

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ, УПРАВЛЕНИЯ И ДИЗАЙНА
"КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ"**

**Лабораторная работа №8
по курсу "Алгоритмы и Структуры данных"
на тему: "Раскраска вершин графов"**

Выполнил студент группы ВТ-21:

Потеренко А.Г.

Проверил преподаватель:

Мингалиев Р.Ш.

Порядок работы.

1. Анализ индивидуального задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке Паскаль.
3. Разработка контрольных примеров.
4. Отладка программ.
5. Составление отчета.

Содержание отчета.

	Стр.
1. Текст постановки задачи.....	3
2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.....	3
3. Проблематика задачи.....	3-4
4. Алгоритм.....	4
5. Описание и обоснование контрольных примеров.....	5
6. Текст программы с комментариями (в виде приложения).....	6-10

1. Текст постановки задачи.

Написать программу раскраски вершин графа и рассмотреть проблему раскраски вершин графа 4-ми и 5-ю красками.

2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.

В данной программе граф представляется матрицей смежности и имеет максимальную размерность $n=10$. Граф называется вершинно- K -раскрашенным, если его вершины можно раскрасить K -красками так, чтобы никакие две вершины не имели бы одинаковый цвет. Минимальное число K , при котором граф является вершинно- K -раскрашенным называется хроматическим числом графа.

3. Проблематика задачи.

Исторически понятие хроматического числа возникло с проблемой четырех красок. Проблема возникла в математике в середине 19 века. Первоначально вопрос формулировался так: сколько нужно красок для раскраски любой географической карты, при которой соседние страны раскрашены в разные цвета? Под географической картой понимается разбиение плоскости на конечное число связных областей, стран, границы которых состоят из замкнутых непрерывных линий без самопересечений, а соседними являются страны, имеющие общую границу ненулевой длины. Довольно очевидно, что четырех красок недостаточно и вопрос формулировался обычно в более конкретном виде: достаточно ли четырех красок для раскраски любой географической карты? Это и есть проблема четырех красок. Положительный ответ на вопрос называется гипотезой четырех красок.

Проблема раскраски географических карт сводится к проблеме (правильной) раскраски плоских графов. Проиллюстрируем это сведение картой, изображенной на рис. 1.

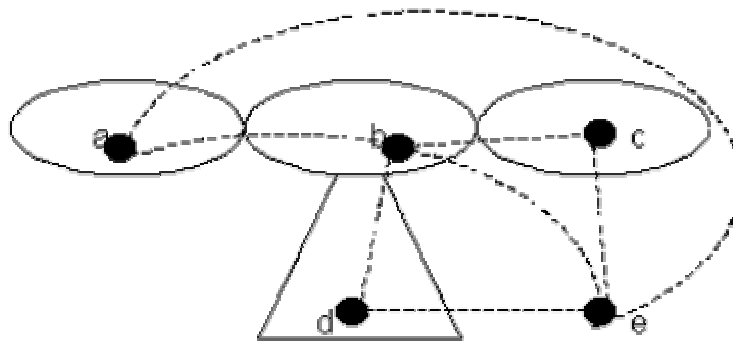


Рис. 1.

На рисунке изображена карта, имеющая пять стран (внешняя область – тоже страны). Внутри каждой страны зафиксируем точку, точки соединим ребром, если страны имеют общую границу. (На рис.1. ребра проведены пунктирными линиями). Ребра при этом можно провести так, чтобы они не пересекались, т.е. чтобы полученный граф был плоским. Ясно, что раскраска карты определяет правильную раскраску графа и наоборот. Проблема четырех красок можно теперь сформулировать так: достаточно ли четырех красок для правильной раскраски плоского графа?

Эта проблема вызвала большой интерес в математике. Есть свидетельства, что ей занимались известные математики Мебиус и де Морган. В 1880 году А. Компе опубликовал положительное решение проблемы четырех красок. Однако в 1890 году Р. Хивуд обнаружил ошибку в этом доказательстве. Одновременно он показал, что пяти красок достаточно для раскраски любого плоского графа. После этого появлялось довольно много «доказательств» гипотезы четырех красок и «контрпримеров» к ней, в которых обнаруживались ошибки. В 1969 году Х. Хели свел проблему четырех красок к исследованию множества S так называемых конфигураций. Множество S является конечным. Но довольно большим (порядка нескольких тысяч). Несколькими годами позже, в 1976 году математикам К. Аппелю и В. Хейкену удалось показать, что все конфигурации из множества S можно правильно раскрасить в четыре цвета. В возникающем при этом переборе существенно использовался компьютер. Такое решение проблемы четырех красок долгое время не признавалось многими математиками, поскольку его сложно повторить. Однако

сейчас практически общепризнано, что К. Аппелем и В. Хейкеном доказана гипотеза четырех красок.

4. Алгоритм решения данной задачи.

В данной задаче мы обходим граф в глубину, используя нерекурсивный алгоритм.

НАЧАЛО ПРОГРАММЫ

ПОВТОРИТЬ

Сформировать граф случайным образом

ПОКА сформированный граф не лес

Окрасили первую вершину

Берем первый цвет

ПОВТОРИТЬ

ЕСЛИ данная вершина, окрашена в цвет F, не имеет смежных вершин, окрашенных в цвет F

ТО окрашиваем в этот цвет

ИНАЧЕ изменяем цвет

ПОКА не окрасим вершину в нужный цвет

Вывести на экран окрашенный граф

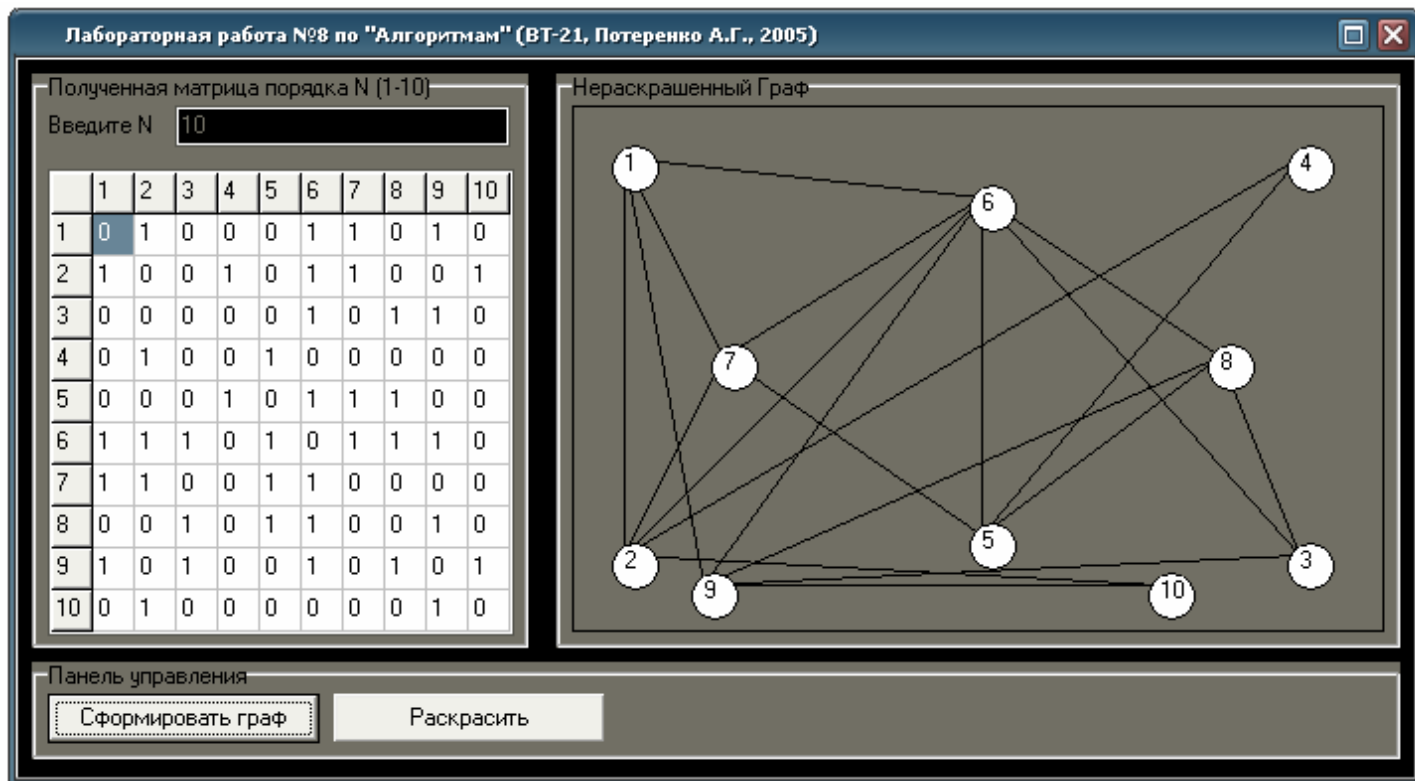
КОНЕЦ ПРОГРАММЫ

5. Описание и обоснование контрольных примеров.

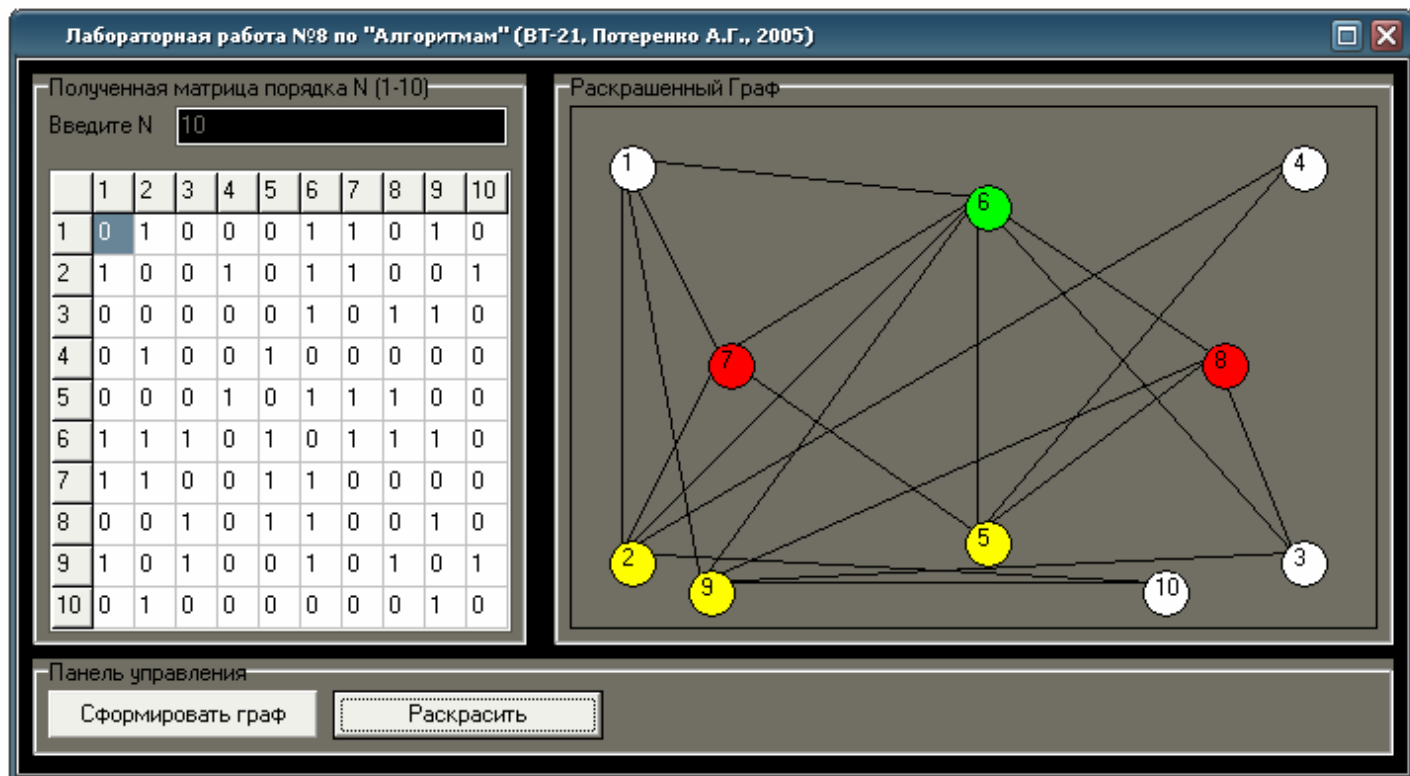
Тестирование программы заключается в том, чтобы проверить ее на "Защиту от пользователя" и ее работоспособности.

- Защита от пользователя поставлена.
- Работоспособность программы проверялась графически.
- Так как граф строится путем задания его ребер случайным образом, то в самой программе проверяется граф на лес, то есть если в результате формирования графа получился лес, то мы продолжаем формирование графа (граф в виде леса не имеет смысла в задаче).

До выполнения раскраски графа:



После выполнения раскраски графов:



6. Текст программы с комментариями (в виде приложения).

Исходный текст программы:

```
program Project1;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.Run;
end.
```

Исходный текст первого модуля программы:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Grids, UNIT2;

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    SG: TStringGrid;
    GroupBox3: TGroupBox;
    Im: TImage;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Edit1: TEdit;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure RASKRAS;
    procedure KORD;
    procedure OCHISTKA_DISPLEI;

  private
    { Private declarations }
  public
    { Public declarations }
  end;

const n=10;
      n1=20;
      n2=20;
      c=24;
      m=TColor(clwhite);

type POLIK = record
      x:word;
      y:word;
      cv:TColor
    end;

var
  mas:array[1..n,1..n] of integer;
  master:array[1..n] of POLIK;
  cvet:array [1..n] of TColor
=(clWhite,clYellow,clLime,clRed,clFuchsia,clTeal,clMaroon,clOlive,clNavy,clInactiveBorder);
```

```

Stek:array[1..1000] of integer;
G:array[1..n] of integer;
Gcv:array[1..n] of integer;
flag,flagglub,FLAG_OK_GRAF_ZAPOLNEN:boolean;
i,j,q,A,k,buf,p:integer;
Form1: TForm1;

implementation

{$R *.dfm}
//-----Загружаем граф-----
//-----
procedure TForm1.Button1Click(Sender: TObject);
label EXI;
begin
  randomize;
  OCHISTKA_DISPLEI;
  try
    q:=StrToInt(Edit1.Text);
    if (q>=1)and(q<=10) then
      begin
        //-----Проверка на количество вершин-----
        //-----
        flag:=false;
        repeat
          //-----Заполняем граф-----
          for j:=1 to q do for i:=j to q do if i<>j then
            begin
              mas[i,j]:=random(2);
              mas[j,i]:=mas[i,j];
            end
            else mas[i,j]:=0;
          //-----
          for i:=1 to n do begin G[i]:=0; end; //Очищаем очередь
          A:=1;
          j:=1;
          flagglub:=false;
          REPEAT
            G[A]:=1;
            for i:=1 to q do if (i<>A)and(mas[A,i]=1)and(G[i]<>1) then
              begin
                Stek[j]:=A;
                j:=j+1;
                A:=i;
                goto EXI;
              end;
            j:=j-1;
            A:=Stek[j];
          EXI:
            if j=0 then flagglub:=true;
          UNTIL flagglub=true;
          //----Проверка, чтобы граф не был лесом-----
          k:=0;
          for i:=1 to q do if G[i]<>0 then k:=k+1;
          if k=q then flag:=true;
        until flag=true;
        //-----
        for j:=1 to q do for i:=1 to q do SG.Cells[i,j]:=VarToStr(mas[i,j]);
        KORD;
        GroupBox2.Caption:='Нераскрашенный Граф';
        RASKRAS;
        FLAG_OK_GRAF_ZAPOLNEN:=true;
        //-----
        //-----
        //-----
      end
      else
        begin
          Form2.Left:=364;
          Form2.Top:=288;
          Form2.ShowModal;
        end
      except

```

```

Form2.Left:=364;
Form2.Top:=288;
Form2.ShowModal;

end;
end;
//-----
//                                     Раскраска графов
//-----
procedure TForm1.Button2Click(Sender: TObject);
label EXIT;
begin
  if FLAG_OK_GRAF_ZAPOLNEN=true then
    begin
      FLAG_OK_GRAF_ZAPOLNEN:=false;
      q:=StrToInt(Edit1.Text);
      for i:=1 to n do begin G[i]:=0;Gcv[i]:=0;end;
      A:=1;
      Gcv[A]:=1;
      j:=1;
      flagglub:=false;
      REPEAT
        G[A]:=1;
        for i:=1 to q do if (i<>A)and(mas[A,i]=1)and(G[i]<>1) then
          begin
            Stek[j]:=A; //Записываем вершину в стек
            j:=j+1;
            A:=i;
            //-----Проверка на раскрас-----
            if Gcv[A]=0 then
              begin
                buf:=0;
                for i:=1 to q do
                  if (mas[A,i]=1)and(A<>i) then buf:=buf+1;
                k:=1;
                repeat
                  p:=buf;
                  for i:=1 to q do
                    if (mas[A,i]=1)and(i<>A)and(k<>Gcv[i]) then p:=p-1;
                  if p=0 then Gcv[A]:=k else k:=k+1;
                until p=0;
              end;
              goto EXIT;
            end;
            j:=j-1;
            A:=Stek[j];
            EXIT: if j=0 then flagglub:=true;
          UNTIL flagglub=true;
        //-----
        for i:=1 to q do master[i].cv:=cvet[Gcv[i]]; //Заполняем поля цветами
        GroupBox2.Caption:='Раскрашенный Граф';
        RASKRAS;
      end;
    end;
  //-----
  //                                     Выводит сам граф на экран
  //-----
  procedure TForm1.RASKRAS;
  begin
    q:=StrToInt(Edit1.Text);
    for j:=1 to q do
      for i:=j to q do
        if (i<>j)and(mas[i,j]=1) then
          begin
            Im.Canvas.MoveTo(n1+master[i].x+6,n2+master[i].y+6);
            Im.Canvas.LineTo(n1+master[j].x+6,n2+master[j].y+6);
          end;
      for i:=1 to q do
        begin
          Im.Canvas.Brush.Color:=master[i].cv;
          Im.Canvas.Ellipse(n1+master[i].x,n2+master[i].y,n1+c+master[i].x,n2+c+master[i].y);
          Im.Canvas.TextOut(n1+master[i].x+6,n2+master[i].y+2,VarToStr(i));
        end;
      end;
  end;
  //-----
  //                                     Заполнить координатами

```



```

//-----
procedure TForm1.KORD;
begin
  master[1].x:=0;
  master[1].y:=0;
  master[1].cv:=clWhite;
  master[2].x:=0;
  master[2].y:=200;
  master[2].cv:=clWhite;
  master[3].x:=340;
  master[3].y:=200;
  master[3].cv:=clWhite;
  master[4].x:=340;
  master[4].y:=0;
  master[4].cv:=clWhite;
  master[5].x:=180;
  master[5].y:=190;
  master[5].cv:=clWhite;
  master[6].x:=180;
  master[6].y:=20;
  master[6].cv:=clWhite;
  master[7].x:=50;
  master[7].y:=100;
  master[7].cv:=clWhite;
  master[8].x:=300;
  master[8].y:=100;
  master[8].cv:=clWhite;
  master[9].x:=40;
  master[9].y:=215;
  master[9].cv:=clWhite;
  master[10].x:=270;
  master[10].y:=215;
  master[10].cv:=clWhite;
end;
//-----
//                                     Инициализация
//-----
procedure TForm1.FormCreate(Sender: TObject);
begin
  Form1.Left:=199;
  Form1.Top:=176;
  for i:=1 to n do
    begin
      SG.Cells[i,0]:=VarToStr(i);
      SG.Cells[0,i]:=VarToStr(i);
    end;
  Im.Canvas.Brush.Color:=cl3DDkShadow;
  Im.Canvas.Rectangle(0,0,409,265);
  FLAG_OK_GRAF_ZAPOLNEN:=false;
end;
//-----
//                                     Очищаем канву
//-----
procedure TForm1.OCHISTKA_DISPLEI;
begin
  Im.Canvas.Brush.Color:=cl3DDkShadow;
  Im.Canvas.Rectangle(0,0,409,265);
  for j:=1 to n do
    for i:=1 to n do
      SG.Cells[i,j]:=VarToStr('');
    end;
  for j:=1 to n do
    for i:=1 to n do
      mas[i,j]:=0;
    end;
  end;
end.

```

Исходный текст второго модуля программы:

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Buttons;

type
  TForm2 = class(TForm)
    BitBtn1: TBitBtn;
    Label1: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

end.
```