



Card卡牌
属性: private int value 牌的数值 private int type 牌的花色 private int x 位置坐标x private int y 位置坐标y final public static int width = 96px 纸牌宽度 (不可以设置) final public static int height = 130px 纸牌高度 (不可以设置) private boolean isFront 卡片是否正面朝上
方法: 相关getter和setter方法, width和height不可以设置 public Card(int type,int value) 自定义构造方法 public Color getColor() 通过纸牌花色获取纸牌颜色, 纸牌相连时颜色要求不同 public void draw(Graphics graphics)在相应地方绘制纸牌

CardStack卡牌堆 (使用堆实现)
属性: protected int x 位置坐标x protected int y 位置坐标y public Stack<Card> cardStack 卡牌堆
方法: public CardStack(int x, int y) 构造方法 public boolean isEmpty() 判断牌堆是否为空 public Card top() 返回牌堆第一张 public Card pop() 取走牌堆第一张 public boolean isCardStack(int pointX,int pointY) 判断鼠标在x,y处是否在牌堆上 public int countCard(int pointX,int pointY) 判断牌堆含有的牌数 public void addCard(Object card) 往牌堆添加新的牌 public void show(Graphics graphics) 牌堆第一张牌的绘制

DeckCard取牌的牌堆
方法: public DeckCard(int x, int y)实现父类的构造方法

DiscardCard用来展示所取的牌, 但是还没放的
方法: public DiscardCard(int x, int y) 实现父类的构造方法 <b>public void addCard(Card card) 改写父类addCard, 需要把牌正面朝上放入堆栈</b>

Suitpiles最后用来放的四个堆
方法: public Suitpiles(int x, int y) 实现父类的构造方法 public boolean canAdd(Card card) 能否放入 第一种情况和前面同花色且大小比它大一时可以放入 第二种情况, 空的时候只能放A

SevenPiles桌面上七个还没放的堆
属性: private int unTurnCardSum 没有翻的纸牌数量 private int cardSum 每堆的数量 final public static int separation = 30 px 正面纸牌间隔 final public static int unTurnSeparation = 10px 背面纸牌间隔
方法: public SevenPiles(int x, int y, int unTurnCardSum) 子类自己的构造方法 相关getter和setter方法, 纸牌间隔不可改变 <b>public Card pop() 弹出栈顶的纸牌,改写父类的方法, 因为子类需要减去cardSum, 移走牌时</b> <b>public int countCard(int pointX,int pointY) 获取牌堆的数量, 改写父类的方法, 因为牌堆有间隔</b> <b>public boolean isCardStack(int pointX,int pointY) 改写父类的isCardStack方法, 因为有牌的间隔</b> public boolean canAdd(Card card) 能否添加纸牌 <b>public void addCard(Object card) 往牌堆添加新的牌,改写父类方法</b> <b>public void show(Graphics graphics) 牌堆的绘制,改写父类方法</b>

CardMoving鼠标拖动的牌 (数组存放)
属性: private int x 坐标位置x private int y 坐标位置y final public static int separation = 30px 正面纸牌之间的偏移量 private CardStack cardMovingStack 存放移动中的纸牌 private ArrayList<Card> cardArrayList
方法: public CardMoving() 构造方法 相关getter和setter方法 public boolean isEmpty() 返回动态数组是否为空 public int size() 返回动态数组大小 public void addCard(Card card) 添加纸牌对象 public Card top() 返回第一个纸牌对象 public Card pop() 取走第一个纸牌对象 public ArrayList<Card> clear() 保存当前cardArrayList, 然后清空当前的cardArrayList public void show(Graphics graphics, int pointX, int pointY) 绘制list里的card的图片, 以鼠标位置为中心绘制

AboutSolitaire extends JDialog在其他中的对话框
属性: private JPanel JPanelRule 规则面板 private JTextArea JTextAreaRule private JPanel JPanelDetails 制作面板 private JTextArea JTextAreaDetails
方法: public AboutSolitaire() 构造函数

SolitairePanel extends JPanel implements MouseListener, ActionListener, MouseMotionListener
方法: private int x 鼠标位置x private int y 鼠标位置y private boolean isDragged = false 是否被拖拽
属性: public SolitairePanel() 构造方法 public void paint(Graphics graphics) 界面绘制 public void mousePressed(MouseEvent e) 鼠标按下时, 如果有纸牌, 自动居中显示拖动的纸牌 public void mouseReleased(MouseEvent e) 鼠标松开事件 public void mouseDragged(MouseEvent e) 鼠标拖拽事件 其他接口都是空方法, 没有实现

SolitaireMenu extends JMenuBar
方法: public SolitaireMenu(SolitaireFrame solitaireFrame) 构造函数

SolitaireFrame extends JFrame
属性: private SolitairePanel solitairePanel
方法: public SolitaireFrame() 构造方法

InitGame游戏初始化
//使用数组存放纸牌 static public ArrayList<Card> cards; //使用堆栈存放牌堆 static public CardStack cardStacks[]; //用来取牌的牌堆 static public DeckCard deckCard; //还没有取走的牌堆 static public DiscardCard discardCard; //还没移动的七个牌堆 static public SevenPiles sevenPiles[]; //用来放四个花色的牌堆 static public Suitpiles suitpiles[]; //移动的纸牌 static public CardMoving cardMoving; //重新开始 public static void reGame() /*判断是否能够放到最终四个牌堆suitpiles上*/ public static boolean canAddToSuitPile(int x, int y) /*从deckCard中翻出一张牌*/ public static boolean showOneCard(int x, int y) 点击discardCard牌, 开始移动 public static boolean toCardMoving(int x, int y) /*把discardCard的牌放回deckCard*/ public static void returnToDiscardToDeck() /*判断能否放到sevenPiles上*/ public static boolean canAddToSevenPiles(int x, int y) /*拿走sevenPiles的第一张后把sevenPiles下一张翻过来*/ public static void turnSevenPiles() 拖拽桌面的牌堆,整堆拖过去 public static boolean testSevenStack(int x, int y) 把cardMoving放回到原来的牌堆 public static void returnToFromPile()