Министерство образования и науки Республики Казахстан

Карагандинский государственный технический университет

Кафедра

***ПОЯСНИТЕЛЬНАЯ***

***ЗАПИСКА***

*к курсовому проекту*

*по дисциплине "Прикладная теория систем"*

*Тема: "Моделирование системы массового обслуживания"*

Руководитель

Студент

Караганда 2009

## Аннотация

Пояснительная записка дает полную информацию о поставленной задаче курсового проекта, необходимой теоретической части, исследуемой до разработки программы, а также здесь приведены описание программного модуля, руководство пользователя, для работы пользователя с программой, экспериментальная часть работы системы и в приложении приведен текст основного кода программы.

Содержание

Аннотация

Введение

Постановка задачи

Алгоритм решения задачи

Структура программы

Руководство пользователя

Результаты проводимых экспериментов

Заключение

Список использованных источников

Приложение

## Введение

Целью выполнения курсового проекта является создание программы, моделирующей систему массового обслуживания. По введенным параметрам программа должна позволять производить анализ работы системы. Необходимо чтобы программа была интуитивно понятна широкому кругу пользователей. Целью программы также является приобретение знаний в моделировании, как общих систем, так и систем массового обслуживания в частности. Необходимо получить навыки системного исследования реальной динамической сложной системы на основе построения ее имитационной модели.

## Постановка задачи

Основной задачей теории систем массового обслуживания является изучение режима функционирования обслуживающей системы и исследование явлений, возникающих в процессе обслуживания. Так, одной из характеристик обслуживающей системы является время пребывания требования в очереди. Очевидно, что это время можно сократить за счет увеличения количества обслуживающих устройств. Однако каждое дополнительное устройство требует определенных материальных затрат, при этом увеличивается время бездействия обслуживающего устройства из-за отсутствия требований на обслуживание, что также является негативным явлением. Следовательно, в теории возникают задачи оптимизации: каким образом достичь определенного уровня обслуживания (максимального сокращения очереди или потерь требований) при минимальных затратах, связанных с простоем обслуживающих устройств. Именно для такого чщательного анализа и необходимо программное средство моделирующее систему массового обслуживания в той или иной области и позволяющее достаточно точно проводить анализ.

В данном курсовом проекте необходимо создать программу, которая по введенным параметрам работы системы: периодичность появления покупателей, количество аппаратов обслуживания, длина очереди, время пребывания в очереди, должна смоделировать процесс работы системы массового обслуживания "Магазин". Очередь общая. Выбор аппарата обслуживания покупателем строится по принципу ближайший свободный. Также необходимо определить оптимальное число аппаратов обслуживания.

## Алгоритм решения задачи

В качестве среды программирования была выбрана визуальная среда Delphi 7, которая удовлетворяет всем требованиям по созданию визуальных приложений для операционной среды Microsoft Windows.

Генерируется поток покупателей. Каждый покупатель генерируется в случайном промежутке времени в интервале заданном в системе. Время обслуживания каждого покупателя также задается случайным образом в заданном интервале времени. Время моделирования и время работы аппаратов генерируется в минутах. Если аппараты заняты, то покупатель встает в очередь.

Когда какой-либо аппарат освобождается он выбирает покупателя из очереди и производит ее обработку.

Существует ограничение по времени ожидания в очереди.

В общем функционирование системы является надежным и быстрым. Программа фиксирует изменение системы каждую минуту времени моделирования и отражает полученые результаты.

## Структура программы

Основной исполняемый модуль программы Unit1. pas, который включает в себя все процедуры.

При нажатии на кнопку "Начать моделирование", выполняется процедура Button1Click, в которой происходит инициализация всех переменных, очистка всех необходимых полей, чтение параметров моделирования и активация таймера.

Далее запускается процедура Timer1Timer, в которой и представлен основной алгоритм работы программы.

Рассмотрим действия программы на одном шаге таймера.

Сначала программа проверяет очередь. Если очередь есть и есть свободный аппарат, то из очереди выбирается покупатель и прикрепляется к свободному аппарату. Аппараты представлены массивом app: array [1. .50] of boolean. Очередь - массивом croud: array [1. .30] of boolean.

Далее, если аппарат занят, то время обслуживания этим аппаратом клиента уменьшается на единицу. Если это время становится равным нулю, то аппарат получает статус свободного.

Затем если пришел новый клиент, т.е. переменная client, отвечающая за время появления нового клиента равна 0, то запускаются сначала процедура clientappear, а затем процедура apparat.

Процедура clientappear генерирует время появления нового покупателя.

В процедуре apparat происходит прикрепление покупателя к свободному аппарату, или же при отсутствии свободных аппаратов прикрепление к очереди, за что отвечает процедура CroudP. Длина очереди и время пребывания в ней задаются пользователем.

Далее, если очередь есть, то время пребывания покупателя в очереди уменьшается на единицу. Если это время становится равным нулю, то покупатель уходит.

Затем идет уменьшение переменной client на 1, а также приращение минут и выполняются действия, связанные с работой таймера на форме.

В конце выполняется процедура statistic, которая отвечает за вывод на экран статистических данных.

Процедура Button2Click останавливает процесс моделирования.

Таким образом, программа имеет следующие глобальные переменные и процедуры:

S: integer - всего обслужено покупателей;

FA: integer - кол-во свободных аппаратов;

Min: integer - минуты;

Hour: integer - часы;

Day: integer - дни;

Minmax: integer - максимальное число минут;

Daymax: integer - максимальное число дней;

Hourmax: integer - максимальное число часов;

Client: integer - время прихода нового клиента;

Croudcount: integer - покупателей в очереди;

cleft: integer - покупателей ушло;

app: array [1. .50] of boolean - аппараты;

apptime: array [1. .50] of byte - время обслуживания клиента;

obsl: array [1. .50] of integer - клиентов обслужено аппаратом;

appworktime: array [1. .50] of integer - время занятости аппарата;

croud: array [1. .30] of boolean - очередь;

croudtime: array [1. .30] of byte - время ожидания в очереди:

procedure Timer1Timer (Sender: TObject) - основной алгоритм работы программы;

procedure Button1Click (Sender: TObject) - запуск моделирования;

procedure FormCreate (Sender: TObject) - настройка формы при загрузке;

procedure ClientAppear (Sender: TObject) - генерирует время появления нового покупателя;

procedure Button2Click (Sender: TObject) - остановка моделирования;

procedure statistic (Sender: TObject) - показывает статистику;

procedure Apparat (Sender: TObject) - прикрепление покупателя к аппарату;

procedure TrackBar1Change (Sender: TObject);

procedure TrackBar2Change (Sender: TObject);

procedure TrackBar3Change (Sender: TObject);

procedure TrackBar5Change (Sender: TObject);

procedure TrackBar6Change (Sender: TObject);

procedure TrackBar7Change (Sender: TObject);

procedure CroudP (Sender: TObject) - прикрепление покупателя к очереди;

procedure N4Click (Sender: TObject) - выход из программы;

procedure N2Click (Sender: TObject) - запуск моделирования;

procedure N3Click (Sender: TObject) - остановка моделирования;

procedure N6Click (Sender: TObject) - показать справку;

## Руководство пользователя

Исполняемый файл программы - Project1. exe.

При запуске программы появляется форма, изображенная на рисунке 1.

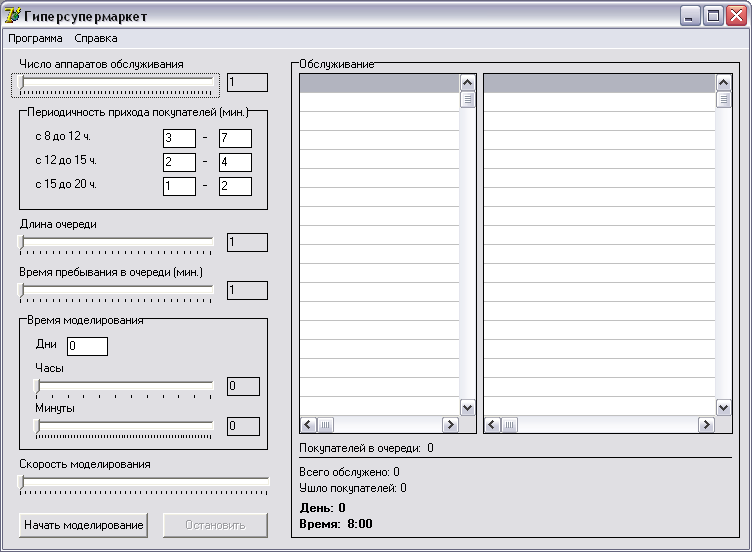


Рисунок 1. Главная форма программы

В левой части окна расположена панель для ввода основных параметров работы системы.

Число аппаратов обслуживания - это число касс, обслуживающих покупателей. Периодичность прихода покупателей - промежуток времени, в течение которого приходит новый покупатель.

Длина очереди - это максимально возможное количество покупателей в очереди. Если приходит новый покупатель, и видит, что очередь имеет максимальную длину, то он уходит.

Время пребывания в очереди - максимальное время, которое покупатель может выстоять в очереди. Если это время истекает, то покупатель уходит.

Время моделирования - это время, в течение которого будет выполняться процесс моделирования. Если время не установлено, то процесс остановится только при нажатии кнопки "Остановить".

Скорость моделирования - задает скорость течения времени. Этот параметр в отличие от вышеуказанных можно менять во время процесса моделирования.

После того, как введены параметры работы системы, для начала процесса моделирования необходимо нажать на кнопку "Начать моделирование". Вид окна в процессе моделирования изображен на рисунке 2.

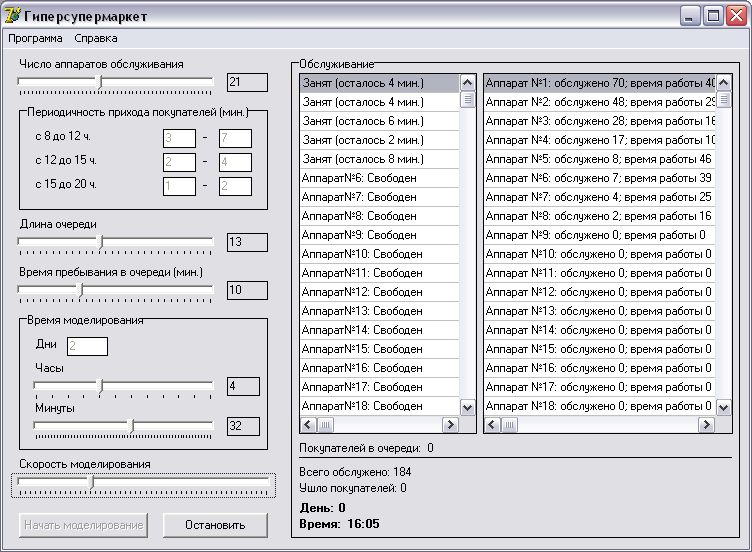


Рисунок 2. Вид программы в процессе моделирования

В нижней части формы расположена информация о времени моделирования. Так как магазин работает с 8.00 до 20.00, то время на часах не будет выходить за эти пределы.

В правой части формы представлена информация о состоянии аппаратов обслуживания (занят/свободен), а также информация о числе обслуженных аппаратом клиентов и времени работы аппарата.

Также в нижней части указана информация о количестве покупателей в очереди и о том, сколько покупателей обслужено всего и сколько ушло, не дождавшись очереди.

Процесс моделирования останавливается, когда истекает время моделирования, или, если время не указано, по нажатию кнопки "Остановить".

Управлять программой можно также с помощью меню, расположенного в верхней части формы (рисунок 3).

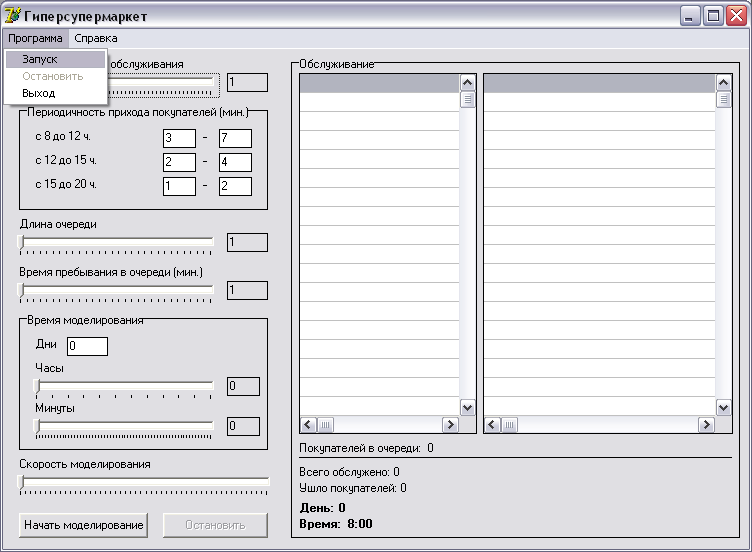


Рисунок 3. Меню программы

Здесь вы можете запустить или остановить процесс моделирования, выйти из программы, а также вызвать форму "О программе" (рисунок 4).

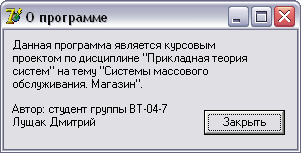


Рисунок 4. Форма "О программе"

## Результаты проводимых экспериментов

**Эксперимент №1**

Количество аппаратов 50

Длительность процесса 1 день

Пришло покупателей 424

Покупателей обслужено 424

Общая полезность системы 100%

Количество использованных аппаратов 9

Примечание: нерационально большое количество аппаратов - большая часть из них попросту простаивает.

**Эксперимент №2**

Количество аппаратов 10

Длительность процесса 1 день

Пришло покупателей 427

Покупателей обслужено 427

Общая полезность системы 100%

Количество использованных аппаратов 9

**Эксперимент №3**

Количество аппаратов 9

Длительность процесса 1 день

Пришло покупателей 424

Покупателей обслужено 424

Общая полезность системы 100%

Количество использованных аппаратов 9

Примечание: дальнейшее уменьшение количества аппаратов ведет к падению полезности системы - аппаратов на всех не хватает и покупатели уходят.

Вывод: Так как в каждом из трех экспериментов количество использованных аппаратов было 9, то оптимальное количество аппаратов, необходимых для обслуживания покупателей с периодичностью прихода с 8 до 12 - 3-7 минут, с 12 до 15 - 2-4 минуты, с 15 до 20 часов - 1-2 минуты, без создания очереди равняется 9.

## Заключение

В ходе работы была создана программа, позволяющая моделировать систему массового обслуживания “Магазин". Практически в данном курсовом проекте была реализована функциональная модель, позволяющая прослеживать изменения поведения, состояния системы.

Из проведенного исследования видно, что система является эффективной.

В результате работы была выработана способность системного рассмотрения проблем и задач, а также методов их решения.

## Список использованных источников

1. Лифшиц А.Л. Статистическое моделирование СМО, М., 1978.
2. Советов Б.А., Яковлев С.А. Моделирование систем, М: Высшая школа, 1985.
3. Гмурман В.Е. Теория вероятностей и математическая статистика, М: Высшая школа, 2001.
4. Пригодин Н.В. Системный подход в моделироании. М., 1986.

## Приложение

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ComCtrls, ExtCtrls, Menus, Grids;

type

TForm1 = class (TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

TrackBar1: TTrackBar;

TrackBar2: TTrackBar;

TrackBar3: TTrackBar;

GroupBox1: TGroupBox;

TrackBar5: TTrackBar;

TrackBar6: TTrackBar;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

GroupBox2: TGroupBox;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Button1: TButton;

Button2: TButton;

Timer1: TTimer;

GroupBox4: TGroupBox;

Label6: TLabel;

Label11: TLabel;

Label12: TLabel;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Edit10: TEdit;

Edit11: TEdit;

Edit12: TEdit;

TrackBar7: TTrackBar;

Label10: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Shape1: TShape;

Label23: TLabel;

Label4: TLabel;

Label5: TLabel;

Label24: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

StringGrid1: TStringGrid;

StringGrid2: TStringGrid;

procedure Timer1Timer (Sender: TObject);

procedure Button1Click (Sender: TObject);

procedure FormCreate (Sender: TObject);

procedure ClientAppear (Sender: TObject);

procedure Button2Click (Sender: TObject);

procedure statistic (Sender: TObject);

procedure Apparat (Sender: TObject);

procedure TrackBar1Change (Sender: TObject);

procedure TrackBar2Change (Sender: TObject);

procedure TrackBar3Change (Sender: TObject);

procedure TrackBar5Change (Sender: TObject);

procedure TrackBar6Change (Sender: TObject);

procedure TrackBar7Change (Sender: TObject);

procedure CroudP (Sender: TObject);

procedure N4Click (Sender: TObject);

procedure N2Click (Sender: TObject);

procedure N3Click (Sender: TObject);

procedure N6Click (Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

S,FA,min,hour,day,minmax,daymax,hourmax,client,croudcount,cleft: integer;

app: array [1. .50] of boolean;

apptime: array [1. .50] of byte;

obsl,appworktime: array [1. .50] of integer;

croud: array [1. .30] of boolean;

croudtime: array [1. .30] of byte;

implementation

uses Unit2;

{$R \*. dfm}

procedure TForm1. Timer1Timer (Sender: TObject);

var i,j: integer;

begin

randomize;

s: =0;

for i: =1 to trackbar1. Position do

begin

if app [i] =false then

begin

for j: =1 to trackbar2. Position do

begin

if croud [j] =true then

begin

croud [j]: =false;

app [i]: =true;

apptime [i]: =random (6) +10;

break

end;

end;

break

end;

end;

for i: =1 to trackbar1. Position do

begin

if app [i] =true then

begin

apptime [i]: =apptime [i] -1;

if apptime [i] =0 then

begin

app [i]: =false;

obsl [i]: =obsl [i] +1;

end;

end;

end;

if client=0 then

begin

clientappear (Sender);

apparat (Sender);

end;

for j: =1 to trackbar2. Position do

begin

if croud [j] =true then

begin

croudtime [j]: =croudtime [j] -1;

if croudtime [j] =0 then

begin

croud [j]: =false;

cleft: =cleft+1;

end;

end;

end;

if croud [1] =false then

begin

for j: =1 to trackbar2. Position-1 do

begin

croud [j]: =croud [j+1] ;

croudtime [j]: =croudtime [j+1] ;

end;

croud [trackbar2. Position]: =false;

end;

for i: =1 to trackbar1. Position do

begin

if app [i] =true then appworktime [i]: =appworktime [i] +1;

end;

client: =client-1;

min: =min+1;

if min>59 then

begin

min: =0;

hour: =hour+1;

end;

if hour>19 then

begin

hour: =8;

day: =day+1;

for i: =1 to trackbar1. Position do

begin

if apptime [i] >0 then obsl [i]: =obsl [i] +1;

apptime [i]: =0;

app [i]: =false;

end;

for j: =1 to trackbar2. Position do

begin

croudtime [j]: =0;

croud [j]: =false;

croudcount: =0;

end;

end;

if length (inttostr (min)) <2 then

label5. Caption: =inttostr (hour) +': 0'+inttostr (min)

else label5. Caption: =inttostr (hour) +': '+inttostr (min);

label24. Caption: ='День: ' + inttostr (day);

statistic (sender);

if (day=daymax) and (hour=hourmax) and (min=minmax)

then Button2Click (Sender);

end;

procedure TForm1. ClientAppear (Sender: TObject);

begin

randomize;

case (hour) of

8. .11: Client: =random (strtoint (edit10. Text) - strtoint (edit7. Text)) +strtoint (edit7. Text);

12. .14: Client: =random (strtoint (edit11. Text) - strtoint (edit8. Text)) +strtoint (edit8. Text);

15.19: Client: =random (strtoint (edit12. Text) - strtoint (edit9. Text)) +strtoint (edit9. Text);

end;

end;

procedure TForm1. Apparat (Sender: TObject);

var i: integer;

begin

FA: =0;

randomize;

for i: =1 to trackbar1. Position do

begin

if app [i] =false then

begin

FA: =FA+1;

app [i]: =true;

apptime [i]: =random (7) +3;

break

end;

end;

if FA=0 then CroudP (sender);

end;

procedure TForm1. CroudP (Sender: TObject);

var j,k: integer;

begin

k: =0;

for j: =1 to trackbar2. Position do

begin

if croud [j] =false then

begin

k: =k+1;

croud [j]: =true;

croudtime [j]: =trackbar3. Position;

break

end;

end;

if k=0 then cleft: =cleft+1;

end;

procedure TForm1. statistic (Sender: TObject);

var stat: string;

i,j: integer;

begin

croudcount: =0;

for i: =1 to trackbar1. Position do

begin

stringgrid1. Cells [0, i-1]: ='';

s: =s+obsl [i] ;

if app [i] =true then stat: ='Занят (осталось '+inttostr (apptime [i]) + ' мин) '

else stat: ='Аппарат№'+inttostr (i) +': Свободен';

stringgrid1. Cells [0, i-1]: =stat;

stringgrid2. Cells [0, i-1]: = ('Аппарат №'+inttostr (i) +': обслужено '+inttostr (obsl [i]) +'; время работы '+inttostr (appworktime [i]));

end;

for j: =1 to trackbar2. Position do

begin

if croud [j] =true then

croudcount: =croudcount+1;

end;

label17. Caption: =inttostr (croudcount);

label23. Caption: ='Ушло покупателей: ' + inttostr (cleft);

label18. Caption: ='Всего обслужено: ' + inttostr (S);

end;

procedure TForm1. Button1Click (Sender: TObject);

var i: integer;

begin

stringgrid1. RowCount: =trackbar1. Position;

stringgrid2. RowCount: =trackbar1. Position;

trackbar1. Enabled: =false;

trackbar2. Enabled: =false;

trackbar3. Enabled: =false;

trackbar5. Enabled: =false;

trackbar6. Enabled: =false;

edit1. Enabled: =false;

edit7. Enabled: =false;

edit8. Enabled: =false;

edit9. Enabled: =false;

edit10. Enabled: =false;

edit11. Enabled: =false;

edit12. Enabled: =false;

button1. Enabled: =false;

n2. Enabled: =false;

n3. Enabled: =true;

button2. Enabled: =true;

daymax: =strtoint (edit1. text);

hourmax: =trackbar5. Position+8;

minmax: =trackbar6. Position;

client: =0;

cleft: =0;

S: =0;

timer1. Enabled: =true;

for i: =1 to 50 do

begin

stringgrid2. Cells [0, i-1]: ='';

end;

for i: =1 to trackbar1. Position do

begin

apptime [i]: =0;

obsl [i]: =0;

appworktime [i]: =0;

end;

label5. Caption: ='8: 00';

timer1. Interval: =round (1000/ (Trackbar7. Position));

min: =0;

hour: =8;

day: =0;

for i: =1 to trackbar1. Position do

begin

app [i]: =false;

end;

for i: =1 to trackbar2. Position do

begin

croud [i]: =false;

croudtime [i]: =0;

end;

statistic (Sender);

end;

procedure TForm1. FormCreate (Sender: TObject);

begin

timer1. Enabled: =false;

button2. Enabled: =false;

n3. Enabled: =false;

end;

procedure TForm1. Button2Click (Sender: TObject);

begin

n3. Enabled: =false;

n2. Enabled: =true;

timer1. Enabled: =false;

button2. Enabled: =false;

button1. Enabled: =true;

trackbar1. Enabled: =true;

trackbar2. Enabled: =true;

trackbar3. Enabled: =true;

trackbar5. Enabled: =true;

trackbar6. Enabled: =true;

edit1. Enabled: =true;

edit7. Enabled: =true;

edit8. Enabled: =true;

edit9. Enabled: =true;

edit10. Enabled: =true;

edit11. Enabled: =true;

edit12. Enabled: =true;

end;

procedure TForm1. TrackBar1Change (Sender: TObject);

begin

edit4. Text: =inttostr (TrackBar1. position);

end;

procedure TForm1. TrackBar2Change (Sender: TObject);

begin

edit5. Text: =inttostr (TrackBar2. position);

end;

procedure TForm1. TrackBar3Change (Sender: TObject);

begin

edit6. Text: =inttostr (TrackBar3. position);

end;

procedure TForm1. TrackBar5Change (Sender: TObject);

begin

edit2. Text: =inttostr (TrackBar5. position);

end;

procedure TForm1. TrackBar6Change (Sender: TObject);

begin

edit3. Text: =inttostr (TrackBar6. position);

end;

procedure TForm1. TrackBar7Change (Sender: TObject);

begin

timer1. Interval: =round (1000/ (Trackbar7. Position));

end;

procedure TForm1. N4Click (Sender: TObject);

begin

close;

end;

procedure TForm1. N2Click (Sender: TObject);

begin

button1. Click;

end;

procedure TForm1. N3Click (Sender: TObject);

begin

button2. Click;

end;

procedure TForm1. N6Click (Sender: TObject);

begin

form2. Show;

end;

end.