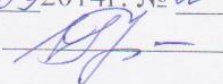



МЕТОДИЧЕСКАЯ РАЗРАБОТКА

Расчётная программа для курсового проектирования по специальности
18.02.05 «Производство тугоплавких неметаллических и силикатных
материалов и изделий»

Рассмотрен
на заседании предметной цикловой комиссии
химико-технологических дисциплин
Протокол от «04» 09 2014г. № 2
Председатель ПЦК 
Е.В. Гусакова

Составил преподаватель
 А.В. Моисеенко

ТЕХНИЧЕСКОЕ ОПИСАНИЕ ПРОГРАММЫ.

Программа создана в среде разработки приложений Delphi.

Для работы программы требуется один исполняемый файл. Никаких инсталляционных пакетов и библиотек динамической компоновки, кроме тех, что заведомо имеются в любой из операционных систем семейства Microsoft Windows, не требуется.

Проект состоит из четырёх модулей: uChoise, uTask1, uTask2, uTask3, uTask4. При запуске программы на экране появляется окно формы fChoise:

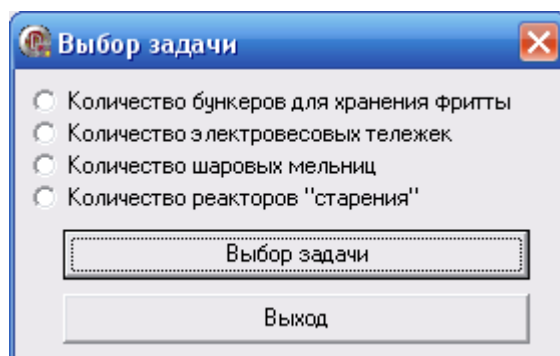


Рис. 1 – Главное окно программы

На форме расположены переключатели для выбора задачи, подтверждаемого одноимённой кнопкой.

Выберем первую задачу – расчёт количества бункеров для хранения фритты.

После ввода данных в соответствующие поля (кроме вычисляемых, защищённых от редактирования), следует нажать кнопку «Расчитать» (или клавишу «Enter»), после чего в вычисляемых полях появятся результаты расчёта.

Введём исходные данные для первой фритты (см. рис.2), нажмём кнопку «Расчитать», и получим следующий результат:

Объём бункера, м3	3
Насыпная плотность фритты, т./м3	1.5
Кратность запаса фритты на участке, т./см.	9
Количество фритты данного вида, т.	6500
Количество смен в год, см.	1080
Потребность фритты данного вида в смену, т./см.	6.0185
Количество бункеров для хранения фритты, шт.	12.0370

Рис. 2 – Окно модуля «Количество бункеров для хранения фритты»

Вернёмся в основное окно программы и выберем вторую задачу – расчёт количества электровесовых тележек (см. рис.3). Введём исходные данные и получим следующий результат:

Потребность фритты на год, т./год	10980
Производительность электровесовых тележек, т./ч	4.38
Годовой коэффициент рабочего времени	8544
Коэффициент загрузки	1
Коэффициент использования оборудования	0.93
Количество электровесовых тележек, шт.	0.3155

Рис. 3 – Окно модуля «Количество электровесовых тележек»

Закроем окно текущего модуля, и в основном окне программы выберем третью задачу – расчёт количества шаровых мельниц (см. рис.4). Введём исходные данные и получим следующий результат:

Годовой выпуск эмалевых шликеров, т./год	9750
Время размола, час	16
Годовой фонд рабочего времени, час	8544
Коэффициент загрузки	1
Коэффициент использования оборудования	0.93
Производительность мельницы, т./час	0.0913
Количество шаровых мельниц, шт.	13.4471

Рис. 4 – Окно модуля «Количество шаровых мельниц»

Вернёмся в основное окно программы и выберем четвёртую задачу – расчёт количества реакторов «старения» (см. рис.5). Введём исходные данные для первого шликера и получим следующий результат:

Годовой выпуск эмалевых шликеров, т./год	9750
Время "старения", час	12
Годовой фонд рабочего времени, час	8544
Коэффициент загрузки	1
Коэффициент использования оборудования	0.93
Производительность реакторов "старения", т./час	0.1217
Количество реакторов "старения", шт.	10.0853

Рис. 5 – Окно модуля «Количество реакторов «старения»»

Исходный код программы

Листинг модуля uChoise.pas:

```

unit uChoise;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TfChoise = class(TForm)
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    Button1: TButton;
    Button2: TButton;
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  fChoise: TfChoise;
implementation
uses uTask1, uTask2, uTask3, uTask4;
{$R *.dfm}
procedure TfChoise.RadioButton1Click(Sender: TObject);
begin
  Button1.Caption:=RadioButton1.Caption
end;
procedure TfChoise.RadioButton2Click(Sender: TObject);
begin
  Button1.Caption:=RadioButton2.Caption
end;
procedure TfChoise.RadioButton3Click(Sender: TObject);
begin
  Button1.Caption:=RadioButton3.Caption
end;
procedure TfChoise.RadioButton4Click(Sender: TObject);
begin
  Button1.Caption:=RadioButton4.Caption
end;
procedure TfChoise.Button1Click(Sender: TObject);
begin
  if RadioButton1.Checked=true then
  ftask1.ShowModal
  else if RadioButton2.Checked=true then
  ftask2.ShowModal

```

```

else if RadioButton3.Checked=true then
ftask3.ShowModal
else if RadioButton4.Checked=true then
ftask4.ShowModal
else MessageDlg('Выберите задачу',mtInformation, [mbYes], 0);
end;
procedure TfChoise.Button2Click(Sender: TObject);
begin
Application.Terminate;
end;
end.

```

После нажатия подтверждающей кнопки появляется окно формы с полями ввода данных, например, fTask1:

Исходный код модуля первой задачи uTask1.pas:

```

unit uTask1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;
type
  TfTask1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Edit6: TEdit;
    Label5: TLabel;
    Edit4: TEdit;
    Label6: TLabel;
    Edit5: TEdit;
    Button1: TButton;
    Label7: TLabel;
    Edit7: TEdit;
    BitBtn1: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit3KeyPress(Sender: TObject; var Key: Char);
    procedure Edit4KeyPress(Sender: TObject; var Key: Char);
    procedure Edit5KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
    v,k,g,ci,m:real; strc,strn:string ;
  public
    { Public declarations }
    Code: integer;
  end;
var

```

```

    fTask1: TfTask1;
implementation
{$R *.dfm}
procedure TfTask1.Button1Click(Sender: TObject);
begin
try
val(Edit1.Text,v,Code);
val(Edit2.Text,g,Code);
val(Edit3.Text,k,Code);
val(Edit4.Text,ci,Code);
val(Edit5.Text,m,code);
str(ci/m:10:4,src);
str(k*ci/(m*v*g):10:4,strn);
except
MessageDlg('Недопустимые символы, повторите ввод',mtError,[mbRetry],0);
end;
Edit6.Text:=trim(src);
Edit7.Text:=trim(strn);
end;
procedure TfTask1.BitBtn1Click(Sender: TObject);
begin
Application.Terminate;
end;
procedure TfTask1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
case key of
#44: key:=#46;
#45,#46,#48..#57,#8:exit;
else key:=#7;
end;
end;
procedure TfTask1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
case key of
#44: key:=#46;
#45,#46,#48..#57,#8:exit;
else key:=#7;
end;
end;
procedure TfTask1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin
case key of
#44: key:=#46;
#45,#46,#48..#57,#8:exit;
else key:=#7;
end;
end;
procedure TfTask1.Edit4KeyPress(Sender: TObject; var Key: Char);
begin
case key of
#44: key:=#46;
#45,#46,#48..#57,#8:exit;
else key:=#7;
end;
end;

```

```

end;
procedure TfTask1.Edit5KeyPress(Sender: TObject; var Key: Char);
begin
  case key of
    #44: key:=#46;
    #45,#46,#48..#57,#8:exit;
    else key:=#7;
  end;
end;
end;
end.

```

Исходный код модуля второй задачи uTask2.pas:

```

procedure TfTask2.Button1Click(Sender: TObject);
begin
  val(Edit1.Text,ci,Code);
  val(Edit2.Text,g,Code);
  val(Edit3.Text,t,Code);
  val(Edit4.Text,kz,Code);
  val(Edit5.Text,ki,code);
  str(ci/(t*ki*kz*g):10:4,strn);
  Edit6.Text:=trim(strn);
end;

```

Исходный код модуля третьей задачи uTask3.pas:

```

procedure TfTask3.Button1Click(Sender: TObject);
begin
  val(Edit1.Text,p,Code);
  val(Edit4.Text,t,Code);
  g:=mnz/t;
  val(Edit5.Text,t,Code);
  val(Edit6.Text,kz,code);
  val(Edit7.Text,ki,code);
  str(g:10:4,strg);
  str(p/(t*ki*kz*g):10:4,strn);
  Edit2.Text:=trim(strg);
  Edit3.Text:=trim(strn);
end;

```

Исходный код модуля четвёртой задачи uTask4.pas:

```

procedure TfTask4.Button1Click(Sender: TObject);
begin
  val(Edit1.Text,p,Code);
  val(Edit4.Text,t,Code);
  g:=mnz/t;
  val(Edit5.Text,t,Code);
  val(Edit6.Text,kz,code);
  val(Edit7.Text,ki,code);
  str(g:10:4,strg);
  str(p/(t*ki*kz*g):10:4,strn);
  Edit2.Text:=trim(strg);
  Edit3.Text:=trim(strn);
end;

```