public class Board{

private int[][] ban; //○×ゲームの盤の状態のインスタンス変数

private int turn; //次の手を示すインスタンス変数

public Board(){ //コンストラクタ

//banを初期化

ban = new int[3][3];

int i,j;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

ban[i][j]=0;

}

}

//turnを初期化

turn=1;

}

//○か×を置くメソッド

public int put(int x, int y){

int i,j;

//指定箇所がすでに置かれている場合か、勝敗が決まっている場合

if(ban[x][y]!=0 || turn==-1)

return -1;

//指定した箇所に現在のターンの値を入れる

ban[x][y]=turn;

//ターンを変える

if(turn==1)

turn=2;

else if(turn==2)

turn=1;

//横にそろって勝敗が決まる場合

for(i=0;i<3;i++){

if(ban[i][0]==ban[i][1] && ban[i][1]==ban[i][2]){

if(ban[i][0]==1){ //○が勝った場合

turn=-1;

return 1;

}

if(ban[i][0]==2){ //×が勝った場合

turn=-1;

return 2;

}

}

}

//縦にそろって勝敗が決まる場合

for(i=0;i<3;i++){

if(ban[0][i]==ban[1][i] && ban[1][i]==ban[2][i]){

if(ban[0][i]==1){ //○が勝った場合

turn=-1;

return 1;

}

if(ban[0][i]==2){ //×が勝った場合

turn=-1;

return 2;

}

}

}

//右下がりのななめにそろって勝敗が決まる場合

if(ban[0][0]==ban[1][1] && ban[1][1]==ban[2][2]){

if(ban[0][0]==1){ //○が勝った場合

turn=-1;

return 1;

}

if(ban[0][0]==2){ //×が勝った場合

turn=-1;

return 2;

}

}

//右上がりのななめにそろって勝敗が決まる場合

if(ban[0][2]==ban[1][1] && ban[1][1]==ban[2][0]){

if(ban[0][2]==1){ //○が勝った場合

turn=-1;

return 1;

}

if(ban[0][2]==2){ //×が勝った場合

turn=-1;

return 2;

}

}

//勝敗がまだ決まっていない場合

for(i=0;i<3;i++){

for(j=0;j<3;j++){

if(ban[i][j]==0)

return 0;

}

}

//引き分けの場合

turn=-1;

return 3;

}

//ゲームを初期化するメソッド

public void reset(){

int i,j;

//banを初期化

for(i=0;i<3;i++){

for(j=0;j<3;j++){

ban[i][j]=0;

}

}

//turnを初期化

turn=1;

}

//盤の状態を確認するメソッド

public int get(int x, int y){

return ban[x][y]; //指定箇所の状態を返す

}

//次の手を示すメソッド

public int getTurn(){

return turn; //turnの値を返す

}

}

//各種インポート

import java.applet.Applet;

import java.awt.Label;

import java.awt.Button;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.Panel;

public class GameApplet extends Applet implements ActionListener{

private Board board; //Boardのインスタンス変数

private Button[][] button; //盤上のボタンのインスタンス変数

private Button reset; //リセットボタンのインスタンス変数

private Panel panel; //パネルのインスタンス変数

private Label message,message2; //ラベルのインスタンス変数

public void init(){

int i,j;

board = new Board(); //Boardを作る

BorderLayout border = new BorderLayout(); //ボーダーレイアウトマネージャー を作る

this.setLayout(border); //アプレットのレイアウトマネージャーをボーダーにす る

message = new Label("ゲームを開始できます"); //messageラベルを作る

this.add(message,BorderLayout.SOUTH); //南側にmessageラベルを貼る

message2= new Label("○×ゲーム!"); //message2ラベルを作る

this.add(message2,BorderLayout.NORTH); //北側にmessage2ラベルを貼る

reset = new Button("reset"); //resetボタンを作る

reset.addActionListener(this); //ボタンのアクションリスナーとしてアプレ ッ トを指定

this.add(reset,BorderLayout.EAST); //東側にresetボタンを貼る

panel = new Panel(); //パネルを作る

GridLayout grid = new GridLayout(3,3); //パネル用のグリッドレイアウトを 作 る

panel.setLayout(grid); //パネルをグリッドレイアウトにする

button = new Button[3][3]; //盤上のボタン配列を作る

for(i=0;i<3;i++){

for(j=0;j<3;j++){

button[i][j] = new Button(); //何も書かれていないボタンを作る

button[i][j].addActionListener(this); //ボタンのアクションリスナー としてアプレットを指定

panel.add(button[i][j]); //パネルにボタンを貼る

}

}

this.add(panel,BorderLayout.CENTER); //中央にパネルを貼る

}

//ボタンがクリックされた時の処理を行うメソッド

public void actionPerformed(ActionEvent e){

int i,j;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

//button[i][j]ボタンが押された場合

if(e.getSource()==button[i][j]){

switch(board.put(i,j)){ //board.put(i,j)の戻り値によって場合分 け

case -1: //戻り値が-1の場合

if(board.getTurn()==-1) //board.getTurn()の戻り値が-1の 場 合

message.setText("既に勝敗が決まっています"); //message ラベルを変える

else //board.getTurn()の戻り値が-1ではない場合

message.setText("不正な位置には置けません"); //message ラベルを変える

break;

case 1: //戻り値が1の場合

button[i][j].setLabel("○"); //button[i][j]ボタンを○に す る

message.setText("○の勝ちです"); //messageラベルを変える

break;

case 2: //戻り値が2の場合

button[i][j].setLabel("×"); //button[i][j]ボタンを×に す る

message.setText("×の勝ちです"); //messageラベルを変える

break;

case 0: //戻り値が0の場合

if(board.get(i,j)==1) //board.get(i,j)の戻り値が1の場合

button[i][j].setLabel("○"); //button[i][j]を○にす る

if(board.get(i,j)==2) //board.get(i,j)の戻り値が2の場合

button[i][j].setLabel("×"); //button[i][j]を×にす る

message.setText("ターンが変わりました、置いてください"); //messageラベルを変える

break;

case 3: //戻り値が3の場合

if(board.get(i,j)==1) //board.get(i,j)の戻り値が1の場合

button[i][j].setLabel("○"); //button[i][j]を○にす る

if(board.get(i,j)==2) //board.get(i,j)の戻り値が2の場合

button[i][j].setLabel("×"); //button[i][j]を×にす る

message.setText("引き分けです"); //messageラベルを変える

break;

default: //その他の場合

break;

}

}

}

}

//resetボタンが押された場合

if(e.getSource()==reset){

board.reset(); //board.resetを行う

//ボタンをリセット

for(i=0;i<3;i++){

for(j=0;j<3;j++){

button[i][j].setLabel("");

}

}

//messageラベルを変える

message.setText("ゲームがリセットされました");

}

}

}

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

246083 佐藤彩子<br>

<applet code="GameApplet.class" archive="u246083.jar" width="400" height="400">

</applet>

</body>

</html>