

```
//
//=====
【名 称】オセロゲーム Othello Version 0.3
【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 : Aug. 21. 2009(Fri)
//=====
```

```
import java.util.Scanner;
public class Othello03{
    static byte yx = 1; //○先攻 1:○ 2:●
    static boolean place; //booleanの初期値 : false
    static boolean place2;
    static boolean sirokuro;
    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 end2=0; //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("\n y:");
            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(" | ○");
                        end2++;
                    }else if(ishi[n][m]==2){
                        System.out.print(" | ●");
                        end2++;
                    }else{
                        System.out.print(" |   ");
                    }
                }
                System.out.print(" | \n");
                for(int m=0 ; m<8 ; m++){
                    System.out.print(" | ____");
                }
                System.out.println(" | ");
            }
            if(end2==64){
                break; //全てのマスが埋まつたら終了
            }
            //★★★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++){
                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);
        }
    }
}
```

```

http://w7.orotii.com/~koyuki/
if(place == true) { //取れる石がどこにあるなら、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 == true){
    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;
}
}
}while(end<8);
}

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;
    }
}

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;
    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!
            x1+=a;
            y1+=b;
            while(x1>=0 && y1>=0 && x1<=7 && y1<=7) {
                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
                    break;
                }
                x1+=a;
                y1+=b;
            }
            if(place==true && check2==true){ //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 0.3 全マス埋まつたらゲーム終了 Aug. 21. 2009
// Version 0.2 先攻〇で、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
// =====

```