動作確認】Windows XP
//   【 種  別 】フリー  JAVAソース
//   【転載条件】koyukiに連絡すること
//               http://w7.oroti.com/~koyuki/php/mail/mail.html
//               last update : Sep.01.2009(Tue)
//  ============================================================

import java.util.Scanner;
public class Othello05{
    static byte yx = 1;                     //○先攻  1:○ 2:●
    static boolean place, place2, sirokuro; //booleanの初期値：false
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        byte[][] ishi = new byte[8][8];
        ishi[3][3]=ishi[4][4]=1;
        ishi[3][4]=ishi[4][3]=2;
        int end=0;          //終了フラグ
        do{
             byte siro=0;   //点数＆ゲーム終了フラグ
             byte kuro=0;   //点数＆ゲーム終了フラグ
            //★★★表示★★★
            System.out.println("次の手は・・・" + ((yx==1) ? "○" : "●"));
            System.out.print(" x:");
            for(int m=0 ; m<8 ; m++){
                System.out.print(" " + (m+1) + "  ");
            }
            System.out.print("¥ny ");
            for(int m=0 ; m<8 ; m++){
                System.out.print("____");
            }
            System.out.println();
            for(int n=0 ; n<8 ; n++){
                System.out.print(n+1);
                for(int m=0 ; m<8 ; m++){
                    if(ishi[n][m]==1){
                        System.out.print("｜○");
                        siro++;
                    }else if(ishi[n][m]==2){
                        System.out.print("｜●");
                        kuro++;
                    }else{
                        System.out.print("｜  ");
                    }
                }
                System.out.print("｜¥n ");
                for(int m=0 ; m<8 ; m++){
                    System.out.print("｜__");
                }
                System.out.println("｜");
            }
            if(siro==0 || kuro==0 || (siro+kuro==64)){  //どちらかが０ or 全てのマスが埋まったら終了
                winner(siro, kuro);
                break;
            }
            //★★★次の手★★★
            int x,y;
            do{
                System.out.print("x:");
                x = in.nextInt()-1;
                System.out.print("y(9で終了):");
                y = in.nextInt()-1;             //↓x,yが8なら要素外なので
            }while(x<0 || x>8 || y<0 || y>8 || ((x!=8 && y!=8) && ishi[y][x]!=0));  //すでに石があれば置けない
            if(x==8 || y==8){   //終了フラグ
                end = 8;        break;
            }
            //★★★奪取★★★
            byte self = yx;
            place = false;                  //コレないと、一度取ったらどこでも置ける
            place2 = false;
            getIshi(ishi,x,y,self,true);
            if(place2 == false){                //そのマスでは敵石が取れない場合
                for(int m=0 ; m<8 ; m++){   //全マスをcheck!
                    for(int n=0 ; n<8 ; n++){
                        if(ishi[m][n]!=0){  //石がまだない場所のみcheck!
                            continue;
                        }
                        getIshi(ishi,n,m,self,false);
                        if(place){  //取れる石がどこかにあるなら、取れる石がないマスには置けない
                            System.out.println("取れる石がまだあります！");
                            break;
                        }
                    }
                    if(place){
                        break;
                    }
                }
                if(place != true){  //取れる石がどこにもないなら置く
                    ishi[y][x] = yx;
```

```
                    yx = (yx==1) ? (byte)2 : (byte)1;
/*  ●確認用●  */  //  System.out.println("mainのyx:" + yx);
/*  ●確認用●  */  //  System.out.println("mainのself:" + self);
                }
            }
        }while(end<8);
    }
    static void winner(byte siro, byte kuro){
        System.out.println("○:" + siro + " vs." + "●:" + kuro);
        if(siro>kuro){
            System.out.print("○ ＷＩＮ！");
        }else if(siro<kuro){
            System.out.print("● ＷＩＮ！");
        }else{
            System.out.print("引き分け！");
        }
    }
    static void getIshi(byte[][] ishi, int x, int y, byte self, boolean check2) {
        get(ishi,x,y,self,-1, 1, check2);   //左下
        get(ishi,x,y,self,-1, -1, check2);  //左上
        get(ishi,x,y,self,1, 1, check2);    //右下
        get(ishi,x,y,self,1, -1, check2);   //右上

        get(ishi,x,y,self,0, -1, check2);   //上
        get(ishi,x,y,self,0, 1, check2);    //下
        get(ishi,x,y,self,-1, 0, check2);   //左
        get(ishi,x,y,self,1, 0, check2);    //右
        if(check2 && sirokuro){ //敵石を取ったら
            yx = (yx==1) ? (byte)2 : (byte)1;
            sirokuro = false;
/*  ●確認用●  */  //  System.out.println("get()のself:" + self + " yx:" + yx);
        }
    }
    static void get(byte[][] ishi, int x, int y, byte self, int a, int b, boolean check2) {
        int x1 = x+a;
        int y1 = y+b;
/*  ●確認用●  */  //  System.out.println("x:" +(x1+1-a) + " y:" +(y1+1-b)+ "に置けば");
        if(x1>=0 && y1>=0 && x1<8 && y1<8){
            byte teki = (self==1) ? (byte)2 : (byte)1;
            if(ishi[y1][x1] == teki){   //敵石と隣接してれば次のマスをcheck!
/*  ●確認用●  */  //  System.out.println("x:" +(x1+1) + " y:" +(y1+1)+ "が隣接の敵");
                x1+=a;
                y1+=b;
                while(x1>=0 && y1>=0 && x1<8 && y1<8){
                    if(ishi[y1][x1] == 0){  //石がなければ処理なし
                        break;
                    }else if(ishi[y1][x1] != teki){
                        place=true; //敵石の先に自分の石があれば
/*  ●確認用●  */  //  System.out.println("x:" +(x1+1) + " y:" +(y1+1)+ "ではさめる");
/*  ●確認用●  */  //  System.out.println("ishi[y1][x1]:" +ishi[y1+1][x1+1]);
                        break;
                    }
                    x1+=a;
                    y1+=b;
                }
                if(check2 && place){    //敵石が取れる状態なら
                    x1-=a;
                    y1-=b;
                    while(ishi[y1][x1] == teki){    //敵石を裏返す
                        ishi[y1][x1] = self;
                        x1-=a;
                        y1-=b;
                    }
                    ishi[y][x] = yx;    //今回置いた場所
                    sirokuro = true;    //白黒交代するかどうか
                    place2=true;
                    place=false;        //裏返したら条件を戻す　ないと以後のget()呼出で裏返してしまう
                }
            }
        }
    }
}
//  ==========================================================
//   【Version 情報】
//      Version 1.0     一通りルールに対応できる     そのうち？
//      Version 0.52    条件式の記述方法を変更       Sep.01.2009
//      Version 0.51    石が0個になったら終了        Sep.01.2009
//      Version 0.42    条件式の記述方法を変更       Aug.25.2009
//      Version 0.41    winner表示                   Aug.24.2009
//      Version 0.3     全マス埋まったらゲーム終了   Aug.21.2009
//      Version 0.2     先攻○で、交代で石を置く     Aug.20.2009
//      Version 0.1     とりあえずオセロらしきもの   Aug.19.2009
//  ==========================================================
```