

Тема урока: Язык программирования как средство информационного компьютерного моделирования.

Разработка моделирующих компьютерных программ.

Теоретическая часть

Рассмотрим процесс построения и исследования модели на конкретном примере движения тела, брошенного под углом к горизонту.

Содержательная постановка задачи «Бросание мячика в стенку»

В процессе тренировок теннисистов используются автоматы по бросанию мячика. Необходимо задать автомату необходимую скорость и угол бросания мячика для попадания в стенку определенной высоты, находящуюся на известном расстоянии.

Качественная описательная информационная модель

Сначала построим качественную описательную модель процесса движения тела с использованием объектов, понятий и законов физики, т.е. в данном случае идеализированную модель движения объекта. Из условий задачи можно сформулировать следующие основные допущения:

- Мячик мал по сравнению с Землей, поэтому его можно считать материальной точкой
- Изменение высоты мячика мало, поэтому ускорение свободного падения можно считать постоянной величиной $g=9,8 \text{ м/с}^2$, следовательно движение по оси Y можно считать равноускоренным.
- Скорость бросания мячика мала, поэтому сопротивлением воздуха можно пренебречь, следовательно, движение по оси X можно считать равномерным.

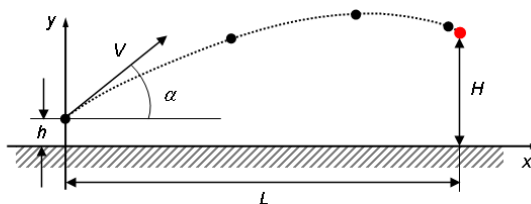
Формальная модель

Для формализации модели обозначим величины:

- Начальная скорость мячика – V
- Угол бросания мячика – α
- Высоту стенки – h
- Расстояние до стенки – s

Используем известные из курса физики формулы равномерного и равноускоренного движения для определения координат мячика. Дальность x и высоту y при заданной начальной скорости V и угле бросания α для любого момента времени t можно рассчитать по следующим формулам:

Графическая модель



Формальная (математическая) модель

$$x = V \cos \alpha \cdot t, \quad y = h + V \sin \alpha \cdot t - \frac{gt^2}{2}$$

Чтобы определить, попадет ли мячик в стенку, необходимо вычислить его координату y в момент времени, когда он будет находиться на расстоянии s :

$$t = \frac{s}{V \cdot \cos \alpha}$$

Подставляем эти значения времени t в формулу для вычисления координаты y . Получаем L – высоту мячика над землей на расстоянии s :

$$L = s \cdot \operatorname{tg} \alpha - \frac{g \cdot s^2}{2 \cdot V^2 \cdot \cos^2 \alpha}$$

Формализуем теперь условие попадания мячика в мишень. Попадание произойдет, если значение высоты мячика L будет удовлетворять условию в форме неравенства:

$$0 \leq L \leq h$$

Если $L < 0$, то это значит «недолет», а если $L > h$, то это означает «перелет».

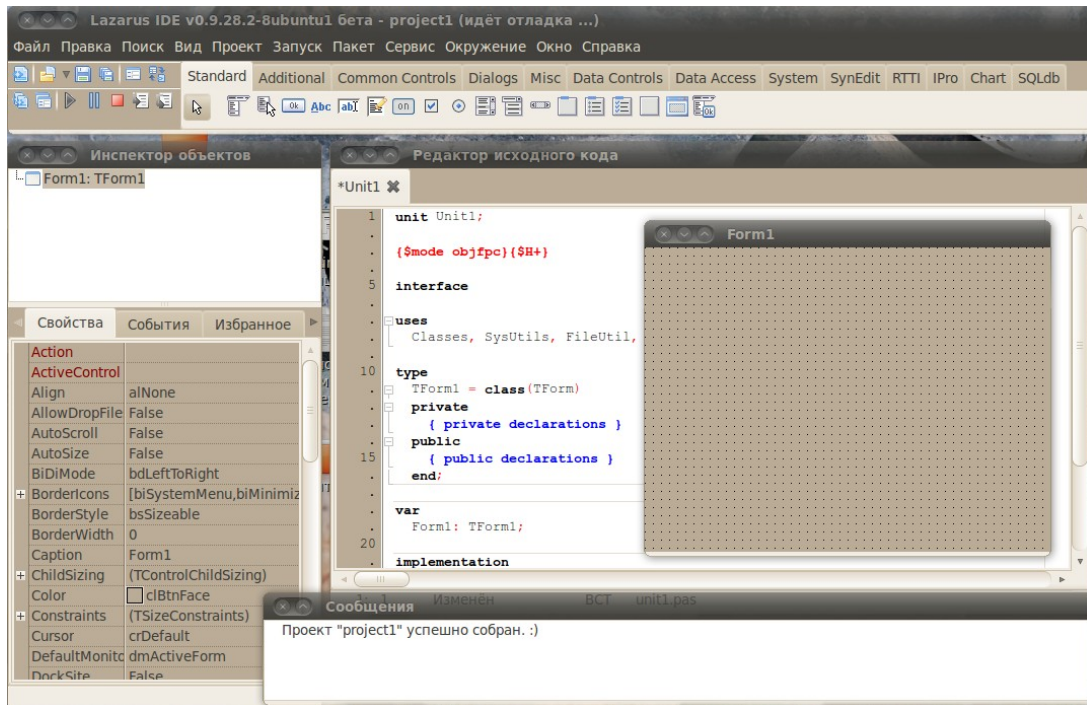
На основании формальной модели, описывающей движение тела, брошенного под углом к горизонту, создадим компьютерную модель с использованием системы программирования Lazarus.

Для этого:

1) Запустить среду разработки приложений (систему программирования) Lazarus

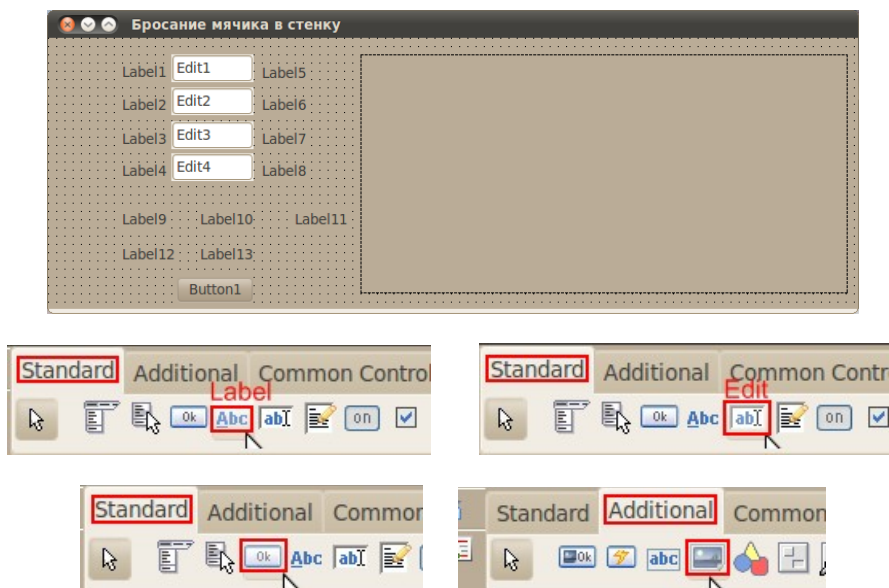
1.1) Приложения – Программирование – Lazarus

1.2) На экране появятся окна данной программы:



2) Создадим сначала графический интерфейс программы:

2.1) Разместите на форме компоненты в соответствии с указанным ниже порядком:



2.2) Сделать настройки для каждого из элементов, расположенных на форме в соответствии с теми, что приведены ниже:

Настройка свойств производится следующим образом:

- на форме щелкаем левой кнопкой мыши по нужному объекту (он выделится квадратиками),
- далее переходим к окну «**Инспектор объектов**» и изменяем свойства так, как указано ниже:

Настройка формы (Form1)

Caption -> **Бросание мячика в стенку** (название окна формы)

Height -> **260** (отступ надписи от левого края формы)

Width -> **776** (отступ надписи от верхнего края формы)

Надпись (Label1)

Caption -> **Начальная скорость (V0):** (название надписи)

Left -> **16** (отступ надписи от левого края формы)

Top -> **22** (отступ надписи от верхнего края формы)

Надпись (Label2)

Caption -> **Угол бросания мячика (A):** (название надписи)

Left -> **16** (отступ надписи от левого края формы)

Top -> **53** (отступ надписи от верхнего края формы)

Надпись (Label3)

Caption -> **Расстояние до стенки (S):** (название надписи)

Left -> **16** (отступ надписи от левого края формы)

Top -> **84** (отступ надписи от верхнего края формы)

Надпись (Label4)

Caption -> **Высота стенки (H):** (название надписи)

Left -> **16** (отступ надписи от левого края формы)

Top -> **118** (отступ надписи от верхнего края формы)

Текстовое поле (Edit1)

Left -> **208** (отступ надписи от левого края формы)

Name -> **EditV0** (название объекта для системы программирования)

Top -> **16** (отступ надписи от верхнего края формы)

Width -> **80** (ширина текстового поля)

Text -> (оставить поле пустым)

Текстовое поле (Edit2)

Left -> **208** (отступ надписи от левого края формы)

Name -> **EditA** (название объекта для системы программирования)

Top -> **47** (отступ надписи от верхнего края формы)

Width -> **80** (ширина текстового поля)

Text -> (оставить поле пустым)

Текстовое поле (Edit3)

Left -> **208** (отступ надписи от левого края формы)

Name -> **EditS** (название объекта для системы программирования)

Top -> **79** (отступ надписи от верхнего края формы)

Width -> **80** (ширина текстового поля)

Text -> (оставить поле пустым)

Текстовое поле (Edit4)

Left -> **208** (отступ надписи от левого края формы)

Name -> **EditL** (название объекта для системы программирования)

Top -> **112** (отступ надписи от верхнего края формы)

Width -> **80** (ширина текстового поля)

Text -> (оставить поле пустым)

Надпись (Label5)

Caption -> **м/с** (название надписи)

Left -> **296** (отступ надписи от левого края формы)

Top -> **22** (отступ надписи от верхнего края формы)

Надпись (Label6)

Caption -> **град** (название надписи)

Left -> **296** (отступ надписи от левого края формы)

Top -> **52** (отступ надписи от верхнего края формы)

Надпись (Label7)

Caption -> **м** (название надписи)

Left -> **296** (отступ надписи от левого края формы)

Top -> **84** (отступ надписи от верхнего края формы)

Надпись (Label8)

Caption -> **м** (название надписи)

Left -> **296** (отступ надписи от левого края формы)

Top -> **118** (отступ надписи от верхнего края формы)

Надпись (Label9)

Caption -> **Точность:** (название надписи)

Left -> **72** (отступ надписи от левого края формы)

Top -> **168** (отступ надписи от верхнего края формы)

Надпись (Label10)

Alignment -> **taCenter** (выравнивание текста по центру)

AutoSize -> **False** (не подгонять размер надписи по длине текста)

Caption -> сделать пустым (название надписи)

Left -> **148** (отступ надписи от левого края формы)

Top -> **168** (отступ надписи от верхнего края формы)

Width -> **190** (ширина надписи)

Надпись (Label11)

Caption -> **м** (название надписи)

Left -> **344** (отступ надписи от левого края формы)

Top -> **168** (отступ надписи от верхнего края формы)

Width -> **24** (ширина надписи)

Надпись (Label12)

Caption -> **Результат:** (название надписи)
Left -> **72** (отступ надписи от левого края формы)
Top -> **192** (отступ надписи от верхнего края формы)

Надпись (Label13)

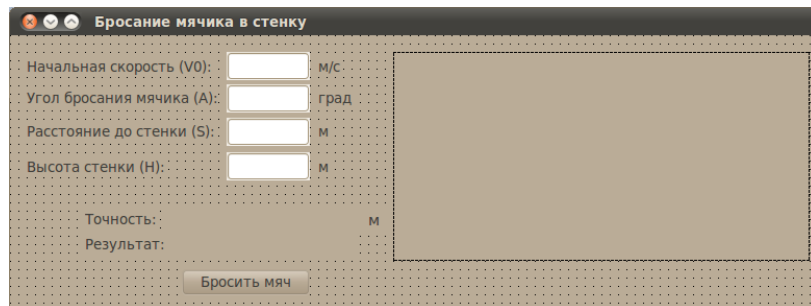
Alignment -> **taCenter** (выравнивание текста по центру)
AutoSize -> **False** (не подгонять размер надписи по длине текста)
Caption -> сделать пустым (название надписи)
Left -> **152** (отступ надписи от левого края формы)
Top -> **192** (отступ надписи от верхнего края формы)
Width -> **177** (ширина надписи)

Кнопка (Button1)

Caption -> **Бросить мяч** (название надписи)
Left -> **165** (отступ надписи от левого края формы)
Top -> **224** (отступ надписи от верхнего края формы)
Width -> **123** (ширина кнопки)

Изображение (Image1)

Height -> **200** (высота картинки)
Left -> **368** (отступ надписи от левого края формы)
Top -> **16** (отступ надписи от верхнего края формы)
Width -> **400** (ширина картинки)



3) Отредактировать код программы в соответствии с тем, как указано ниже:

3.1) Перейти к окну «Редактор исходного кода»:

3.2) Находим следующий код:

```
uses
```

```
Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs;
```

...и дописываем к нему **Math** (модуль, который обеспечивает выполнение математических функций в создаваемой программе). Должно получиться следующее:

```
uses
```

```
Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs, Math;
```

3.3) Далее находим следующий код:

```
var
```

```
Form1: TForm1;
```

...и дописываем к нему переменные, которые будут использоваться во время работы программы

```
var
```

```
Form1: TForm1;  
v0, a, s, h : real;
```

v0 – для хранения значения начальной скорости
a – для хранения значения угла бросания мячика
s – для хранения значения расстояния до стенки
h – для хранения значения высоты стенки

... далее выше этого кода дописываем следующий код:

```
Const
```

```
G = 9.8;  
Pi = 3.14;
```

G – ускорение свободного падения
Pi – число Пи

Должно получиться следующее:

```
Const
```

```
G = 9.8;  
Pi = 3.14;
```

```
var
```

```
Form1: TForm1;  
v0, a, s, h : real;
```

3.4) Далее напишем код программы, который будет выполняться при нажатии на кнопку **Бросить мяч**. Для этого делаем двойной щелчок по этой кнопке. На первый план перейдет окно «**Редактор исходного кода**» со следующим кодом:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
end;
```

...приводим его к следующему виду:

```
procedure TForm1.Button1Click(Sender: TObject);
Var L, t : real;
      x, y, n : integer;
Begin
//ниже написан код вычисления точности попадания мячика в стенку. Эту строку переписывать не надо
v0 := strtoint(editV0.text);
a := strtoint(editA.text);
s := strtoint(editS.text);
h := strtoint(editL.text);
L := s * tan(a*pi/180) - g*sqr(s)/(2*sqr(v0*cos(a*pi/180)));
Label10.Caption := floattostr(L);
if L<0 then label13.Caption := 'Недолет' else
if L > h then label13.Caption := 'Перелет' else
Label13.Caption := 'Попадание';
//ниже написан код построения траектории движения мячика. Эту строку переписывать не надо
image1.Picture := nil;
with image1.Canvas do
begin
while t < 5 do
begin
t:=t+0.005;
y:=180-round(10*(v0*sin(a*pi/180)*t-g*t*t/2));
x:=5+round(10*(v0*cos(a*pi/180)*t));
pixels[x,y]:=clblack;
end;
moveto(0,180); lineto(400,180);
moveto(5,0); lineto(5,400);
moveto(round(5+10*s),round(180));
lineto(round(5+10*s),180-round(10*h));
n:=0;
while n<400 do
begin
n:=n+50;
moveto(5+n,180); lineto(5+n,200);
textout(7+n,180,inttostr(round(n/10)));
end;
n:=0;
while n < 200 do
begin
n:=n+50;
moveto(0,180-n); lineto(10,180-n);
textout(0,180-n,inttostr(round(n/10)));
end;
end;
end;
```

4) Нажать кнопку **F9** (произойдет компиляция и запуск программы на выполнение) и анализируем модель при помощи программы:

4.1) Анализ результатов:

Например, диапазон значений угла бросания мячика от **32,6** до **36,1 градусов**, обеспечивает попадание в стенку высотой **1 метр**, находящуюся на расстоянии **30 метров**, мячиком, брошенным со скоростью **18 м/с**.

Например, диапазон значений угла бросания мячика от **55,8** до **57,4 градусов**, обеспечивает попадание в стенку высотой **1 метр**, находящуюся на расстоянии **30 метров**, мячиком, брошенным со скоростью **18 м/с**.

