Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

Кузбасский государственный технический университет

Кафедра информационных и автоматизированных производственных систем

**Пояснительная записка**

к курсовой работе по курсу “Технология программирования”

**Содержание**

1. Задание
2. Математическая постановка
   1. Уточнение формулировки задачи
   2. Математическое описание процесса
   3. Спецификация переменных (обозначения, типы, назначение, классификация)
   4. Спецификация объектов (назначение, свойства)
   5. Организация процесса и расчетные формулы
3. Разработка блок-схемы
4. Текст программы на VB
5. Контрольный пример
6. Инструкция пользователя

Список использованной литературы

1. ***Задание***

Разработать проект программного средства "Движение по эллиптическому маршруту с регулируемой скоростью и графической визуализацией процесса".

Параметры маршрута задаются в м. или км. Скорость движения в м/с или км/час. Образ движущегося объекта – цветное пятнышко, перемещающееся по линии эллипса.

По ходу движения отображаются координаты объекта, пройденный путь от места старта , время движения, число полных пройденных кругов. По каждому кругу фиксируется средняя скорость и время прохождения.



1. ***Математическая постановка***
   1. ***Уточнение формулировки задачи и выбор языка программирования***

Для решения задачи необходимо в первую очередь выбрать язык программирования и с помощью его средств и возможностей, решать поставленную задачу.

Для решения задачи воспользуемся языком программирования Visual Basic, поскольку он является идеальным средством для быстрого создания небольших проектов. Средства интегрированной среды разработки языка Visual Basic позволяют использовать *Форму,* для создания графического интерфейса проекта, что удобно для решения поставленной задачи.

Параметры эллипса, вращающегося тела, используемые объекты, а также их свойства в условии задачи не оговорены, и задаются произвольно, таким образом, чтобы отвечать всем условиям задачи, а также обеспечивать лёгкость применения программы, её функциональность, эффективность, мобильность, надёжность, эстетичность.

* 1. ***Математическое описание процесса***

программный математический пользователь

Параметрически эллипс задаётся в виде: *x=a\*cos(fi) y=b\*cos(fi),* где *fi* – угол в промежутке *[0; 2π].* Эти формулы используются для организации движения объекта по траектории эллипса. Параметры *a* и *b* задаются произвольно, подбираются опытным путём, в зависимости от размера формы.

Длина эллипса рассчитывается с помощью эллиптического интеграла, вида , таким образом, достаточно рассчитать длину дуги эллипса при t € *[*0; *π/2]* и умножить на 4.



* 1. ***Спецификация переменных (обозначения, типы, назначение, классификация)***

*x* – текущая координата объекта, тип Single.

*y* – текущая координата объекта, тип Single.

*x2* – предыдущая координата объекта, тип Single.

*y2* – предыдущая координата объекта, тип Single.

*v* – мгновенная скорость объекта, тип Single.

*fi* – угол в радианах, который изменяется во время работы программы, тип Double.

*k* – переменная, хранящая количество кругов, пройденных объектом, тип Integer.

*S* – переменная, хранящая путь, пройденный телом, тип Single.

Исходные данные: *переменных, хранящих исходные данные, нет.*

Результаты: *x, y, v, k, S.*

Промежуточные результаты: *fi, x2, y2*.

* 1. ***Спецификация объектов (назначение, используемые свойства)***

**Form Form1 –** форма, на которой расположены другие объекты управления.

*AutoRedraw = -1 'True* (рисунки, выполненные графическими методами, автоматически восстанавливаются после изменения размера формы или после того как форма была закрыта другим окном)

*Caption = "Движение по эллиптическому маршруту с регулируемой скоростью и визуализацией"* (надпись в заголовке формы)

**PictureBox Picture1 –** внутри объекта происходит графическая визуализация процесса.

*AutoRedraw = -1 'True*

*ScaleMode = 3 'Pixel* (единицы измерения, используемые в системе координат, связанной с объектом)

Height = 3975 (высота объекта)

Width = 6255 (ширина объекта)

**Line Line1 ­–** объект, используемый для изображения оси ординат.

X1 = 20

X2 = 20

Y1 = 8

Y2 = 256

**Line Line2 –** объект, используемый для изображения оси абсцисс.

X1 = 408

X2 = 8

Y1 = 240

Y2 = 240

**Shape Shape1 –** объект, выводит изображение геометрических фигур, в программе это круг, перемещающийся по лини эллипса.

Shape = 3 'Circle

Width = 240

Height = 240

**Label Label9 –** метка, выводит на экран надпись. Располагается на объекте **Picture1.** Указывает начало координат.

Caption = "0"

**Label Label8 –** метка, выводит на экран надпись. Располагается на объекте **Picture1.** Подпись к оси ординат.

Caption = "Y,м"

**Label Label7 –** метка, выводит на экран надпись. Располагается на объекте **Picture1.** Подпись к оси абсцисс.

Caption = "X,м"

**Label Label13 –** метка, выводит на экран надпись. Располагается на объекте **Picture1.** Подпись к оси ординат.

Caption = "100"

**Label Label14 –** метка, выводит на экран надпись. Располагается на объекте **Picture1.** Подпись к оси абсцисс.

Caption = "50"

**Timer Timer1 –** позволяет организовать циклическое повторение группы команд, отвечающих за движение объекта **Shape1,** расчёт значений переменных x, y, k, S и их вывод в соответсвующие метки

Enabled = 0 'False (при загрузке формы выключен)

Interval = 1 (промежуток времени в миллисекундах, через который повторяется выполнение команд)

**Timer Timer2 –** позволяет организовать циклическое повторение группы команд, отвечающих за движение объекта **Shape1,** расчёт значений переменных x2, y2, v и их вывод в соответсвующие метки

Enabled = 0 'False (при загрузке формы выключен)

Interval = 100 (промежуток времени в миллисекундах, через который повторяется выполнение команд)

**CommandButton Command1 –** объект управления командная кнопка. Нажатие на кнопку начинает выпонение действий, программный код которых связан с ней, объект **Shape1** начинает движение.

Caption = "Старт"

Height = 495 (высота)

Width = 1455 (ширина)

**CommandButton Command2 –** объект управления командная кнопка. Нажатие на кнопку начинает выпонение действий, программный код которых связан с ней, объект **Shape1** прекращает движение.

Caption = "Стоп"

Height = 495 (высота)

Width = 1455 (ширина)

**Label Label12 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к объекту выводящему время с начала движения объекта **Shape1**.

Caption = "Общее время, с"

Left = 6720 (положение объекта, координата х верхнего левого угла объекта)

Top = 2880 (положение объекта, координата y верхнего левого угла объекта)

**Label Label5 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к объекту, выводящему путь, пройденный объектом **Shape1**.

Caption = "Общий путь, м"

Left = 6720

Top = 2400

**Label LabelY –** метка, в которую выводится координата Y. Располагается на объекте **Form1.**

BorderStyle = 1 'Fixed Single (задаёт рамку вокруг метки)

Left = 8880

Top = 1800

**Label LabelX**

BorderStyle = 1 'Fixed Single

Left = 8160

Top = 1800

**Label Label10 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к меткам, выводящим текущие координаты объекта **Shape1**.

Caption = "Координаты"

Left = 6720

Top = 1800

**Label Label6 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к метке, выводящей мгновенную скорость объекта **Shape1**.

Caption = "Скорость:"

Left = 480

Top = 5040

**LabelK –** метка, в которую выводится количество полных кругов. Располагается на объекте **Form1.**

Caption = "0"

Left = 8640

Top = 1200

**Label LabelV LabelK –** метка, в которую выводится количество полных кругов, пройденных объектом **Shape1**. Располагается на объекте **Form1.**

Caption = "0" (изначально количество кругов считается равным 0)

Left = 8400

Top = 3360

**Label Label3 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к метке, выводящей число кругов, пройденных объектом **Shape1.**

BorderStyle = 1 'Fixed Single

Caption = "Пройдено кругов"

Left = 6720

Top = 1200

**Label Label2 –** метка, выводит на экран надпись. Располагается на объекте **Form1.** Подпись к метке, выводящей скорость объекта **Shape1.**

Caption = "Скорость м/c"

Left = 6720

Top = 3360

**Label Label1 –** метка, выводит на экран надпись. Располагается на объекте **Form1.**

Caption = "Движение по эллиптическому маршруту с регулируемой скоростью и графической визуализацией процесса"

**Slider Slider1 –** объект управления, с помощью которого регулируется скорость движения объекта **Shape1.**

Max = 30 (максимальное число делений)

Value = 2 (деление на котором располагается бегунок по умолчанию)

* 1. ***Организация процесса и расчетные формулы***

Организовывается процесс работы программы следующим образом. При запуске программы появляется форма, на которой отображаются объекты управления. Для объекта **Picture1** свойство DrawWidth = 8, это толщина точек, при помощи которых на объекте изображается эллипс, прорисовка точек с использованием метода PSet происходит в теле цикла For…Next следующим образом: Picture1.PSet (180 \* Cos(i) + 200, 100 \* Sin(i) + 140)

В результате выполнения цикла получаем изображение эллипса, у которого большая полуось равна 90,а малая 50, и смещённым относительно левого верхнего угла объекта **Picture1** на 200 пикселей влево и 140 – вниз.

С помощью метода PSet рисуются точки на осях абсцисс и ординат, задающие единичный отрезок. Масштаб задан следующий 360 пикселей = 100 метров. Таким образом, для пользователя большая полуось будет составлять 45 метров, а меньшая – 50 метров.

При нажатии на командную кнопку **Command1** проверяется условие если Slider1.Value = 0, таймер, организующий движение объекта не включается (Timer1.Enabled = False) так как это означает, что скорость равна нулю. Во всех остальных случаях таймер включается (Timer1.Enabled = True).

При нажатии на командную кнопку **Command2** таймеры **Timer1** и **Timer2** выключаются Timer1.Enabled = False: Timer2.Enabled = False, что приводит к остановке движения тела и остановке отсчёта времени.

Теперь последовательно рассмотрим программный код процедур Timer1\_Timer() и Timer2\_Timer().

С помощью объекта **Timer1,** организуется движение объекта. В первую очередь активируется **Timer1.** Затем рассчитываются координаты объекта **Shape1** по формулам

x = 180 \* Cos(fi) + 200

y = 100 \* Sin(fi) + 140

Формулы совпадают с формулами, по которым происходила прорисовка эллипса, это обеспечивает то, что объект движется по траектории, совпадающей с эллипсом. После того как координаты рассчитаны, они присваиваются свойствам Left и Top объекта **Shape1,** определяющих положение объекта на объекте **Picture1.** Также координаты выводятся в соответствующие метки на форме LabelX.Caption = (x - 20) \ 4: LabelY.Caption = (240 - y) \ 4.

Переменная fi изменяется по закону fi = fi + (Slider1.Value / 2) \* 3.1415 / 180, т.е. угол разбиения, а следовательно и скорость объекта **Shape1**, зависит от свойства Value, объекта Slider1, иначе говоря от положения бегунка.

Число полных кругов, пройденных объектом **Shape1,** рассчитывается по формуле k = Int(fi / 6.28), и после расчёта выводится в соответствующую метку LabelK.Caption = k.

Пройденный телом путь, рассчитывается по формуле S = Round(223.112 / 6.28 \* fi).

Число 223.112 – это длина эллипса в метрах, рассчитана с помощью определённого интеграла:

4\*.



После расчёта значение выводится в соответствующую метку LabelS.Caption = S.

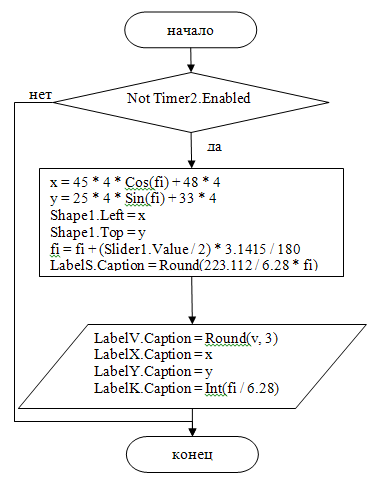
Свойство Interval объекта **Timer1**, равно 1, это значит, что все описанные выше действия повторяются 1 раз в миллисекунду. Величина интервала подобрана опытным путём.

В программном коде процедуры Timer2\_Timer() осуществляется подсчёт времени, по формуле LabelTime.Caption = LabelTime.Caption + 0.1,а свойство Interval объекта **Timer2** равно 100, что обеспечивает отсчёт времени в секундах, с точностью до десятых.

Скорость объекта **Shape1**, рассчитывается по формуле v = (Sqr((x2 - x) ^ 2 + (y2 - y) ^ 2)) / (Timer2.Interval) \* 1000. Расстояние между двумя соседними точками определяется как корень квадратный из суммы квадратов разностей координат этих точек, и делится на время, т.е интервал таймера **Timer1.**

***3. Разработка блок-схемы***

Представим графический алгоритм, в виде блок-схемы, для процедуры Private Sub Timer1\_Timer(), отвечающей за передвижение тела по эллиптической траектории.



***4. Текст программы на VB***

Dim fi As Double 'угол'

Dim x As Integer 'текущая координата Х

Dim y As Integer 'текущая координата Y

Dim x2 As Integer

Dim y2 As Integer

Dim v As Single 'скорость тела

Private Sub Command1\_Click()

If Slider1.Value = 0 Then Timer1.Enabled = False Else Timer1.Enabled = True 'при нажатии на СТАРТ если скорость установлена 0,тело не движется

End Sub

Private Sub Command2\_Click()

Timer1.Enabled = False 'кнопка СТОП.Движение тела и отсчёт времени прекращается.

Timer2.Enabled = False

End Sub

Private Sub Form\_Load()

Picture1.DrawWidth = 8 'устанавливается толщина линии 8 пикселей

For i = 0 To 360

Picture1.PSet (180 \* Cos(i) + 200, 100 \* Sin(i) + 140) 'при помощи метода PSet рисуется эллипс

Next

Shape1.Left = -20 'координаты объекта при загрузке формы,таковы,что объект отсутствует на форме

Shape1.Top = -20

fi = 0

Picture1.PSet (380, 240) 'точки на осях x и y,единичный отрезок

Picture1.PSet (20, 40)

End Sub

Private Sub Timer1\_Timer()

If Not Timer2.Enabled Then Timer2.Enabled = True

x = 45 \* 4 \* Cos(fi) + 48 \* 4 'траектория движения тела, координаиа Х

y = 25 \* 4 \* Sin(fi) + 33 \* 4 'Y

Shape1.Left = x

Shape1.Top = y

LabelV.Caption = Round(v, 3) 'вывод значения скорости в метку,точность до тысячных

LabelX.Caption = x 'вывод координат в метки

LabelY.Caption = y

LabelK.Caption = Int(fi / 6.28) 'количество кругов

LabelS.Caption = Round(223.112 / 6.28 \* fi) 'пройденный путь

fi = fi + (Slider1.Value / 2) \* 3.1415 / 180 'закон изменения угла разбиения

End Sub

Private Sub Timer2\_Timer()

LabelTime.Caption = LabelTime.Caption + 0.1 'время

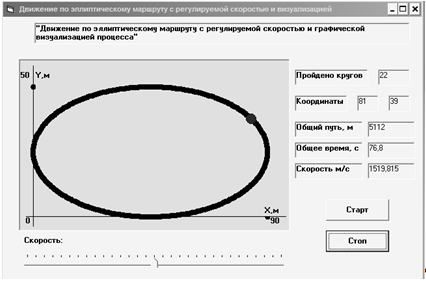
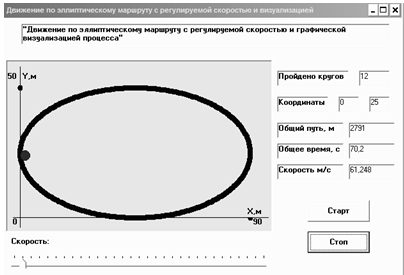
v = (Sqr((x2 - x) ^ 2 + (y2 - y) ^ 2)) / (Timer2.Interval) \* 1000 'скорость

x2 = x

y2 = y

End Sub

***5.******Контрольный пример***



***6. Инструкция пользователя***

Данная программа является графической визуализацией движения объекта по эллиптическому маршруту, с регулируемой скоростью.

Запустите файл ”Project1.exe”. На экране появится форма, большую её часть занимает рамка с изображением прямоугольной системы координат и изображённого в ней эллипса. При нажатии на кнопку «Старт», появляется объект в виде небольшого круга, движущегося по траектории, обозначенной эллипсом. В окошках, расположенных на форме слева указываются параметры движения тела: координаты, количество полных пройденных кругов, общий пройденный путь и скорость. Скорость тела регулируется с помощью бегунка, расположенного внизу формы. Процесс движения можно остановить, нажав на кнопку «Стоп», при повторном нажатии на «Старт» тело продолжит движение при этом характеристики движения не обнулятся, и их отсчёт будет продолжен с момента остановки. Для выхода из программы нажмите на кнопку с крестиком в верхнем правом углу окна.

***Список использованной литературы***

1. А.Г. Паутова. Visual Basic Творческое проектирование. В 3 ч. – М.: Классикс Стиль, 2003. – 128 с.
2. Марченко А.Л. «Введение в программирование»: Учебное пособие. - Московский университет, 2005. – 258 с.
3. Либерти Д. «Программирование на VB»: Пер. с англ. – 2-ое изд.- Сивол-Плюс, 2004. – 688 с.
4. Мультимедийное пособие «Программирование для студентов», Alex Soft.