

# Лабораторные работы по дисциплине Компьютерное моделирование

## Лабораторная работа №1

*Тема: Моделирование простейших логических устройств без памяти*

### Цель занятия:

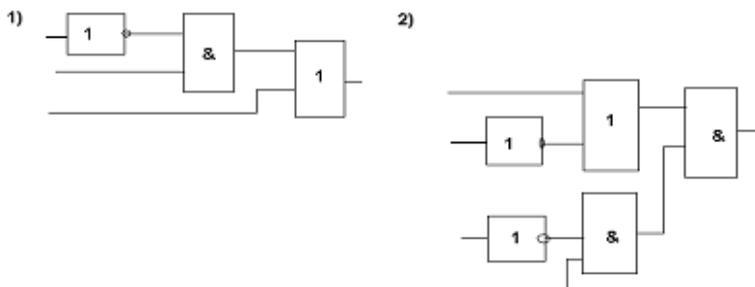
Получить практические навыки моделирования простейших логических устройств без памяти

### Ход занятия:

1. Разработка алгоритмов моделирования простейших логических устройств.
2. Реализация программ, моделирующих работу простейших логических устройств без памяти.
3. Анализ результатов моделирования.

### Вариант 1

1. Смоделируйте работу логического устройства, заданного схемой:



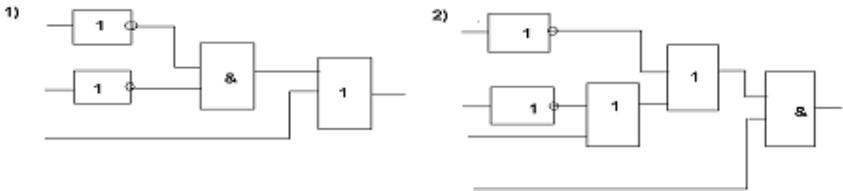
2. Используя базовые логические элементы И, ИЛИ, НЕ, смоделируйте работу устройства, реализующего следующую логику:

<i>a</i>	<i>b</i>	<i>c</i>	<i>F(a,b,c)</i>
0	0	0	<b>0</b>
0	0	1	<b>1</b>
0	1	0	<b>1</b>
0	1	1	<b>1</b>
1	0	0	<b>0</b>
1	0	1	<b>0</b>
1	1	0	<b>1</b>
1	1	1	<b>0</b>

3. Смоделируйте работу устройства из задания 2, используя базисный элемент И-НЕ.

### Вариант 2

1. Смоделируйте работу логического устройства, заданного схемой:



2. Используя базовые логические элементы И, ИЛИ, НЕ, смоделируйте работу устройства, реализующего следующую логику:

<i>a</i>	<i>b</i>	<i>c</i>	<i>F(a,b,c)</i>
0	0	0	<b>1</b>
0	0	1	<b>0</b>
0	1	0	<b>1</b>
0	1	1	<b>0</b>
1	0	0	<b>1</b>
1	0	1	<b>1</b>
1	1	0	<b>0</b>
1	1	1	<b>0</b>

3. Смоделируйте работу устройства из задания 2, используя базисный элемент И-НЕ.

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

## Лабораторная работа №2

### *Тема: Моделирование математических операций*

**Цель занятия:** Получить практические навыки моделирования математических операций

**Ход занятия:**

1. Разработка алгоритмов моделирования математических операций.
2. Реализация программы, моделирующей работу математических операций.
3. Анализ результатов моделирования.

#### Вариант № 1.

1. Смоделировать логические операции NOT AND, NOT OR с трехбайтовыми операндами 2-ой системе счисления.
2. Смоделировать арифметическую операцию сложения в 3-ой системе счисления.

#### Вариант № 2.

1. Смоделировать операцию инвертирования битов четырехбайтового операнда, заданного в 2-ой системе счисления.
2. Смоделировать арифметическую операцию сложения в 4-ой системе счисления.

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления

- (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

### Лабораторная работа № 3

**Тема: Методы исследования объектов, динамика которых описывается дифференциальными уравнениями первого порядка**

**Цель занятия:**

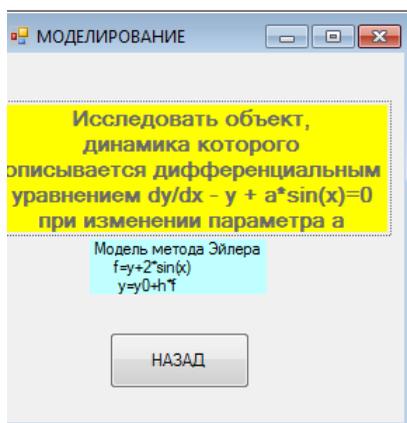
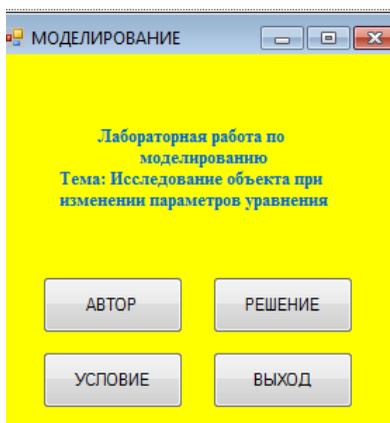
1. Получить практические навыки исследования систем (объектов), динамика которых описывается дифференциальными уравнениями 1-го порядка
2. Научиться разрабатывать алгоритм и программу с использованием языков программирования
3. Практически усвоить численные методы Эйлера и Рунге-Кутты для решения дифференциальных уравнений 1-го порядка.

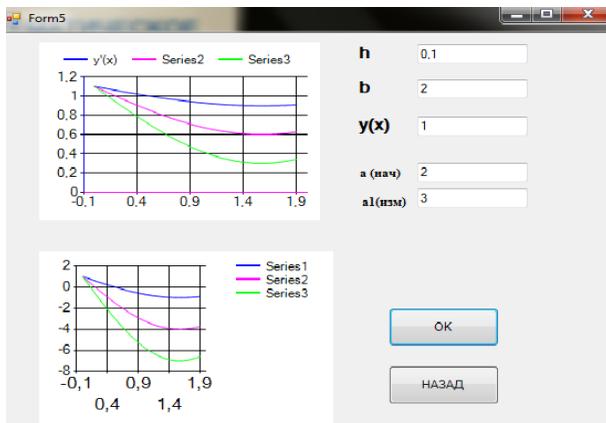
**Ход занятия:**

1. Разработка алгоритма в виде блок-схемы
2. Построение графиков кривых  $y(x)$ ,  $dy/dx$  при параметрах  $a = \text{const}$  и  $\text{var}$ ,
3. Анализ результатов исследований.

**Пример. Исследовать объект, динамика которого представляется дифференциальным уравнением**

$$\dot{y} - y - a \cdot \sin(x) = 0;$$





```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Hide();
}
```

```
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
```

```
    // int i = 1;
```

```
    // int n = 100;
```

```
    double f1,f2,x, y1;
```

```
    double h = Convert::ToDouble(textBox1->Text);
```

```
    double b = Convert::ToDouble(textBox2->Text);
```

```
    double y2 = Convert::ToDouble(textBox3->Text);
```

```
    double a = Convert::ToDouble(textBox4->Text);
```

```
    double a1 = Convert::ToDouble(textBox5->Text);
```

```
    x = 0;
```

```
    // double y1 = y2;
```

```
    int a3=a;
```

```
    //for int i=0;i<=1;i++)
```

```
    //{
```

```
    chart1->Series[0]->Points->Clear();
```

```
    chart1->Series[1]->Points->Clear();
```

```
    chart1->Series[2]->Points->Clear();
```

```
    chart2->Series[0]->Points->Clear();
```

```

chart2->Series[1]->Points->Clear();
chart2->Series[2]->Points->Clear();
//}
do
{
// for (int i=1;i<=5;i++)
// {
    a3=a;
    f1 = y2 - a3*sin(x);
    // int i=0;
    double y = y2+f1*h;
    //int i=0;
    chart1->Series[0]->Points->AddXY(x, y);
    chart2->Series[0]->Points->AddXY(x,f1);
a3=a3+a1;
    f1 = y2 - a3*sin(x);
    y = y2+f1*h;
    // f2 = y2 - 3*sin(x);
    // y = y2+f2*h;
    // int i=1;
    chart1->Series[1]->Points->AddXY(x,y);
    chart2->Series[1]->Points->AddXY(x,f1);
    a3=a3+a1;
    f1 = y2 - a3*sin(x);
    y = y2+f1*h;
    chart1->Series[2]->Points->AddXY(x,y);
    chart2->Series[2]->Points->AddXY(x,f1);
    //y=y2;
x = x+h;
}
// a=a+3;
while (x<=b);
    x=0;
}
private: System::Void chart2_Click(System::Object^ sender,
System::EventArgs^ e) {
    }

```

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

#### **Лабораторная работа № 4**

**Тема:** *Методы исследования объектов, динамика которых описывается дифференциальными уравнениями второго порядка.*

**Цель занятия:** Получить практические навыки исследования систем (объектов), динамика которых описывается дифференциальными уравнениями 2- го порядка.

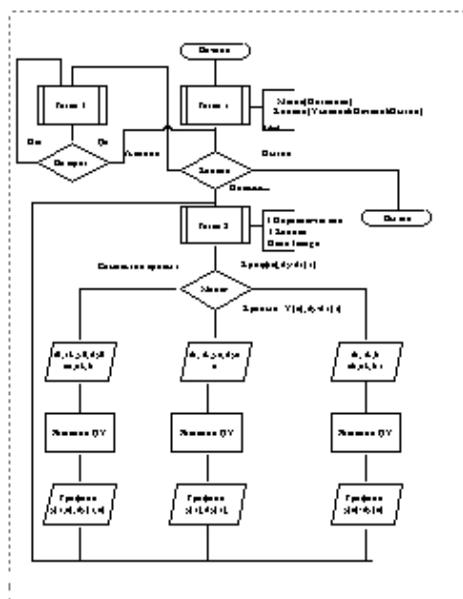
**Задачи занятия:**

1. Разработка алгоритма в виде блок - схемы
2. Построение графиков кривых  $y(x)$ ,  $dy/dx$  при параметрах  $a = \text{const}$ ,
3. Анализ результатов исследований.

Пример. Составить алгоритм и проект моделирования объекта, динамика которого описывается дифференциальным уравнением 2 порядка методом Рунге Кутта

$$y'' + a * x * y' - y = 0,4.$$

**Блок схема алгоритма**



## Решение диф. уравнений 2 порядка:

Исследовать диф. уравнение 2 порядка:  
 $y'' + 6y' + ay = 10$  методом Рунге-Кутты. В  
 процессе исследования выяснить:  
 1) зависимость функций  $y(x)$  и  $y'(x)$  при  $a = \text{const}$   
 2) зависимость функций  $y(x)$  и  $y'(x)$  при изменяющемся  $a$ .  
 3) зависимость  $y(a)$ , при  $x = \text{const}$

Исследование  $y(x)$  и  $y'(x)$  при  $a = \text{const}$

Исследование  $y(x)$  и  $y'(x)$  при изменении  $a$

Исследование  $y(a)$  и  $y'(a)$  при  $x = \text{const}$

Назад

Назад

Вид уравнения:  $y''+by'+ay=10$

Параметры:

параметр a

Левая граница

Правая граница

Шаг

Начальное значение u(0)

Начальное значение u'(0)

**Примечание**  
Отрезок, на котором рассматривается функция не должен превышать 10

Таблица значений:

x	f(x)	f'(x)
0,001	1	0
0,002	1	0,001994011982
0,003	1,0000019950093	0,0039760957126
0,004	1,0000059730937	0,0059463265252
0,005	1,0000119223944	0,0079047792786
0,006	1,0000198311274	0,0098515283602
0,007	1,0000296875831	0,0117866476889

Исследование  
Назад

Графики:

## Программа unit Unit2;

interface

var

Form2: TForm2;

implementation

uses Unit1,Unit4;

procedure TForm2.SpeedButton1Click(Sender: TObject);

begin

form4.Show;form2.Hide;

end;

procedure TForm2.SpeedButton2Click(Sender: TObject);

var

i,p :integer; // переменные, использующиеся в циклах

a : integer; // a - параметр a в нашем уравнении

h,t:real; // h - точность, t - шаг

n :integer; // число шагов

// x0,y0 :real; // начальные значения x и y

tn, tk : integer; // границы

```

x,y,kx1,kx2,kx3,kx4,ky1,ky2,ky3,ky4,dx,dy :array[1..10000] of real;
// x,y - значения x и y
// kx1-4,ky1-4 - переменные , принимающие участие в формуле
Рунге-Кутты
//dx, dy - отрезки x и y, из ,которых складывается решение
уравнения по Р-Кутту
begin
a:=strtoint(edit1.Text);tn:=strtoint(edit2.Text);
tk:=strtoint(edit3.Text);h:=strtofloat(edit4.Text);
x[1]:=strtofloat(edit5.Text);y[1]:=strtofloat(edit6.Text);
n:=trunc((tk-tn)/h);
t:=tn;
chart1.SeriesList[0].Clear;chart2.SeriesList[0].Clear;
stringgrid1.RowCount:=n;
for i:= 1 to (n) do
  begin
    kx1[i]:=h*(-a*y[i]-6*x[i]);
    kx2[i]:=h*(-a*y[i]-6*(x[i]+kx1[i]/2));
    kx3[i]:=h*(-a*y[i]-6*(x[i]+kx2[i]/2));
    kx4[i]:=h*(-a*y[i]-6*(x[i]+kx3[i]));
    dx[i]:=1/6*(kx1[i]+2*kx2[i]+2*kx3[i]+kx4[i]);
    x[i+1]:=x[i]+dx[i];
    ky1[i]:=h*x[i];
    ky2[i]:=h*(x[i]+ky1[i]/2);
    ky3[i]:=h*(x[i]+ky2[i]/2);
    ky4[i]:=h*(x[i]+ky3[i]);
    dy[i]:=1/6*(ky1[i]+2*ky2[i]+2*ky3[i]+ky4[i]);
    y[i+1]:=y[i]+dy[i];
    t:=t+h;
    stringgrid1.Cells[0,i]:=floattostr(t);
    stringgrid1.Cells[1,i]:=floattostr(y[i]);
    stringgrid1.Cells[2,i]:=floattostr(x[i]);
    chart1.SeriesList[0].AddXY(t,y[i]," ,clblue);
    chart2.SeriesList[0].AddXY(t,x[i]," ,clblue);
  end;end;
procedure TForm2.FormCreate(Sender: TObject);
begin
stringgrid1.Cells[0,0]:='x';

```

```

stringgrid1.Cells[1,0]:=f(x);
stringgrid1.Cells[2,0]:=f'(x)';end;end.
unit Unit5;
interface
    var
        Form5: TForm5;
implementation
uses Unit2, Unit4;
procedure TForm5.SpeedButton2Click(Sender: TObject);
var
    i,p :integer; // переменные, использующиеся в циклах
    a,a1,da : real; // a - параметр a в нашем уравнении
    h,t:real; // h - точность, t - шаг
    n,j :integer; // число шагов
    // x0,y0 :real; // начальные значения x и y
    tn, tk : integer; // границы
    x,y,kx1,kx2,kx3,kx4,ky1,ky2,ky3,ky4,dx,dy :array[1..10000] of
real;
    // x,y - значения x и y на соответствующем шаге
    // kx1-4,ky1-4 - переменные опинимающие участие в формуле
Рунге-Кутта
    //dx, dy - отрезки x и y из,которых складывается решение ур-ия
по Р-Кутту
begin
for j:= 0 to 5 do begin
chart1.SeriesList[j].Clear;chart2.SeriesList[j].Clear;
end;
a:=strtofloat(edit1.Text);a1:=strtofloat(edit7.Text);
da:=strtofloat(edit8.Text);
j:=0;a:=a-da;
repeat
tn:=strtoint(edit2.Text);tk:=strtoint(edit3.Text);
h:=strtofloat(edit4.Text);x[1]:=strtofloat(edit5.Text);
y[1]:=strtofloat(edit6.Text);n:=trunc((tk-tn)/h);
t:=tn;chart1.SeriesList[0].Clear;chart2.SeriesList[0].Clear;
    for i:= 1 to (n) do
        begin
            kx1[i]:=h*(-a*y[i]-6*x[i]);

```

```

kx2[i]:=h*(-a*y[i]-6*(x[i]+kx1[i]/2));
kx3[i]:=h*(-a*y[i]-6*(x[i]+kx2[i]/2));
kx4[i]:=h*(-a*y[i]-6*(x[i]+kx3[i]));
dx[i]:=1/6*(kx1[i]+2*kx2[i]+2*kx3[i]+kx4[i]);
x[i+1]:=x[i]+dx[i]; ky1[i]:=h*x[i];
ky2[i]:=h*(x[i]+ky1[i]/2); ky3[i]:=h*(x[i]+ky2[i]/2);
ky4[i]:=h*(x[i]+ky3[i]);
dy[i]:=1/6*(ky1[i]+2*ky2[i]+2*ky3[i]+ky4[i]);
y[i+1]:=y[i]+dy[i]; t:=t+h;
chart1.SeriesList[j].AddXY(t,y[i],"clblue");
chart2.SeriesList[j].AddXY(t,x[i],"clblue");
    end;
    a:=a+da; j:=j+1;
until a > a1

end;
procedure TForm5.SpeedButton1Click(Sender: TObject);
begin
form4.show;form5.Hide;
end;end.
unit Unit6;
interface
var
i,p :integer; // переменные, использующиеся в циклах
h,t:real; // h - точность, t - шаг
n :integer; // число шагов
// x0,y0 :real; // начальные значения x и y
tn, tk : integer; // границы
x,y,kx1,kx2,kx3,kx4,ky1,ky2,ky3,ky4,dx,dy :array[1..10000] of
real;
// x,y - значения x и y на соответствующем шаге
// kx1-4,ky1-4 - переменные принимающие участие в формуле
Рунге-Кутты
//dx, dy - отрезки x и y , из которых складывается решение
уравнения по Рунге-Кутту
a:real; // a - parametr
begin
tn:=strtoint(edit2.Text);tk:=strtoint(edit3.Text);

```

```

h:=strtofloat(edit4.Text);x[1]:=strtofloat(edit5.Text);
y[1]:=strtofloat(edit6.Text);n:=trunc((tk-tn)/h);
t:=tn;
chart1.SeriesList[0].Clear;chart2.SeriesList[0].Clear;
a:=4;
  for i:= 1 to (n) do
    begin
      kx1[i]:=h*(-a*y[i]-6*x[i]);
      kx2[i]:=h*(-a*y[i]-6*(x[i]+kx1[i]/2));
      kx3[i]:=h*(-a*y[i]-6*(x[i]+kx2[i]/2));
      kx4[i]:=h*(-a*y[i]-6*(x[i]+kx3[i]));
      dx[i]:=1/6*(kx1[i]+2*kx2[i]+2*kx3[i]+kx4[i]);
      x[i+1]:=x[i]+dx[i]; ky1[i]:=h*x[i];
      ky2[i]:=h*(x[i]+ky1[i]/2);
      ky3[i]:=h*(x[i]+ky2[i]/2);
      ky4[i]:=h*(x[i]+ky3[i]);
      dy[i]:=1/6*(ky1[i]+2*ky2[i]+2*ky3[i]+ky4[i]);
      y[i+1]:=y[i]+dy[i];
      t:=t+h;
      chart1.SeriesList[0].AddXY(t,y[i],"clblue");
      chart2.SeriesList[0].AddXY(t,x[i],"clblue); end;end;

```

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления
- (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

## Лабораторная работа № 5

**Тема: Моделирование физических процессов**

**Цель занятия:** Получить практические навыки исследования физических процессов средствами компьютерного моделирования.

**Задачи занятия:**

1. Разработка алгоритма в виде блок-схемы
2. Построение моделей физических процессов.
3. Анализ результатов исследований.

### **Часть 1. Свободное падение тел.**

**Задача:** Изучить движение тел с высоты  $h$  при различных начальных условиях: коэффициентов  $k_1, k_2$ ,  $r$  (радиуса шарика), плотности среды и формы тела.

**Исследуем падение шарика радиуса  $R$  с высоты  $h$ .**

1. Получить зависимости  $h=f(t)$ ,  $v=f(t)$  при начальных значениях  $r, v_0, h_0, k_1, k_2$ .
2. Получить семейство графиков  $h=f(t, r)$ ,  $v=f(t, r)$ .
3. Получить зависимость  $h=f(r)$ ,  $v=f(r)$  при  $t = \text{const}$

### **Модель объекта**

$$\frac{dh}{dt} = v,$$

$$\frac{dv}{dt} = \frac{mg - k_1 v - k_2 v^2}{m}.$$

Параметры  $k_1$  и  $k_2$  зависят от свойств среды и геометрии тела.

**Задачи исследования:**

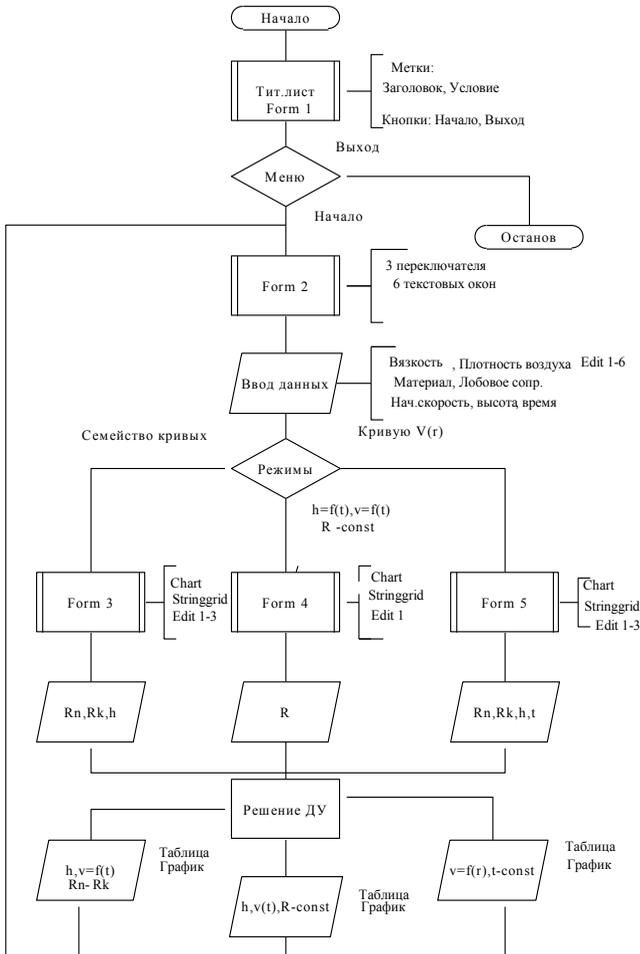
1. Получить зависимости  $h=f(t)$ ,  $v=f(t)$  при начальных значениях  $r, v_0, h_0, k_1, k_2$ .
2. Получить семейство графиков  $h=f(t, r)$ ,  $v=f(t, r)$ .
3. Получить зависимость  $h=f(r)$ ,  $v=f(r)$  при  $t = \text{const}$

**Исходные данные:**

1. Модель объекта падения,  $R$  - радиус шарика,  $h$  - высота падения,  $v_0$  - начальная скорость падения,  $k_1, k_2$  - коэффициенты внешней среды.

$$\begin{cases} \frac{dh}{dt} = v, \\ \frac{dv}{dt} = \frac{m \cdot g - k_1 \cdot v - k_2 \cdot v^2}{m}, \\ k_1 = 6 \cdot \pi \cdot \mu \cdot r, k_2 = 0.5 \cdot c \cdot S \cdot \rho, \end{cases}$$

2. - площадь сечения тела, поперечного по отношению к потоку
3.  $\mu = 0.0182 \frac{H \cdot c}{M^2}$       плотность шарика;
4.  $\rho_{\text{шарика}} = 800 \frac{K^2}{M^3}$  – плотность воздуха.
- 5.
6.  $c = 0.4$  – коэффициент лобового сопротивления шарика;



var

Form4: TForm4;

km,r,m,vo,ho,s,p,v,a,dv,da,c:double ;

h1,h2,cn,ck,dc,t,dt,y1,y2,dy1,dy2:double;

ind,k:integer;

arrt,arrv,arra,arrrh:array[1..10000] of double;

implementation

```

uses Unit1, Unit5;
//описание функции
function ff(m,km,r,c,s,p,v:double):double;
begin
  ff:=(m*9.8-6*km*Pi*r*v-0.5*c*s*p*v*v)/m;
end;
//Кнопка возврата к титульному листу
procedure TForm4.N3Click(Sender: TObject);
begin
  form5.show;
  for ind:=0 to 7 do
  begin
    form5.Chart1.SeriesList[ind].Clear;
    form5.Chart2.SeriesList[ind].Clear;
  end;
  //Входные параметры
  r:=strtofloat(edit1.Text);//радиус падающего тела
  km:=strtofloat(edit2.text);//коэффициент вязкости
  s:=strtofloat(edit3.Text);//площадь падающего тела
  p:=strtofloat(edit4.Text);//плотность среды
  ho:=strtofloat(edit5.Text);//начальная высота
  vo:=strtofloat(edit6.Text);//начальная скорость
  m:=strtofloat(edit7.Text);//масса падающего тела
  c:=0.1;//коэффициент лобового сопротивления среды
  k:=0;//номер графика
  // 1
  while c<=1.5 do
  begin
    ind:=0;
    t:=0;dt:=0.1;
    y1:=vo;
    y2:=ff(m,km,r,c,s,p,vo);
    while t<=10 do
    begin
      form5.Chart1.SeriesList[k].Add(y1,floattostr(t));
      dy1:=dt*y2;
      dy2:=dt*ff(m,km,r,c,s,p,y1);
      y1:=y1+dy1;

```

```

        y2:=y2+dy2;
        t:=t+dt;
        end;
        c:=c+0.2;
        k:=k+1;
end;
//2
k:=0;
t:=30;y1:=vo;y2:=ff(m,km,r,c,s,p,vo);
while t<=44 do
begin
    c:=0.1;
    while c<=2 do
    begin
        form5.Chart2.SeriesList[k].Add(y2,floattostr(c));
        dy1:=dt*y2;
        dy2:=dt*ff(m,km,r,c,s,p,y1);
        y1:=y1+dy1;
        y2:=y2+dy2;
        c:=c+0.1;
    end;
    t:=t+2;
    k:=k+1;
end;
end;
//Кнопка ввода данных
procedure TForm4.N1Click(Sender: TObject);
//Кнопка построения графика
procedure TForm4.N2Click(Sender: TObject);
begin
    //очистка старых графиков
    for ind:=0 to 7 do
    begin
        form4.Chart1.SeriesList[ind].Clear;
        form4.Chart2.SeriesList[ind].Clear;
    end;
    //Входные параметры
    r:=strtofloat(edit1.Text);//радиус падающего тела

```

```

km:=strtofloat(edit2.text);//коэффициент вязкости
s:=strtofloat(edit3.Text);//площадь падающего тела
p:=strtofloat(edit4.Text);//плотность среды
ho:=strtofloat(edit5.Text);//начальная высота
vo:=strtofloat(edit6.Text);//начальная скорость
m:=strtofloat(edit7.Text);//масса падающего тела
cn:=0.1;ck:=1.5;//интервал лобового сопротивления среды
dc:=(1.5-0.1)/7;//шаг изменения лобового сопротивления
k:=0;//номер графика
// цикл по лобовому сопротивлению
while cn<=ck do
begin
    v:=vo;//первая производная расстояния по времени-
начальная скорость
    a:=1; //вторая производная расстояния по времени-
ускорение
    ind:=1;//счетчик записи данных в массивы
    t:=0;//счетчик подчета времени падения тела
    h1:=ho;//высота с которой падает тело
    h2:=0; // тело лежит на земле
    //цикл времени падения тела
    while h1>=h2 do
    begin
        //запись в массивы полученных данных
        arrh[ind]:=h1;//высота
        arrv[ind]:=v;//скорость
        arrt[ind]:=t;//время
        //вывод графиков

form4.Chart1.SeriesList[k].Add(arrh[ind],floattostr(arrt[ind]));

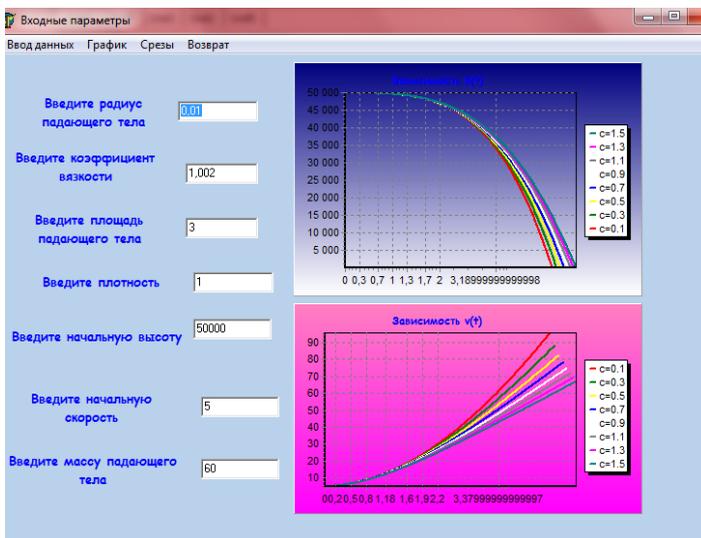
form4.Chart2.SeriesList[k].Add(arrv[ind],floattostr(arrt[ind]));
// решение дифференциального уравнения методом
Эйлера
        dv:=0.01*a;
        da:=0.01*ff(m,km,r,cn,s,p,a);
        v:=v+dv;
        a:=a+da;

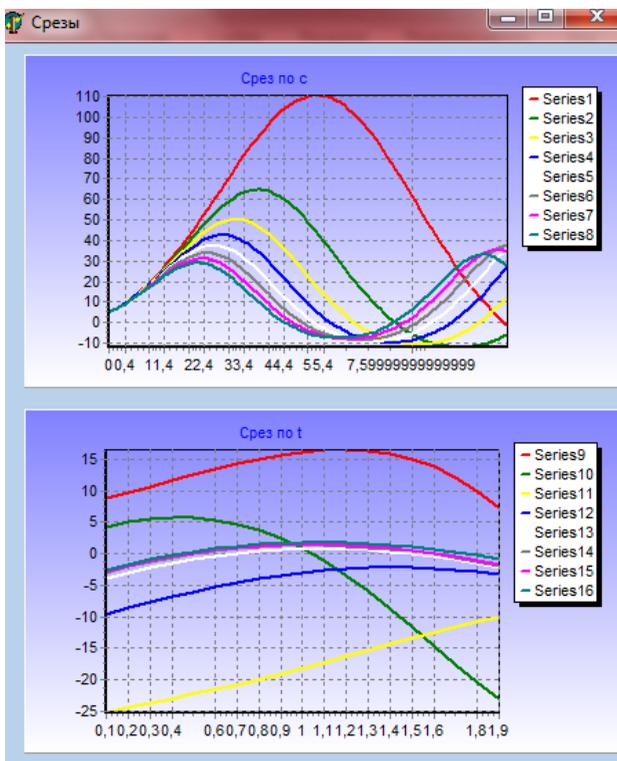
```

```

h1:=h1-arrv[ind]*arrt[ind];//изменение высоты за время t
t:=t+0.01;//изменение времени
ind:=ind+1;
end;
//переход к следующему коэффициенту лобового
сопротивления
// и соответственно к следующему графику
k:=k+1;
cn:=cn+dc;
end;
end;
procedure TForm4.N4Click(Sender: TObject);
begin
form4.Hide;
form5.hide;
form1.Show;
end;

```





## ***Часть 2. Движение исследовательского зонда вертикально вверх с летящего самолета***

**Задача. Исследовать движение исследовательского зонда вертикально вверх с летящего самолета.**

Промоделировать движение исследовательского зонда, «выстреленного» вертикально вверх с летящего над землей самолета. В верхней точке траектории над зондом раскрывается парашют, и он плавно спускается на землю.

*Входные параметры.*

Математическая модель свободного падения тела - уравнение второго закона Ньютона с учетом двух сил, действующих на тело - силы тяжести и силы сопротивления среды. Движение является одномерным; проецируя силу,

скорость и перемещение на ось, направленную вертикально вверх, получаем

$$\frac{dh}{dt} = v,$$

$$\frac{dv}{dt} = \frac{mg - k_1 v - k_2 v^2}{m}.$$

Входные параметры модели:

| начальная высота тела;  
 | начальная скорость тела;  
 | величины, определяющие коэффициенты сопротивления среды

function FmO=fun(t,y)

tv=10;

a=100;

g=9,8

mn=10000;

mk=3000;

m=mn-a\*t;

if m>mk

else

m=mk end; mm=m; if t>=tv

p=0;

m=500; else

p=1000000;

m=mam; end;

FmO=[(p\*cos(y(2))-cos(y(2))-g\*sin(y(2)))/m;

((p\*sin(y(2))+sin(y(2)))/m-

g\*cos(y(2)))/y(1);-y(1)\*cos(y(2))/10;y(1)\*sin(y(2))/10];

Y0=[1000 pi/2 0 600];

[T,Y]=ode15s(@fun,[0 1000],Y0);

plot(Y(:53),Y(:,4),'g');

hold on;

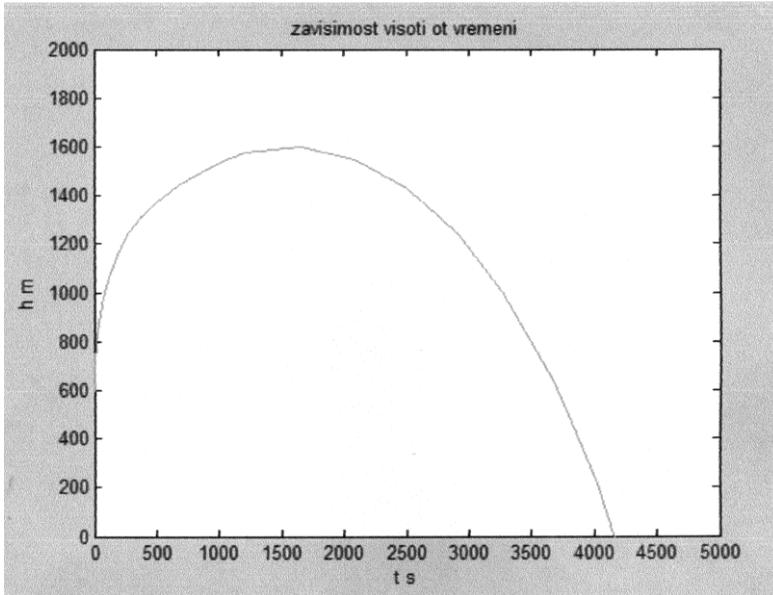
title("zavisimost visoti ot vremeni");

xlabel('fts');

ylabel('Бт');

axis([0,5000,0,2000]);

## Результат.



### Часть 3. Модели движения небесных тел

#### Задача. Построить модель движения небесных тел

По закону всемирного тяготения сила притяжения, действующая между двумя телами, пропорциональна их массам и обратно пропорциональна квадрату расстояния между ними.

$$F = -G \frac{Mm}{r^3} \vec{r}.$$

В этой формуле  $G=6,67 \cdot 10^{-11} \text{ м}^3/(\text{кг} \cdot \text{с}^2)$  - гравитационная постоянная. В данной системе координат начало координат расположено на одном теле  $M$ . Уравнения, описывающие движение тела  $m$  в указанной системе координат, имеет вид

$$G \frac{Mm}{r^3} \vec{r} = m \frac{d^2 \vec{r}}{dt^2}.$$

В проекциях на оси координат уравнение можно привести к следующей системе из 4-х дифференциальных уравнений

$$\begin{cases} \frac{dx}{dt} = v_x, \\ \frac{dv_x}{dt} = -GM \frac{x}{\sqrt{(x^2 + y^2)^3}}, \\ \frac{dy}{dt} = v_y, \\ \frac{dv_y}{dt} = -GM \frac{y}{\sqrt{(x^2 + y^2)^3}}. \end{cases}$$

Исследование выполнения второго закона Кеплера при движении различных планет.

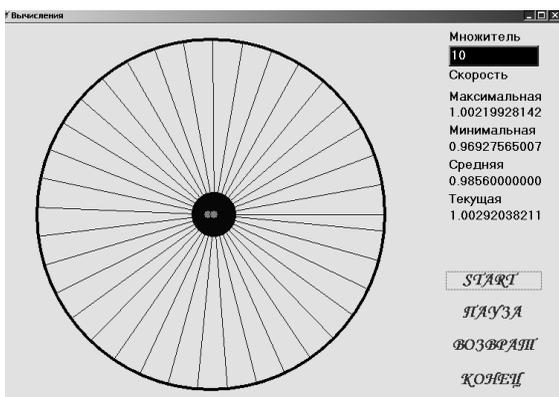
Задание значений

Планеты	Бол. полуось	Эксцентриситет	Средняя угловая скорость
Меркурий	0.387110	0.20564	4.0923
Венера	0.72334	0.0068	1.6021
Земля	1.00000	0.01670	0.9856
Марс	1.52368	0.09346	0.5240
Юпитер	5.20337	0.04845	0.0831
Сатурн	9.55447	0.05538	0.0335
Уран	19.182	0.04756	0.0117
Нептун	30.058	0.00859	0.0060
Плутон	39.46	0.248	0.0040

Большая полуось	Эксцентриситет	Средняя угловая скорость
1.00000	0.01670	0.9856

НАЧАЛО



```

unit Unit2;
var
  Form2: TForm2;
  angle,j,k,kx,ky,ra,rb,e,x,y,va,vb,vc,a,t,t2,speed:real;
  xc,yc,w,h,xf,n,cc,step:integer; flag:boolean;
implementation
uses Unit1, Unit3;
{$R *.dfm}
procedure TForm2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  form1.close;
end;
procedure TForm2.BitBtn1Click(Sender: TObject);

var rrr:trect; aa,bb,cc:integer; s:string;
begin
  // дано
  val(form1.edit1.text,ra,cc);
  if cc<>0 then ra:=1; val(form1.edit2.text,e,cc);
  if cc<>0 then e:=0.0167; val(form1.edit3.text,vc,cc);
  if cc<>0 then vc:=0.9856;
  t:=360/vc; t2:=t/2; flag:=true;
  // вычисляем
  va:=vc*sqrt((1+e)/(1-e)); vb:=vc*sqrt((1-e)/(1+e));
  a:=(va-vb)/t2; val(edit1.Text,step,cc);
  if cc<>0 then step:=1; str(va:3:11,s);
  label1.caption:=s; str(vb:3:11,s);
  label2.caption:=s; str(vc:3:11,s);
  label3.caption:=s; rb:=ra*sqrt(1-e*e);
  // графика
  w:=image1.Width; h:=image1.Height;
  xc:=w div 2; yc:=h div 2;
  kx:=ra/(xc-50); ky:=rb/(yc-50);
  if kx>ky then k:=(xc-20)/ra else k:=(yc-20)/rb;
  timer1.Enabled:=true;
  aa:=round(ra*k); bb:=round(rb*k);

```

```

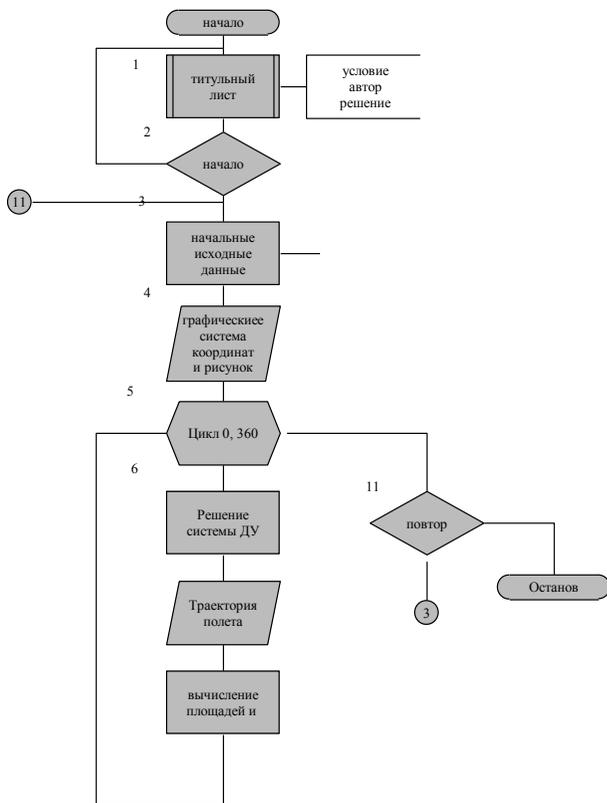
image1.Canvas.Brush.Color:=clnavy;
rrr:=rect(0,0,w,h); image1.Canvas.FillRect(rrr);
image1.Canvas.Brush.Color:=clwhite;
rrr:=rect(xc-aa,yc-bb,xc+aa,yc+bb);
image1.Canvas.Ellipse(rrr);
cc:=3; image1.Canvas.Brush.Color:=clnavy;
rrr:=rect(xc-aa+cc,yc-bb+cc,xc+aa-cc,yc+bb-cc);
image1.Canvas.Ellipse(rrr); xf:=round(ra*e*k);
cc:=30; image1.Canvas.Brush.Color:=clyellow;
rrr:=rect(xc+xf+cc,yc+cc,xc+xf-cc,yc-cc);
image1.Canvas.Ellipse(rrr);
cc:=6; image1.Canvas.Brush.Color:=clred;
rrr:=rect(xc-xf+cc,yc+cc,xc-xf-cc,yc-cc);
image1.Canvas.Ellipse(rrr);
rrr:=rect(xc+xf+cc,yc+cc,xc+xf-cc,yc-cc);
image1.Canvas.Ellipse(rrr);
angle:=0; n:=0; speed:=va; j:=pi/180;
end;
procedure TForm2.Timer1Timer(Sender: TObject);
var rrr:trect;s:string;i:integer;
begin
  image1.Canvas.MoveTo(xc+xf,yc);
  image1.Canvas.pen.Color:=clyellow;
  if angle<360 then image1.Canvas.
  LineTo (xc+round(ra*cos(angle*j)*k),
  (c-round(rb*sin(angle*j)*k)); for i:=1 to step do
  begin
    inc(n);
    if angle<180 then speed:=speed-a else speed:=speed+a;
    angle:=angle+speed; end;
  str(speed:3:11,s); label4.Caption:=s;
  if angle>=(360-a) then begin timer1.Enabled:=false; flag:=false;
  end;
  cc:=6; image1.Canvas.Brush.Color:=clred;
  rrr:=rect(xc-xf+cc,yc+cc,xc-xf-cc,yc-cc);
  image1.Canvas.Ellipse(rrr);
  rrr:=rect(xc+xf+cc,yc+cc,xc+xf-cc,yc-cc);
  image1.Canvas.Ellipse(rrr);end;

```

```

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
  if flag then timer1.enabled:=not timer1.enabled;
end;
procedure TForm2.BitBtn3Click(Sender: TObject);
begin
  form2.Hide; form1.Show;end;
procedure TForm2.BitBtn4Click(Sender: TObject);
begin
  form2.Hide; form3.Show;
end;end.

```



**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

### **Лабораторная работа № 6**

**Тема: Имитационное моделирование. Разработка информационной модели обучающегося.**

**Цель занятия:** Получить практические навыки в разработке информационных систем в средах БД Access , MS Excel

**Задачи занятия:**

Создать графическую модель обучающегося с отображением диаграммы оценок по темам конкретной дисциплины

Информационная модель обучающегося

Предметная информационная модель знаний обучающихся курса или факультета позволит получить следующие характеристики учебного процесса:

- качество обучения в целом по университету, факультету, курсу и успеваемость отдельных обучающихся;
- стабильность обучения по отдельным предметам, темам и дисциплинам;
- прогнозирование результатов обучения в предстоящем семестре;
- систематическое уточнение модели знаний обучающихся с использованием тестовых программ;
- разработка индивидуальных вопросов и задач на основе моделей знаний обучающихся;

- сравнительные оценки состояния учебного процесса как на факультете, так и по университету в целом.

## Часть 1. Разработка имитационной модели знаний обучающихся в среде Excel



Семестр 1	16				
Тема 1. Введение	2				
Основы алгоритмизации					
Тема 2. Введение в П	2				
Тема 3. Структура пр	2				
Тема 4. Циклы и усл	4				
Тема 5. Массивы, на	4				
Тема 6. Специальны	2				
2-й семестр					
Тема 7. Символьны	2				
Тема 8. Графика. Ос	2				
Тема 9. Основные гр	4				
Тема 10. Построени	4				
Тема 11. Файлы	2				
Тема 12. Комбиниро	2				
Зачет					
Итого	32				
1. Вид промежуточной аттестации: зачет в семестре 2.					

Возврат

4.2. Содержание лекционных занятий!

Введение. Основы алгоритмизации – 2 час.

Программирование: сущность и назначение. Придмет дисциплины и ее задачи.

Структура, содержание дисциплины: ее связь с другими дисциплинами учебного плана специальности. Этапы решения задач на ЭВМ. Понятие алгоритма, свойства алгоритма. Данные и

2. Введение в язык программирования.

История и классификация языков программирования. Поколения ЭВМ. Компилятор и интерпретатор. Структура и способы описания языков программирования высокого уровня. Общая ст

3. Структура программ, типы данных на языках С++, Паскаль

Структура программы на Паскале. Элементы языка: алфавит, идентификаторы, комментарии. Этапы обработки программы на Турбо Паскале. Компоненты системы программирования Турб

4. Арифметические операции, функции, выражения, на языках С++, Паскаль

Арифметические операции, функции, выражения. Арифметические типы данных. Арифметические операции, выражения. Арифметический оператор присваивания.

Ввод с клавиатуры и вывод на экран. Ввод и вывод данных. Устройства ввода-вывода. Форматы вывода.

5. Циклы и условные операторы на языках С++, Паскаль

Ветвления и циклы в алгоритмах. Условный оператор. Оператор выбора. Программирование циклов. Цикл с предусловием. Цикл с постусловием. Цикл с параметром. Оператор continue.

6. Массивы, на языках С++, Паскаль

Табличные данные и массивы. Одномерные и многомерные массивы. Ввод и вывод массивов. Основные операции с массивами. Поиск и сортировки

7. Специальные типы данных: функции, на языках С++, Паскаль

# Информационная "Модель знаний по алгебре"

Об авторе

Условие

Учебная  
программа

Успеваемость



группа Дикт 32-11

Ф.И.О	Посещен	тема1		тема2		тема3		ЛР4		ЛР5		ЛР6		Индив. задание		сумма	Зач
		балл	оценка	балл	оценка	балл	оценка	балл	оценка	балл	оценка	балл	оценка	балл	оценка		
Андреев А		19														19	неЗ
Засильев		11														11	неЗ
Зармин																0	неЗ
Захаров																0	неЗ
Двинцов																0	неЗ
Егоров																0	неЗ
Иванов П		17		8												25	неЗ
Иванов Р		18		7												25	неЗ
Леонтьев																0	неЗ
Михайлов																0	неЗ
Петров		18														18	неЗ
Рисс																0	неЗ
Салтыкова																0	неЗ
Салтыков		17		9												17	неЗ
Ридоров																0	неЗ
Сорокин		15		7		9		18		9		9		18		85	Зач
Тавк		20		10		10		20		10		10		20		100	Зач
Тав		10		5		5		10		5		5		10		50	неЗ

Возврат

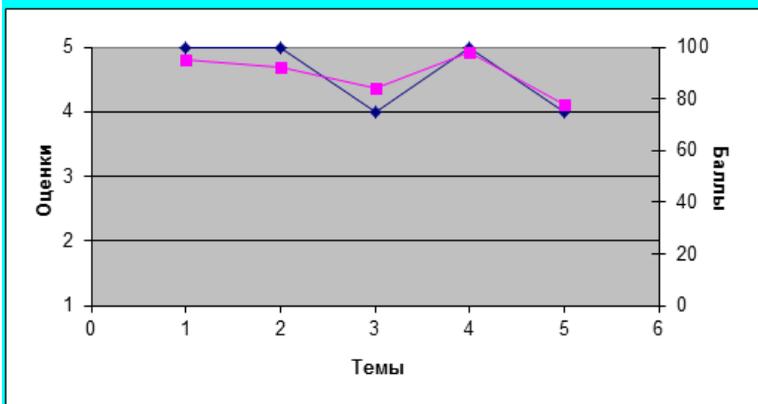
1. Отчеты за 1-й семестр 20
2. Реферат 10
3. Площадь круга 10
4. Метод Минимума 20

№ пп	Фамилия	Тема1	Тема2	Тема3	Тема4	Тема5	Итого	Балл1	Балл2	Балл3	Балл4	Балл5	Итого
1	Ананьева	5	5	5	5	5	5	91	90	94	95	92	93
2	Антонов	5	4	5	5	4	4,6	94	92	92	95	91	88,8
3	Анисимова	4	5	5	5	5	4,8	90	97	94	90	96	93,2
4	Андреев Ал.	5	5	4	5	4	4,6	95	92	94	98	70	89,4
5	Андреев Н.	5	3	3	4	5	4	91	65	97	97	91	86,2

Общий  
итого

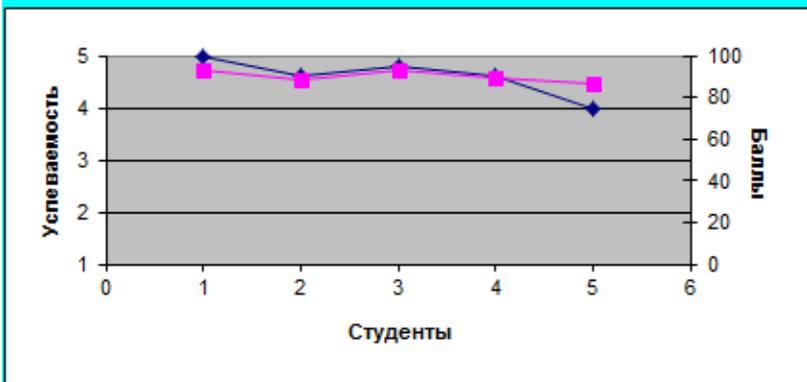
Вернуться

Диаграмма успеваемости Андреева Ал.



Вернуться

Графическая модель группы



Вернуться

# Учебная программа для 11 класса по дисциплине "Информатика"

## Глава 1:

1. Введение в информатику
2. Общие принципы организации и работы компьютеров
3. Классификация компьютеров
4. Арифметические основы компьютера
5. Логические основы компьютера
6. Программное обеспечение
7. Алгоритмы. Алгоритмические языки
8. Технология подготовки и решения задач с помощью компьютера
9. Применение компьютерной техники

Цели

Назад

## Глава 1:

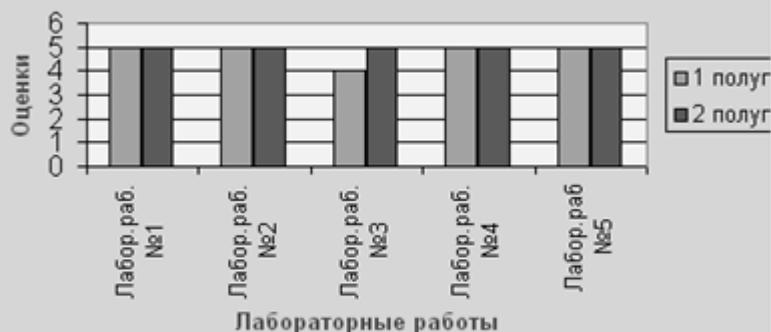
1. Алгоритмы линейной и разветвляющейся структуры
2. Алгоритмы, реализуемые с помощью циклов типа ДЛЯ
3. Алгоритмы, реализуемые с помощью вложенных циклов типа ДЛЯ
4. Алгоритмы, реализуемые с помощью циклов типа ПОКА
5. Алгоритмы, реализуемые с помощью вложенных циклов типа ПОКА
6. Алгоритмы, реализуемые с помощью комбинаций циклов типа ДЛЯ и ПОКА
7. Алгоритмы обработки символьной информации
8. Использование графики и звука в языке Turbo Pascal

## 1 полугодие

Назад

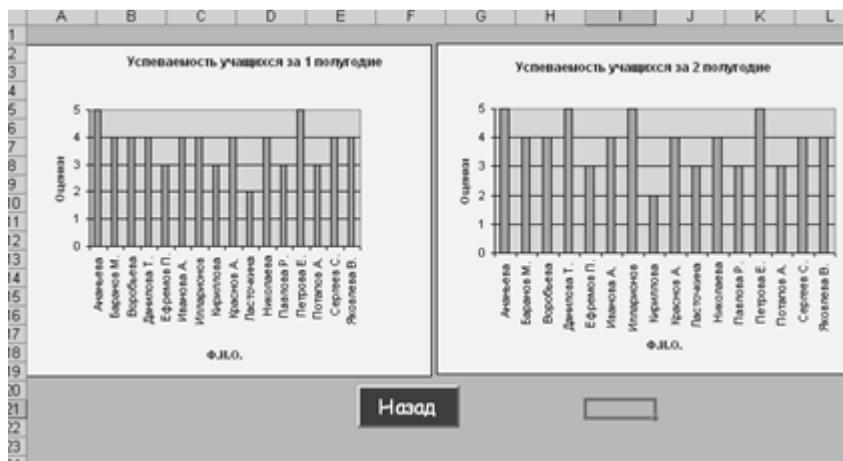
№	Ф.И.О.	Л.Р. №1		Л.Р. №2		Л.Р. №3		Л.Р. №4		Л.Р. №5		Сред. бал.	Оценка за 1 полугод.
		баллы	оценка										
1	Ананьева А.	10	5	9	5	10	5	9	5	10	5	9,60	5
2	Баранов М.	7	4	6	3	8	4	8	4	9	5	7,60	4
3	Воробьева И.	8	4	7	4	7	4	9	5	8	4	7,80	4
4	Данилова Т.	9	5	8	4	9	5	10	5	8	4	8,80	4
5	Ефремов П.	6	3	7	4	7	4	6	3	7	4	6,60	3
6	Иванова А.	9	5	8	4	8	4	7	4	7	4	7,80	4
7	Илларионов Н.	10	5	8	4	9	5	8	4	9	5	8,80	4
8	Кириллова Л.	3	2	5	3	6	3	8	4	4	3	5,20	3
9	Краснов А.	8	4	7	4	7	4	9	5	6	3	7,40	4
10	Ласточкина Л.	4	3	3	2	5	3	3	2	2	2	3,40	2
11	Николаева Н.	9	5	8	4	7	4	7	4	7	4	7,60	4
12	Павлова Р.	5	3	6	3	7	4	6	3	6	3	6,00	3
13	Петрова Е.	10	5	9	5	8	4	10	5	9	5	9,20	5
14	Потапов А.	6	3	7	4	5	3	7	4	6	3	6,20	3
15	Сергеев С.	8	4	9	5	9	5	9	5	9	5	8,80	4
16	Яковлева В.	8	4	7	4	8	4	9	5	7	4	7,80	4

Диаграмма успеваемости Кирилловой Л.



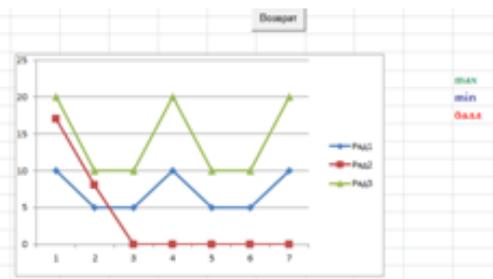
назад

Успеваемость учащихся за:				Ананьева А.		
Введите фамилию и инициалы учащегося:				Базаев М.		
				Воробьева И.		
				Данилова Т.		
				Ефремов П.		
				Иванова А.		
				Илларионов Н.		
				Кириллова Л.		
				Козлов А.		
				Ласточкина Л.		
				Николаева Н.		
				Павлова Р.		
				Петрова Е.		
				Поталов А.		
				Сергеев С.		
				Яковлев В.		
1						
2						
3						
4	<i>1 полугодие</i>			<i>2 полугодие</i>		
5	Лаб. раб.	ФИО	Петрова Е.	Лаб. раб.	ФИО	Петрова Е.
6	Лабор. раб. №1		5	Лабораторная работа №1		5
7	Лабор. раб. №2		5	Лабораторная работа №2		5
8	Лабор. раб. №3		4	Лабораторная работа №3		5
9	Лабор. раб. №4		5	Лабораторная работа №4		5
10	Лабор. раб. №5		5	Лабораторная работа №5		5
11	Оценка за полугодие		5	Оценка за полугодие		5
12						
13						
14	<b>Изменить</b>			<b>Изменить</b>		
15						
16						
17						
18						
19						
20						
21						



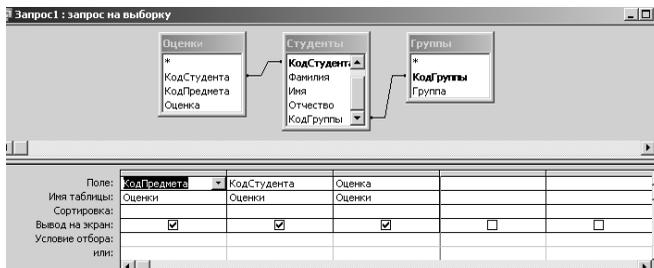
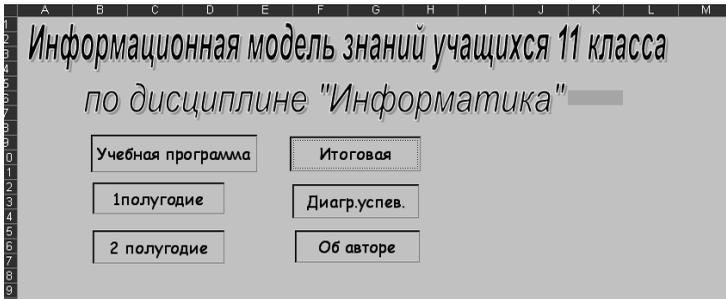
№ Тем	Иванов Павел	Зубаев	Б
1	10	17	20
2	5	8	10
3	5	0	10
4	10	0	20
5	5	0	10
6	5	0	10
7	10	0	20
8	50	20	100

используя результаты задания



1. Отчеты за 1-й семестр	20
2. Реферат	10
3. Площадь круга	10
4. Метод Монте Карло	20
5. Символы	10
6. Делфи	10
7. Калькулятор	20
всего	100 для задания 5! баллы при условии сдачи всех заданий

## Часть 2. Разработка имитационной модели обучающихся в среде MS Access



**Предметы : таблица**

	КодПредмета	Предмет
▶ +	1	Тема1
+	2	Тема2
+	3	Тема3
+	4	Тема4
+	5	Тема5
*	(Счетчик)	

Модель студента : форма

Заголовок формы

Область данных

Код студента	КодСтудента				
Фамилия	Фамилия				
Имя	Имя				
Отчество	Отчество				
Код группы	Код группы				

Выход

Примечание формы

Модель студента : запрос на выборку

Оценки	Предметы
* КодСтудента КодПредмета Оценка	* КодПредмета Предмет

Поле:	Предмет	Оценка	КодСтудента	
Имя таблицы:	Предметы	Оценки	Оценки	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			[Forms][Модель студента]	
или:				

## Информационная "Модель знаний по алгебре"

Об авторе

Условие

Учебная программа

Успеваемость



**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

## Лабораторная работа № 7

**Тема: Оптимизационные модели.**

**Цель занятия:** Получить практические навыки разработки оптимизационных моделей.

**Ход занятия:**

1. Изучение алгоритмов решения оптимизационных задач.
2. Разработка оптимизационных моделей.
3. Анализ результатов моделирования.

### Примеры разработки моделей транспортных задач

#### 1. Размещения предприятий

**Исходные условия.** Пусть в нескольких пунктах расположены предприятия, производящие некоторый продукт. В пунктах размещаются потребители готовой продукции с соответствующими потребностями. Затраты на производство единицы продукта в пункте равны  $a_i$ , объем производства в этом пункте равен  $b_i$ , а затраты по транспортировке единицы продукта из  $i$  в  $j$  равны  $c_{ij}$ . Количество перевезенных продуктов из  $i$  в  $j$  составляет  $x_{ij}$  единиц. Уравнения модели:

1. (перевозится неотрицательное количество продукта);
2. (выпускаемое количество продукта не больше возможного объема производства и равно вывозимому количеству продукта);
3.  $\sum_{j=1}^n x_{ij} = b_i$ ,  $\sum_{i=1}^m x_{ij} = a_j$  ( $a_j$  – потребности потребителей),  
(каждый пункт потребления получает столько, сколько ему требуется).

**Конечная цель.** Необходимо получить минимальные затраты на производство и на транспортировку продукции, т.е. обеспечить минимум функции

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$
$$x_i = \sum_{j=1}^n x_{ij} \leq a_i, \quad \sum_{i=1}^m x_{ij} = b_j, \quad \sum_{i=1}^m x_{ij} = b_j, \quad \sum_{i=1}^m x_{i,j+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

$$\min \left\{ c_i x_i + \sum_{i=1}^m \sum_{j=1}^{n+1} x_{ij} d_{ij} \right\}$$

,  
где - затраты на производство,

- затраты на транспортировку.

Целевая функция состоит из аналогичной целевой задачи.

Однако, различие состоит в том, что здесь добавляются затраты на производство. Для однотипности подхода к решению этих двух моделей добавляется фиктивный пункт с характеристикой: потребность равна разности между возможным объемом производства продукта и суммарной потребнос

,  
где - затраты на производство,

- затраты на транспортировку.

Целевая функция состоит из аналогичной целевой задачи. Однако, различие состоит в том, что здесь добавляются затраты на производство. Для однотипности подхода к решению этих двух моделей добавляется фиктивный пункт с характеристикой: потребность равна разности между возможным объемом производства продукта и суммарной

$$\sum_{i=1}^m \sum_{j=1}^{n+1} x_{ij} d_{ij}$$

**ТРАНСПОРТНАЯ ЗАДАЧА - Распределение производства**

Исходные данные:  Оптимальный план  Графическое изображение  0 программе

Ввод данных

Введите кол-во пунктов производства и потребления

Ввод кол-во товара у производителей

Затраты по произ-ву ед. продукта

Ввод заявок из пунктов потребления

Затраты по перевозке продукции: стоимость ед.перевозки из пункта произ-ва в пункт потребления



**ИСХОДНЫЕ ДАННЫЕ:**

Объем и стоимость производства в 4 пункте(ах):

Пункты	A1	A2	A3	A4
Кол-во	100	200	300	50
Стоим-ть	70	20	40	10

Потребность в 4 пункте(ах) потребления:

Пункты	B1	B2	B3	B4
Кол-во товара	10	40	15	75

Матрица затрат на перевозку Потребитель

	B1	B2	B3	B4
A1	10	15	14	18
A2	9	4	17	7
A3	12	6	9	19
A4	6	9	15	16

П  
р  
о  
и  
з  
в  
о  
д  
и  
т.

## 2. Планирование производства товаров на основе модели получения максимальной прибыли с использованием метода линейного программирования.

Пусть предприятие производит столы и стулья. Расход ресурсов на их производство и прибыль от их реализации представлены в таблице. столы стулья объем ресурсов

Расход древесины

на изделие, м3 0,5 0,04 200

Расход труда(чел-час) 12 0,6 1800

Прибыль от реализации единицы изделия, руб 180 20

Кроме того на производство 80 столов имеется заказ с министерством

**Уравнения:**

$x_1$  - количество столов;  $x_2$  - количество стульев

$$0,5x_1 + 0,04x_2 \leq 200$$

$$12x_1 + 0,6x_2 \leq 18000$$

$$x_1 + 20x_2 = \max \text{ (целевая функция)}$$

Microsoft Excel - книга31

Файл Правка Вид Вставка Формат Сервис Данные Work It

New List Get Work Items Publish Refresh

D15 =B15+C15

	A	B	C	D	E
1	<b>поиск решения</b>				
2			столы	стулья	объем
3	расход древесины		0,95	0,076	380
4	расход труда		22,8	1,14	3420
5	прибыль		342	38	
6					
7					
8					
9					
10					
11	x	ограничения			ресурс
12	80	228	478,8	420	
13	4700	3780	7938	3780	
14	прибыль	378	42		
15	целевая функц	30240	197400	227640	
16					
17					
18			ПРИБЫЛЬ СОСТАВИТ 227640 РУ		
19					

Поиск решения

Установить целевую ячейку:

Равной:  максимальному значению  значению:

минимальному значению

Изменяя ячейки:

Ограничения:

Линейное программирование			
Решить следующую задачу оптимального планирования:			
$1^{\circ}x_1+6^{\circ}x_2\leq 60$			
$2^{\circ}x_1+10^{\circ}x_2\leq 100$			
$F=30^{\circ}x_1+100^{\circ}x_2$			
Плановые показатели			
	x1	x2	
	0	10	
Ограничения			
	Левая часть	Знак	правая часть
	60	<=	60
	100	<=	100
Целевая функция			
	1080		

После выполнения работ студентами представляется отчет, содержащий следующие разделы:

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

## Лабораторная работа №8

**Тема:** *Стохастическое моделирование: применение равномерного распределения вероятностей*

**Цель занятия:** Получить практические навыки использования равномерно распределенных случайных чисел при организации стохастического моделирования

### **Ход занятия:**

1. Разработка алгоритмов решения задач.
2. Реализация программы на избранном языке программирования.
3. Анализ результатов моделирования.

### Вариант 1.

1. Примените метод Монте-Карло для подсчёта площади:
  - а) круга с центром в точке  $O(-3;2)$  и радиусом  $R=2$ ;
  - б) фигуры, ограниченной линиями  $y=-x^2+3$  и  $y=-6$ ;
  - в) фигуры, ограниченной линиями  $y=-5/x$ ,  $y=4-x$  и  $x=1$ .
2. Площадь тех же фигур получите теоретически и сравните с результатами применения метода Монте-Карло.
3. Смоделируйте блуждание материальной точки вдоль прямой, перпендикулярной краю пропасти. Подсчитайте вероятность падения точки в пропасть, если она находится на расстоянии 5 шагов от края пропасти, а безопасным можно считать расстояние 15 шагов от края.
4. Смоделируйте процесс тестирования. Тест содержит 10 вопросов. На каждый из них предлагается 3 варианта ответов, один из которых верный. Ученик плохо подготовлен и выбирает ответы наугад. Определите вероятность того, что ученик получит "3", "4", "5", если "3" выставляется за 5-6 верных ответов, "4" - за 7-8 верных ответов, "5" - за 9-10 верных ответов.

### Вариант 2.

1. Примените метод Монте-Карло для подсчёта площади:
  - а) круга с центром в точке  $O(-3;-5)$  и радиусом  $R=3$ ;

б) фигуры, ограниченной линиями  $y=x^2+2$  и  $y=6$ ;

в) фигуры, ограниченной линиями  $y=2/x$ ,  $y=x-3$ ,  $x=1$  и  $x=2$ .

2. Площадь тех же фигур получите теоретически и сравните с результатами применения метода Монте-Карло.

3. Смоделируйте блуждание материальной точки вдоль прямой, перпендикулярной краю пропасти. Подсчитайте вероятность падения точки в пропасть, если она находится на расстоянии 8 шагов от края пропасти, а безопасным можно считать расстояние 20 шагов от края.

4. Смоделируйте процесс тестирования. Тест содержит 6 вопросов. На каждый из них предлагается 4 варианта ответов, один из которых верный. Ученик плохо подготовлен и выбирает ответы наугад. Определите вероятность того, что ученик получит "3", "4", "5", если "3" выставляется за 3-4 верных ответа, "4" - за 5 верных ответов, "5" - за 6 верных ответов.

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления
- (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

## Лабораторная работа №9

**Тема:** *Стохастическое моделирование: применение нормального распределения вероятностей*

**Цель занятия:** Получить практические навыки использования нормально распределенных случайных чисел при организации стохастического моделирования

**Ход занятия:**

1. Разработка алгоритмов решения задач.
2. Реализация программы на избранном языке программирования.
3. Анализ результатов моделирования.

### Вариант 1.

1. Разработайте программу для получения случайных чисел, подчиненных нормальному закону распределения с параметрами  $a$  и  $\sigma^2$ . Получите с ее помощью выборку объемом  $N=100$  при  $a=50$  и  $\sigma^2=16$ .

Организуйте расчет среднего, дисперсии, стандартного отклонения и размаха выборки. Сопоставьте полученные значения с соответствующими теоретическими параметрами.

2. Известно, что рост мужчин некоторой возрастной группы есть нормально распределённая случайная величина  $X$  с параметрами  $a=173$  см и  $\sigma^2=36$  см. Определите экспериментально, каковы должны быть доли костюмов 3-го роста (170-176 см) и 4-го роста (176-182 см) в общем объеме производства. Сравните полученные значения с теоретическими.

### Вариант 2.

1. Разработайте программу для получения случайных чисел, подчиненных нормальному закону распределения с параметрами  $a$  и  $\sigma^2$ . Получите с ее помощью выборку объемом  $N=100$  при  $a=50$  и  $\sigma^2=16$ .

Организуйте расчет среднего, дисперсии, стандартного отклонения и размаха выборки. Сопоставьте полученные значения с соответствующими теоретическими параметрами.

2. Пачки с сыпучим продуктом упаковываются автоматически. Их средняя масса равна 1,05 кг, стандартное отклонение 0,01кг. Предполагая, что вес пачек распределен нормально, определите вероятность покупки пачки, весящей

а) менее 1 кг; б) от 1,07 кг до 1,08 кг.

Сравните полученные значения с теоретическими.

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления
- (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.

### **Лабораторная работа №10.**

**Тема:** *Исследование вероятностных систем (систем массового обслуживания) с использованием различных информационных технологий.*

**Цель занятия:** Получить практические навыки исследования систем ( объектов) с вычислением параметров статистических моделей и определением эффективности исследуемых систем.

**Ход занятия:**

1. Разработка алгоритмов
2. Разработка программы исследования систем массового обслуживания
3. Анализ результатов исследований с вычислением параметров и эффективности систем.

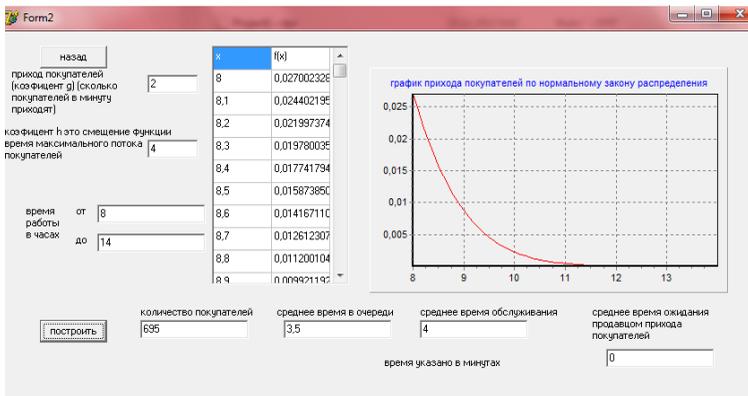
**Пример:** Исследовать параметры обслуживания покупателей в магазине с одним продавцом

Провести моделирование очереди в магазине с одним продавцом при нормальном законе распределения описанных выше случайных величин: прихода покупателей и длительности обслуживания (при некотором фиксированном наборе параметров). Получить устойчивые характеристики: средние значения ожидания в очереди покупателем и простой продавца в ожидании прихода покупателей. Оценить их достоверность. Оценить характер функции распределения величин  $g$  и  $h$ .

Ввод значений

Начало рабочего дня	10
Конец рабочего дня	12
Интенсивность потока покупателей	0,1

Пуск



## Программа

unit Unit2;

function f(x:real;g:real;h:real):real;

begin

```
f:=(1/(g*sqrt(3.14*2)))*power(exp(1,(((x-h)*(x-h))/(-2*g*g)))); //
```

нормальный закон распределения

end;

procedure TForm2.Button2Click(Sender: TObject);

var n,i,a,b,rk:integer;

g,h,f1,fn,h1,x1,sr,t,y,kol,vrop:real;

begin

randomize;

```
g:=strtofloat(edit1.text); // среднеквадратич отклонение X
```

```
h:=strtofloat(edit2.text); //мат ожидание
```

```
Chart1.Series[0].Clear; //очистка графика
```

```
a:=StrToint(Edit4.Text); //левая граница
```

```
b:=StrToint(Edit5.Text); //правая граница
```

```
h1:=0.1;
```

```
x1:=a;
```

```
n:=1;
```

```
fn:=0;
```

```
t:=h1;
```

```
stringgrid1.Cells[0,0]:='x';
```

```
stringgrid1.Cells[1,0]:='f(x)';
```

```
repeat
```

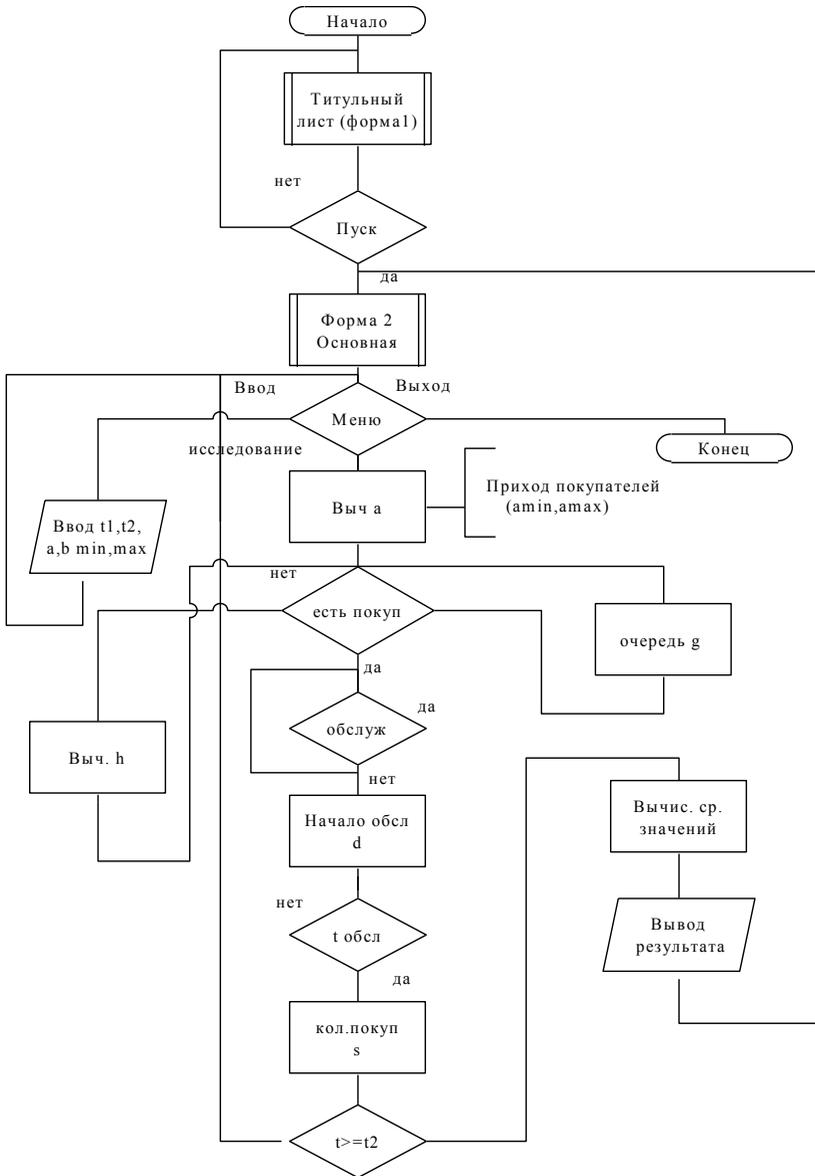
```
  f1:=f(x1,g,h);
```

```
  Chart1.Serieslist[0].AddXY(x1,f1,"",clRed); //график F(x)
```

```

stringgrid1.Cells[0,n]:=floattostr(x1);
stringgrid1.Cells[1,n]:=floattostr(f1);
x1:=x1+h1;
n:=n+1;
fn:=f1+fn;
stringgrid1.RowCount:=n+1;
until x1>b;
kol:=(b-a)*60*g;
edit6.text:=floattostr(kol);
rk:=strtoint(edit6.text);
i:=random(rk);
edit7.text:=inttostr(random(5));
sr:=strtofloat(edit7.text);
vrop:=((b-a)*60-sr*kol)/kol;
if vrop>=0 then
edit8.Text:=floattostr(((b-a)*60-sr*kol)/kol)
else edit8.Text:='0';
if vrop<=0 then
edit3.Text:=floattostr(-vrop) else
edit3.Text:='0';
edit6.text:=floattostr(i);
end;
end.

```



Ввод значений

Начало рабочего дня

Конец рабочего дня

Интенсивность потока покупателей

Form3

Старт    Характеристические показатели    Выход

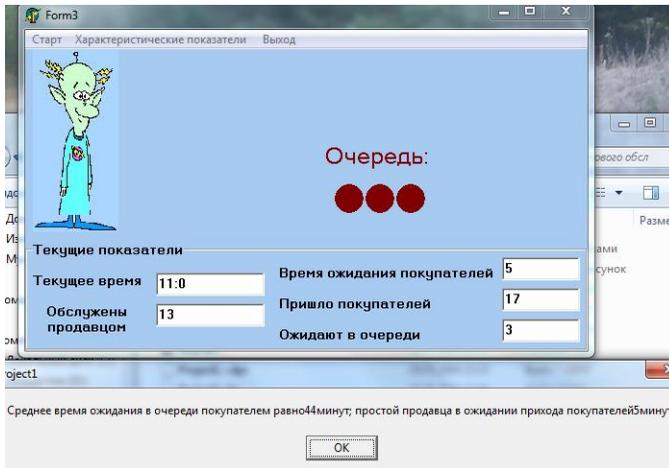


Очередь:



Текущие показатели

Текущее время	<input type="text" value="10:23"/>	Время ожидания покупателей	<input type="text" value="5"/>
Обслужены продавцом	<input type="text" value="3"/>	Пришло покупателей	<input type="text" value="7"/>
		Ожидают в очереди	<input type="text" value="3"/>



```
unit Unit3;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics,  
Controls, Forms, Dialogs, Menus, ExtCtrls, StdCtrls, jpeg;
```

```
type
```

```
TForm3 = class(TForm)
```

```
  MainMenu1: TMainMenu;
```

```
  N1: TMenuItem;
```

```
  N2: TMenuItem;
```

```
  Timer1: TTimer;
```

```
  GroupBox1: TGroupBox;
```

```
  Edit1: TEdit;
```

```
  Edit2: TEdit;
```

```
  Edit3: TEdit;
```

```
  Edit4: TEdit;
```

```
  Edit5: TEdit;
```

```
  Label1: TLabel;
```

```
  Label2: TLabel;
```

```
  Label3: TLabel;
```

```
  Label4: TLabel;
```

```

Label5: TLabel;
Timer2: TTimer;
Image1: TImage;
N3: TMenuItem;
Image2: TImage;
Label6: TLabel;
procedure N2Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure N3Click(Sender: TObject);
var
Form3: TForm3;
a,b,t,kol,za,k,i,l,j,ll,gdet,zanvr:Integer;
nach,vrem,kon,pokgdet,pok,x1,y1:Integer;
Buyx,Buyy:Integer;
obsl,prish:Integer;
tekvr,chas,minut:String;
pokin:array[1..1000] of real;
pokvr1:array[1..1000] of real;
pokvr:array[1..1000] of real;
prod:array[1..1000] of real;
ogid:array[1..1000] of real;
ob:array[1..1000] of real;
p,s,inte,buy,x,y,maxt,tt,vr:real;
implementation
uses Unit2, Unit1;
procedure TForm3.N2Click(Sender: TObject);
begin
form3.Hide;
form1.show;
end;
procedure TForm3.Timer1Timer(Sender: TObject);
begin
If zanvr=prod[k] Then {если пок-ля обслужили}
begin
timer2.Enabled:=true;
image1.Visible:=true;

```

```

timer2.Interval:=100;
zanvr:=0;
k:=k+1;
If ogid[k]=0 Then {если пок-ль не в очереди}
begin
za:=za+1; {увеличиваем кол-во пришедших}
edit4.Text:=inttostr(za);
end;
If kol>0 Then begin
kol:=kol-1; {уменьшаем кол-во ждущих в очереди}
edit5.Text:=inttostr(kol);
form3.Canvas.Pen.Color:=clskyblue;
form3.Canvas.Brush.Color:=clskyblue;
form3.Canvas.Ellipse(x1,y1,x1+30,y1+30);
x1:=x1-32;
end;
obs1:=obs1+1; {увел-м кол-во обслуженных}
edit2.Text:=inttostr(obs1);
end;
buy:=pokvr[k];
If buy=i Then begin {если покупатель пришел}
zanvr:=zanvr+1; {начинаем обслуживать}
pokvr[k]:=i+1;
end
Else gdet:=gdet+1; {иначе увел-ся простой продавца}
For j:=k+1 To l-1 do
begin
x:=pokvr[j];
y:=pokvr[k];
If x<y Then begin {если еще есть пришедшие}
pokvr[j]:=pokvr[k];
If ogid[j]=0 Then begin
kol:=kol+1; {увел-м кол-во ждущих}
edit5.Text:=inttostr(kol);
x1:=x1+32;
form3.Canvas.Pen.Color:=clmaroon;
form3.Canvas.Brush.Color:=clmaroon;
form3.Canvas.Ellipse(x1,y1,x1+30,y1+30);

```

```

        {увел-м кол-во пришедших}
        za:=za+1;
        edit4.Text:=inttostr(za);
        ogid[j]:=1;
end;
end;
end;
i:=i+1;    {увеличиваем время}
{преобразуем время в часы и минуты}
str((i div 60),chas);
str((i mod 60),minut);
tekvr:=chas+':' +minut;
edit1.Text:=tekvr;
edit3.Text:=inttostr(gdet);
if i>=kon then timer1.Enabled:=false;
end;

procedure TForm3.N1Click(Sender: TObject);
var stroka,stroka1:string;
begin
    For i:=1 To obsl do
        begin
            ob[i]:=pokvr[i]-prod[i]-pokvr1[i];
            s:=s+ob[i];
        end;
    For i:=obsl+1 To l-1 do {суммируем время ожидания в оч-ди}
        begin
            ob[i]:=pokvr[i]-pokvr1[i];
            s:=s+ob[i];
        end;
    s:=s/(l-1); {находим среднее арифметич-ое}
    stroka1:=inttostr(gdet);
    stroka:=inttostr(round(s));
    showmessage('Среднее время ожидания в очереди покупателем
    равно'+stroka+'минут; простой продавца в ожидании прихода
    покупателей'+stroka1+'минут');
end;

```

```

procedure TForm3.Timer2Timer(Sender: TObject);
begin
  Image1.Left:=Image1.Left-30;
  If Image1.Left<112 Then begin
    Timer2.Enabled:=False;
    Image1.Visible:=False;
    Image1.Left:=208;
  End;
end;

```

```

procedure TForm3.N3Click(Sender: TObject);
begin
  nach:=strtoint(form2.edit1.Text)*60;
  inte:=strtofloat(form2.Edit3.Text);
  kon:=strtoint(form2.edit2.Text)*60;
  pok:=0;
  gdet:=1;
  k:=0;
  zanvr:=0;
  vrem:=0;
  obsl:=0;
  i:=1;
  vr:=nach;
  gdet:=0;
  k:=1;
  x1:=290;
  y1:=140;
  maxt:=random(5); {максимальное время}
  maxt:=(maxt+1)/100;
  While vr<kon do
  begin
    {интервал прихода покупателей}
    pokin[i]:=random(7); {в соответствии с функцией Пуассона}
    pokin[i]:=Int((1/inte)*((pokin[i]+1)/10))+1;
    tt:=(inte*pokin[i])*(exp((-inte*pokin[i])*ln(2.71)));
    If tt>maxt Then
    begin
      vr:=vr+pokin[i];
      pokvr[i]:=vr; {время прихода пок-лей}
    end;
  end;

```

```

pokvr1[i]:=vr;
i:=i+1;
end;
end;
For l:=1 To i do
begin          {время, кот. тратит продавец}
prod[l]:=random(8);
prod[l]:=Int((1/inte)*((prod[l]+1)/10))+1;
ob[l]:=0;
ogid[l]:=0;
end;
i:=nach;
timer1.enabled:=true;
timer1.interval:=500;
end;
end.

```

**После выполнения работ студентами представляется отчет, содержащий следующие разделы:**

- постановка задачи (начальные условия);
- математическая модель;
- метод исследования модели;
- алгоритм моделирования задачи (блок-схема);
- программа на языке программирования, заданным преподавателем;
- результаты в различных формах представления
- (табличный, графический, динамический);
- содержательный анализ результатов моделирования, выводы.