Л.р. №12 ст. группы 252005 Волчецкого А.М.

Написать отдельный модуль, содержащий класс, в состав которого входят пять методов решения указанной задачи и необходимые свойства класса (property).Составить программу решения задачи оптимального выбора различными методами.

unit Unit2;

interface

uses sysutils, math;

type

Tset = Set of byte;

tovary = record w,c:integer end;

mas=array [1..20] of tovary;

TMetod=class(Tobject)

N:Byte; //число элементов

CWmas:mas;

s,Optimal\_Vybor:Tset;

Wmax,Cmax:word;

Procedure Vetvi\_granici(i:byte;wt,cost:word); //Метод ветвей и границ

Procedure PP(I:byte; Full\_Weight,Full\_Cost:word); //Полный перебор

Procedure Max\_Cost(); //Метод максимальной стоимости

Procedure Min\_Weight(); //Метод минимального веса

Procedure Balanced\_Cost(); //Метод сбалансированной стоимости

Procedure Random\_Search(); //Метод случайного поиска

end;

implementation

Procedure TMetod.Vetvi\_granici;

var wt1,cost1:Word;

begin

wt1:=wt+CWmas[i].w;

if wt1<=Wmax then

begin

Include(S,i);

if i<n then Vetvi\_granici(i+1,wt1,cost)

else

if cost>Cmax then begin Cmax:=cost; Optimal\_Vybor:=S end;

Exclude(S,i);

end;

cost1:=cost-CWmas[i].c;

if cost1>Cmax then

if i<n then Vetvi\_granici(i+1,wt,cost1)

else begin Cmax:=cost1; Optimal\_Vybor:=S end;

end;

Procedure TMetod.PP;

begin

Include(s,i);

Full\_Weight:=Full\_Weight+CWmas[i].w;

Full\_Cost:=Full\_Cost+CWmas[i].c;

if i<n then PP(i+1,Full\_Weight,Full\_Cost)

else if (Full\_Weight<=Wmax) and (Full\_Cost>Cmax) then

begin Cmax:=Full\_Cost; Optimal\_Vybor:=S; end;

Exclude(s,i);

Full\_Weight:=Full\_Weight-CWmas[i].w;

Full\_Cost:=Full\_Cost-CWmas[i].c;

if i<n then PP(i+1,Full\_Weight,Full\_Cost)

else if (Full\_Weight<=Wmax) and (Full\_Cost>Cmax) then

begin Cmax:=Full\_Cost; Optimal\_Vybor:=S; end;

End;

Procedure TMetod.Max\_Cost(); //Метод максимальной стоимости

Var mc:extended; i:byte; Full\_Weight,Full\_Cost:word;

Function Imax():byte; // Элемент с максимальной стоимостью,

var i:Byte; //ещё не вошедший в Optimal\_Choice

Begin

mc:=0;

for i:=1 to n do

if not(i in Optimal\_Vybor) and (CWmas[i].c>mc) then

begin mc:=CWmas[i].c; Imax:=i end;

end;

Begin

Full\_Weight:=0; Full\_Cost:=0;

Optimal\_Vybor:=[]; i:=Imax(); Full\_Weight:=0; Full\_Cost:=0;

while Full\_Weight+CWmas[i].w<=Wmax do

begin Include(Optimal\_Vybor,i);

Full\_Weight:=Full\_Weight+CWmas[i].w; Full\_Cost:=Full\_Cost+CWmas[i].c;

i:=Imax();

end;

Cmax:=Full\_Cost;

end;

Procedure TMetod.Min\_Weight;

Var mw:extended; i,k,Full\_Weight,Full\_Cost:word;

Function Imin():word; // Элемент с минимальным весом

var i:Byte;

Begin

mw:=Wmax;

for i:=1 to n do

if not(i in Optimal\_Vybor) and (CWmas[i].w<mw) then

begin mw:=CWmas[i].w; Imin:=i end;

end;

Begin

Full\_Weight:=0; Full\_Cost:=0;

i:=Imin; k:=1;

while (k<=n) and (Full\_Weight+CWmas[i].w<=wmax) do

begin k:=k+1; Include(Optimal\_Vybor,i);

Full\_Weight:=Full\_Weight+CWmas[i].w; Full\_Cost:=Full\_Cost+CWmas[i].c;

i:=Imin;

end;

Cmax:=Full\_Cost;

end;

Procedure TMetod.Balanced\_Cost;

Var mw:extended; i,k,Full\_Weight,Full\_Cost:word;

Function I\_balanced():word;

var i:Byte;

Begin

mw:=0;

for i:=1 to n do

if not(i in Optimal\_Vybor) and (CWmas[i].c/CWmas[i].w>mw) then

begin mw:=CWmas[i].c/CWmas[i].w; I\_balanced:=i end;

end;

Begin

Full\_Weight:=0; Full\_Cost:=0;

i:=I\_balanced(); k:=1;

while (k<=n) and (Full\_Weight+CWmas[i].w<=wmax) do

begin k:=k+1; Include(Optimal\_Vybor,i);

Full\_Weight:=Full\_Weight+CWmas[i].w; Full\_Cost:=Full\_Cost+CWmas[i].c;

i:=I\_balanced(); inc(k);

end;

Cmax:=Full\_Cost;

end;

Procedure TMetod.Random\_Search;

Var i,j,k,Full\_Weight,Full\_Cost:word; Optimal\_Choice\_Bufer:Tset;

Function I\_random():word; // cлучайный элемент, ещё не использованный

var m:Byte;

Begin

repeat

m:=random(n)+1;

until not (m in Optimal\_Choice\_Bufer);

result:=m;

end;

Begin

Full\_Weight:=0; Full\_Cost:=0;

randomize;

Cmax:=0;

Optimal\_Choice\_Bufer:=[];

for j:=1 to n\*20 do

begin

i:=I\_random();

k:=1;

while (k<n) and (Full\_Weight+CWmas[i].w<=wmax) do

begin

Include(Optimal\_Choice\_Bufer,i);

Full\_Weight:=Full\_Weight+CWmas[i].w; Full\_Cost:=Full\_Cost+CWmas[i].c;

i:=I\_random();

Inc(k);

end;

if Full\_Cost>Cmax then begin Cmax:=Full\_Cost; Optimal\_Vybor:=Optimal\_Choice\_Bufer; Optimal\_Choice\_Bufer:=[]; Full\_Weight:=0; Full\_Cost:=0 end

else begin Optimal\_Choice\_Bufer:=[]; Full\_Weight:=0; Full\_Cost:=0 end

end;

end;

end.

unit Unit1;

var

Form1: TForm1;

Metod:TMetod;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

Metod:=TMetod.Create;

memo1.clear;

RadioGroup1.itemindex:=0;

StringGrid1.Cells[0,0]:='Вес'; StringGrid1.Cells[1,0]:='Цена';

StringGrid1.Cells[0,1]:='8'; StringGrid1.Cells[1,1]:='18';

StringGrid1.Cells[0,2]:='11'; StringGrid1.Cells[1,2]:='20';

StringGrid1.Cells[0,3]:='12'; StringGrid1.Cells[1,3]:='17';

StringGrid1.Cells[0,4]:='13'; StringGrid1.Cells[1,4]:='19';

StringGrid1.Cells[0,5]:='14'; StringGrid1.Cells[1,5]:='21';

Metod.n:=5;

BitBtn2.Hide;

end;

procedure TForm1.ComboBoxEx2Change(Sender: TObject); // при изменении количества элементов

begin // через ComboBox

case ComboBoxEx2.ItemIndex of

0: begin StringGrid1.RowCount:=4; StringGrid1.height:=103; StringGrid1.width:=133; Metod.N:=3; BitBtn2.Hide; end;

1: begin StringGrid1.RowCount:=6; StringGrid1.height:=153; StringGrid1.width:=133; Metod.N:=5; BitBtn2.Hide; end;

2: begin StringGrid1.RowCount:=11; StringGrid1.height:=278; StringGrid1.width:=133; Metod.N:= 10; BitBtn2.Show; end;

3: begin StringGrid1.RowCount:=21; StringGrid1.height:=315; StringGrid1.width:=151; Metod.N:=20; BitBtn2.Show; end;

end;

end;

procedure TForm1.BitBtn1Click(Sender: TObject);

var i:integer; ves,stoimost:word;

begin

Memo1.Clear; ves:=0; stoimost:=0;

With Metod do begin

try for i:=1 to N do

begin

CWmas[i].w:=strtoint(StringGrid1.Cells[0,i]);

CWmas[i].c:=strtoint(StringGrid1.Cells[1,i]);

ves:=ves+CWmas[i].w;

stoimost:=stoimost+CWmas[i].c;

end;

case ComboBoxEx1.ItemIndex of

0: Wmax:=30;

1: Wmax:=40;

2: Wmax:=50;

else Wmax:=30;

end;

if ves<Wmax then ShowMessage('Мы можем взять все предметы') else

begin

Cmax:=0; S:=[]; Optimal\_Vybor:=[];

case RadioGroup1.ItemIndex of

0: Vetvi\_granici(1,0,stoimost);

1: PP(1,0,0);

2: Max\_Cost();

3: Min\_Weight();

4: Balanced\_Cost();

5: Random\_Search();

end;

Memo1.Lines.Add('Результаты:');

Memo1.Lines.Add('');

Memo1.Lines.Add('№ Вес Цена');

for i:=1 to n do

if i in Optimal\_Vybor then Memo1.Lines.Add(IntToStr(i)+' '+FloatToStr(CWmas[i].w)+' '+FloatToStr(CWmas[i].c));

Memo1.Lines.Add('');

Memo1.Lines.Add('Общая стоимость — '+FloatToStr(Cmax)+', максимальный вес — '+FloatToStr(Wmax)+'.');

end;

except ShowMessage('Заполните таблицу целыми значениями'); end;

end;

end;

procedure TForm1.BitBtn2Click(Sender: TObject); //заполняет пустые ячейки

var i,j:Byte;

begin

Randomize;

for i:=5 to Metod.n do

for j:=0 to 1 do

StringGrid1.Cells[j,i]:=IntToStr(Random(20)+1);

end;

end.