21　トランプソリテア　一並べ

21.1　ホイル

トランプのカードを縦に 4 枚，横に 14 枚置けるだけの場を必要とします．まず 52 枚のカードを全部表を向けて 4 行 13 列に並べます． このとき，各行の左側に 1 枚分の空き地を空けておきます．4 枚のＡを取り出して左の空き地に上から，クラブ，ダイヤ，ハート，スペードの順に置きます．目的は，各行に同じスーツのＡからＫまでランク順に並べることです．そのためにカードを移動しますが，移動できるのは空き地にその左にあるカードの次のカードを移すことだけです．たとえば，ダイヤの８の右が空いていたら，そこにダイヤの９を移すことができます． それ以外は移せません．

※　この本来のルールだと成功率が低いので， Kの右の空き地には同じスーツの札なら何でも移せる， という特別ルールをつけると成功率が高くなります。

21.2　未完成プログラム

下記のプログラムは，カードを並べるところまでできています。これを完成させなさい。

なお，ImageTable の Width が 885，Height が 340 になるように Form のサイズを調整してください。

メソッドはアルファベット順に並んでいます。

```
unit IchiNarabeU;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,TrumpU, ExtCtrls, StdCtrls;

type
  TRow       = 0..3;
  TCol       = 0..13;
  TColAndRow = record
                 Row: TRow;
                 Col: TCol;
               end;
  TTable     = array [TRow,TCol] of TCardNo;
  TMaisuu    = array [TRow,TCol] of 0..1;
  TDodai     = array [TRow,TCol] of TPoint;
  TBasyo     = array [TCardNo] of TColAndRow;
  TFormIchi = class(TForm)
    TimerSakusei: TTimer;
    PanelMain:    TPanel;
    ButtonClose:  TButton;
    ImageTable:   TImage;
    ButtonStart:  TButton;
    ButtonNew:    TButton;
    ButtonGiveUp: TButton;
    procedure TimerSakuseiTimer(Sender: TObject);
    procedure ButtonCloseClick(Sender: TObject);
    procedure CardMouseDown(Sender: TObject; Button: TMouseButton;
```

```
            Shift: TShiftState; X, Y: Integer);
    procedure CardMouseUp(Sender: TObject; Button: TMouseButton;
            Shift: TShiftState; X, Y: Integer);
    procedure CardMouseMove(Sender: TObject; Shift: TShiftState;
            X,Y: Integer);
    procedure ButtonStartClick(Sender: TObject);
    procedure ButtonNewClick(Sender: TObject);
    procedure ButtonGiveUpClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private 宣言 }
    Trump      : TTrump;
    Table      : TTable;
    Maisuu     : TMaisuu;
    Dodai      : TDodai;
    Basyo      : TBasyo;
    SentakuChuu: Boolean;
    XDown      : Integer;
    YDown      : Integer;
    procedure DodaiSakusei;
    procedure GameOver(Seikou : Boolean);
    function  Kansei : Boolean;
    procedure Layout;
    function  Okeru(Row : TRow; Col : TCol; CardNo : TCardNo) : Boolean;
    procedure Oku(Row : TRow; Col : TCol; CardNo : TCardNo);
    procedure PickUpAce;
    procedure Start;
    procedure Toru(Row : TRow; Col : TCol);
  public
    { Public 宣言 }
  end;

var
  FormIchi: TFormIchi;

implementation

{$R *.DFM}
(********** Method **********)
procedure TFormIchi.DodaiSakusei;
          (* 置く位置を定める *)
  var
    Col : TCol;
    Row : TRow;
  begin
    with ImageTable.Canvas do
      begin
        Brush.Color := clTeal;
        FillRect(ClipRect);
        Brush.Color := clAqua;
        for Row := 0 to 3 do
          for Col := 0 to 13 do
            with Dodai[Row,Col] do
              begin
                X := 63*Col+3;
                Y := 84*Row+4;
                FillRect(Bounds(X,Y,60,80));
              end;
```

```
      end;
  end; {DodaiSakusei}

procedure TFormIchi.GameOver(Seikou : Boolean);
          (* ゲーム終了 *)
  begin
    TimerSakusei.Enabled := True;
    if Seikou
      then ShowMessage(#13+#13+#13+'      おめでとう      '+#13+#13+#13)
      else ShowMessage(#13+#13+#13+'      残念でした      '+#13+#13+#13);
  end; {GameOveer}

function  TFormIchi.Kansei : Boolean;
          (* 完成したか *)
  var
    CardNo : TCardNo;
  begin




  end; {Kansei}

procedure TFormIchi.Layout;
          (* 開始状態に並べる *)
          (* Table を作る      *)
  var
    Col    : TCol;
    Row    : TRow;
    CardNo : TCardNo;
  begin
    for Row := 0 to 3 do
      for Col := 0 to 13 do
        Maisuu[Row,Col] := 0;
    for CardNo := 0 to 51 do
      Oku(CardNo div 13, CardNo mod 13 +1,CardNo);
  end; {Layout}

function  TFormIchi.Okeru(Row : TRow; Col : TCol; CardNo : TCardNo) : Boolean;
          (* CardNo が Row 行 Col 列に置けるか *)
  var
    LeftCardNo : TCardNo;
    Ok : Boolean;
  begin
    Ok := False;







    Okeru := Ok;
  end; {Okeru}
```

```
procedure TFormIchi.Oku(Row : TRow; Col : TCol; CardNo : TCardNo);
          (* Row 行, Col 列に CardNo 番の札を追加する *)
  begin
    Table[Row,Col] := CardNo;
    Inc(Maisuu[Row,Col]);
    Basyo[CardNo].Col := Col;
    Basyo[CardNo].Row := Row;
    with Trump.Cards[CardNo] do
      begin
        Top  := Dodai[Row,Col].Y;
        Left := Dodai[Row,Col].X;
        Visible := True;
      end;
  end; {Oku}

procedure TFormIchi.PickUpAce;
          (* エースを左端に取り出す *)
  var
    Suit : TSuit;
    CardNo : TCardNo;
  begin
    with Trump do
      begin
        for Suit := stClub to stSpade do
          begin
            CardNo := No[Suit,1];
            Toru(Basyo[CardNo].Row,Basyo[CardNo].Col);
            Oku(Ord(Suit),0,CardNo);
          end;
      end;
  end; {PickUpAce}

procedure TFormIchi.Start;
  begin
    TimerSakusei.Enabled := False;
    DodaiSakusei;
    Layout;
    PickUpAce;
    SentakuChuu := False;
  end; {Start}

procedure TFormIchi.Toru(Row : TRow; Col : TCol);
         (* Row 行, Col 列の札を取る *)
  begin
    Dec(Maisuu[Row,Col]);
  end; {Toru}

(********** Event Handler **********)
procedure TFormIchi.CardMouseDown(Sender: TObject; Button: TMouseButton;
          Shift: TShiftState; X, Y: Integer);
begin
  if not Sentakuchuu
    then with TCard(Sender) do
           begin
```

```
          end;
end; {CardMouseDown}

procedure TFormIchi.CardMouseUp(Sender: TObject; Button: TMouseButton;
          Shift: TShiftState; X, Y: Integer);
var
  Row : TRow;
  Col : TCol;
begin
  if SentakuChuu
    then with TCard(Sender) do
          begin









          end;
end; {CardMouseUp}

procedure TFormIchi.CardMouseMove(Sender: TObject; Shift: TShiftState;
          X,Y: Integer);
var
  NewLeft,NewTop : Integer;
begin
  if SentakuChuu
    then with TCard(Sender) do
          begin









          end;
end; {CardMouseMove}

procedure TFormIchi.ButtonCloseClick(Sender: TObject);
begin
  Close;
end; {ButtonCloseClick}

procedure TFormIchi.TimerSakuseiTimer(Sender: TObject);
var
  CardNo : TCardNo;
  RectSaki : TRect;
  RectMoto : TRect;
  X1,Y1 : Integer;
begin
  if Trump = NiL                    // 未作成なら作成する
    then begin                      // 最初に１回だけ実行
          Trump := TTrump.Create;
          Trump.Sakusei(Self);
          Trump.MouseHandler(CardMouseDown,CardMouseUp,CardMouseMove);
```

```
            for CardNo := 0 to 51 do
              with Trump.Cards[CardNo] do
                begin
                  Height := 80;
                  Width  := 60;
                end;
          end;
  Trump.Hide;
  X1 := Random(ClientWidth)-36;
  Y1 := Random(ClientHeight)-48;
  CardNo := Random(52);
  with  Trump.Cards[CardNo] do
    begin
      RectMoto := Canvas.ClipRect;
      RectSaki := Bounds(X1,Y1,60,80);
      ImageTable.Canvas.CopyRect(RectSaki,Canvas,RectMoto);
    end;
end; {TimerSakuseiTimer}


procedure TFormIchi.ButtonStartClick(Sender: TObject);
begin
  Start;
end; {ButtonStartClick}

procedure TFormIchi.ButtonNewClick(Sender: TObject);
begin
  Trump.Shuffle;
  Start;
end; {ButtonNewClick}

procedure TFormIchi.ButtonGiveUpClick(Sender: TObject);
begin
  GameOver(False);
end; {ButtonClick}

procedure TFormIchi.FormCreate(Sender: TObject);
begin
  DodaiSakusei;
end; {FormCreate}

end.
```