

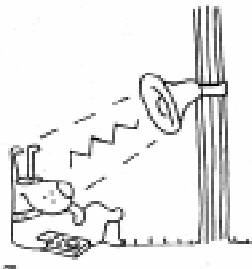
ЗАНЯТИЕ 2. РАБОТА С ФАЙЛАМИ И ГРАФИКОЙ В VISUAL BASIC

Если Вы после первого занятия заинтересовались Visual Basic и у Вас появилось желание, не дожидаясь очередных статей в журнале, идти вперед самостоятельно, то Вы можете найти интересующую Вас информацию по Visual Basic: о книгах, русскоязычных сайтах, телеконференциях, о примерах программирования и FAQ (часто задаваемые вопросы), посетив сайт <http://vbrussian.com>.

Прежде, чем продолжить изучение следующих объектов управления и контроля в Visual Basic, необходимо сделать замечание о назначении имен. Имена объектов так же, как и имена переменных, должны раскрывать основную суть этих объектов. Существует соглашение, по которому имя объекта состоит из префикса, указывающего непосредственно на объект, и основной части, раскрывающей функциональное назначение данного объекта. Например: *frmMain* – имя основной формы в программе, *cmdExit* – имя командной кнопки, предназначенной для выхода из программы, *lblText* – имя метки над текстовым окном, *hsbSpeed* – имя горизонтальной линейки прокрутки, используемой в качестве регулятора скорости, *vsbTemper* – имя вертикальной линейки прокрутки, используемой в качестве термометра и т.п.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ OPTION BUTTON

Option Button (Кнопка опций) – это кнопка-переключатель, иногда называемая также радиокнопкой. С ее помощью отображается устройство, индуцирующее включение или выключение чего-



либо. Обычно такие кнопки используют как часть некоторой группы или множества устройств, из которых пользователь может выбрать только одно. Когда пользователь выбирает кнопку-переключатель, другие кнопки из той же группы автоматически отключаются.

Основные свойства:

Наряду со стандартными свойствами *Alignment*, *BackColor*, *Caption*, *Font*, *Enabled*, *Left*, *Name*, *Top*, *Visible*, *Width* и *Height*, Кнопка опций обладает еще одним важным свойством:

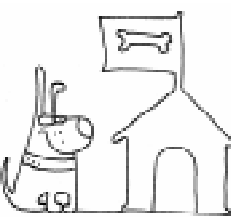
Value (значение) – при *true* кнопка активна, при *false* кнопка пассивна.

Основные события для Кнопки опций:

Click (Db1Click) – событие наступает, когда пользователь делает щелчок (или двойной щелчок) на объект кнопкой мыши.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ CHECK BOX

Check Box (Флажок) – окно контроля, отображающее информацию о включении или выключении каких-то операций в дальнейших действиях пользователя. В отличие от Кнопок опций, контрольных окон может быть выбрано любое число.



Основные свойства и события у Флажка такие же, как у Кнопки опций.

Упражнения (для самостоятельного выполнения):

1. Измените название формы «Form1» на «Свой-



ства Текстового окна», используя окно свойств.

2. Нанесите на форму Текстовое окно и занесите в него фразу «Пусть всегда будет солнце!»

3. Расположите на форме с левой стороны группу из трех Кнопок опций, подписав их, соответственно, «Цвет фона», «Цвет букв», «Размер шрифта».

4. Расположите на форме с правой стороны группу из трех Флажков, подписав их так же, как и Кнопки опций: «Цвет фона», «Цвет букв», «Размер шрифта».

5. В процедуры обработки события Click для первой Кнопки опций и первого Флажка занесите код, изменяющий цвет фона Текстового окна на другой.

6. В процедуры обработки события Click для второй Кнопки опций и второго Флажка занесите код, изменяющий цвет букв в Текстовом окне.

7. В процедуры обработки события Click для третьей Кнопки опций и третьего Флажка занесите код, изменяющий размер шрифта в Текстовом окне.

8. Проверьте действие свойства Value Кнопок опций.

9. Проверьте действие свойства Value Флажков.

Задание 1. Тест № 1.



1. Создайте форму, как на рисунке 1, переименуйте ее.

2. Нанесите на фор-

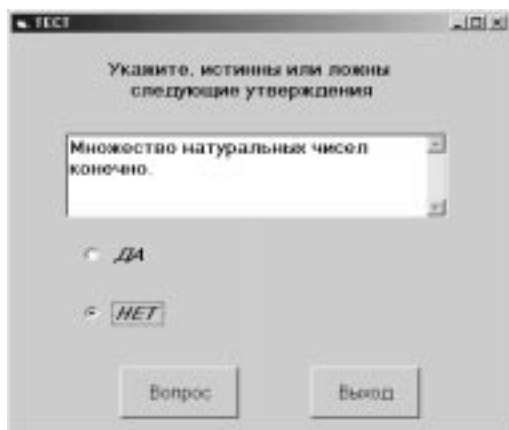


Рисунок 1

му Метку, Текстовое окно, две Кнопки Опций и две Командные Кнопки. Измените их свойства в соответствии с рисунком.

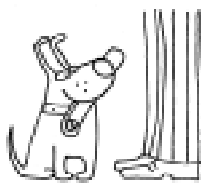
3. Напишите программу, выполняющую следующие функции:

- При нажатии на Кнопку «Вопрос» анализируется ответ ДА или НЕТ на предыдущий вопрос, считается количество правильных ответов, а в Текстовом окне появляется очередной вопрос теста. Когда все вопросы закончены, в Текстовом окне появляется сообщение о количестве правильных ответов;

- При нажатии на Кнопку «Выход» программа завершает свою работу.

Выполнение задания 1.

Первая часть – *визуальное программирование.*



1. Необходимо создать форму и переименовать ее в соответствии с рисунком 1, свойству

Caption (название) присвоив значение «ТЕСТ №1».

2. Нанести на форму метку, изменив свойство Caption на строку «Укажите, истинны или ложны следующие утверждения», свойство Font в соответствии с требованиями к размеру, цвету и начертанию букв, свойству Alignment присвоить значение 2 – Center.

3. Нанести на форму Текстовое окно, изменив свойство Text на пустую строку, свойству Multiline присвоив значение True, а свойству ScrollBars присвоив значение 2 – Vertical.

4. Нанести на форму Кнопки опций 1 и 2, изменив их размеры и свойство Font, а также свойство Caption на «ДА» и «НЕТ», соответственно.

5. Нанести на форму Командные кнопки 1 и 2, изменив их размеры и свойство Font, а также свойство Caption на «Вопрос» и «Выход» соответственно.

Вторая часть – *написание кода программы.*

6. В данном приложении будут использоваться переменная-счетчик k цело-

го типа, нумерующая вопросы, переменная `kp`, считающая количество правильных ответов, и переменная `n`, содержащая номер правильного ответа, поэтому требуется сделать их описание в общем разделе объявлений `General Declarations`.

```
Dim k, kp, n
```

7. Первое событие – загрузка формы. В процедуре обработки данного события необходимо подготовиться к работе с файлом, в котором содержится тест с вопросами и номерами правильных ответов, то есть открыть его. Предполагается, что файл текстовый, последовательного доступа, организован следующим образом: первая строка – вопрос; вторая строка – номер правильного ответа: 1, если правильный ответ ДА, или 2, если правильный ответ НЕТ.

```
Private Sub Form_Load()
    Open "test.txt" For Input As #1
End Sub
```

8. Второе событие после загрузки формы – щелчок на кнопку «Вопрос». В процедуре обработки данного события необходимо предусмотреть следующие действия:

- проверить номер правильного ответа и нажатие соответствующей Кнопки опций, в случае верного ответа увеличить счетчик правильных ответов на 1 (первая и вторая строки);
- увеличить счетчик вопросов на 1 (третья строка);
- кнопки опций сделать свободными для следующего нажатия (строки 4-я и 5-я);
- используя функцию проверки на конец файла `EOF` (строка 6), прочитать из файла строку вопроса и строку с номером ответа (строки 7 и 8) с помощью оператора `LINE INPUT`;
- занести в текстовое окно вопрос (строка 9);

- при обнаружении конца файла в текстовое окно вывести сообщение о количестве правильных ответов (строка 11) и сделать Кнопку «Вопрос» недоступной (строка 12).*

```
Private Sub CmdQuestion_Click()
    If n = 1 And OptYes.Value = True
        Then kp = kp + 1
    If n = 2 And OptNo.Value = True
        Then kp = kp + 1
    k = k + 1
    OptYes.Value = False
    OptNo.Value = False
    If Not (EOF(1)) Then
        Line Input #1, a$
        Line Input #1, n
        TxtQuestion.Text = a$
    Else
        TxtQuestion.Text = "Из" + Str$(k - 1) + "вопросов правильно_отвечено на" + Str$(kp)
        CmdQuestion.Enabled = False
    End If
End Sub
```

9. Следующее событие – щелчок на кнопку «Выход». В процедуре обработки данного события необходимо завершить программу оператором `END`, предварительно закрыв файл с помощью оператора `CLOSE`.

```
Private Sub CmdExit_Click()
    Close #1
End Sub
```

10. Запустить программу на выполнение, протестировать ее и завершить.

Задание 2. Тест № 2 (для самостоятельного выполнения).

1. Создайте форму, как на рисунке 2, переименуйте ее.

2. Нанесите на форму Метку, Текстовое окно, четыре Флажка и две Командные Кнопки. Измените их свойства в соответствии с рисунком.



* Поскольку ширина страницы книги не бесконечна, некоторые операторы программы разбиты на несколько строк. В конце таких строк используется символ подчеркивания (`_`), который является стандартным символом продолжения языка `Visual Basic`. При вводе текста в окно редактора вы вполне можете опустить символы продолжения и записать оператор в одной строке. Обе записи правомерны, и компилятор правильно их воспримет.

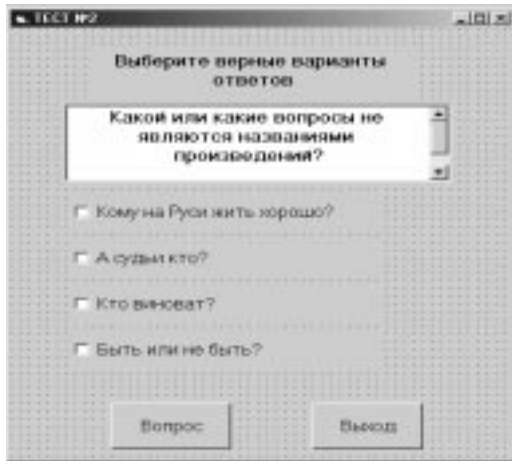
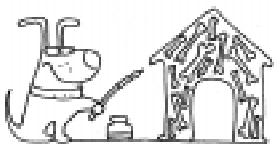


Рисунок 2

3. Напишите программу, выполняющую следующие функции:

- При нажатии на Кнопку «Вопрос» анализируются варианты ответа на предыдущий вопрос, считается количество правильных ответов (их может быть несколько), а в Текстовом окне появляется очередной вопрос теста. Когда все вопросы закончены, в Текстовом окне появляется оценка за тест.
- При нажатии на Кнопку «Выход» программа завершает свою работу.

ОБЪЕКТЫ УПРАВЛЕНИЯ И КОНТРОЛЯ PICTURE BOX И IMAGE



Picture Box (Окно рисунка) и *Image* (Окно изображений) позволяют размещать графическую информацию в определенных участках формы.

Окна рисунков – более гибкие объекты, но требуют большего объема памяти и времени на обработку. Они лучше подходят для динамической среды, когда при выполнении программы изображения рисуют непосредственно на экране или создают анимационные эффекты, перемещая значок по экрану.

Графические объекты Окна изображений удобнее использовать в статической среде, то есть в тех случаях, когда созданную или скопированную растровую картинку не предполагается изменять.

 – инструмент Picture Box

 – инструмент Image

Основные свойства:

Наряду со стандартными свойствами BackColor, BorderStyle, Enabled, Left, Name, Top, Visible, Width и Height, Окно рисунка и Окно изображения обладают еще одним важным свойством Picture (Картинка), которое позволяет выводить на экран либо растровые, либо векторные картинки.

Для его настройки при проектировании программы служит окно Properties, а в период выполнения – функция Loadpicture(). Необходимо помнить, что координаты объекта в случае свойств Height, Left, Top и Width относительны, а не абсолютны, то есть изменяются как смещение от начала координат формы, а не экрана.

У Окон рисунков, во многом похожих на небольшие формы внутри основной формы, есть ряд свойств, которых нет у Окна изображений: AutoRedraw, Fontbold, FontItalic, FontName, Fontsize, FontUnderline. Эти свойства аналогичны уже рассмотренным свойствам форм. А Окна изображений имеют одно дополнительное свойство, отсутствующее у Окон рисунков, Stretch (масштабирование). По умолчанию, оно равно False; это приводит к автоматической подгонке размеров графического объекта-элемента управления под содержащуюся в нем картинку. При изменении значения False на True картинка будет масштабироваться в соответствии с размерами Окна изображения.

Основные события для Окон рисунков и изображений:

Click (Db1Click) – событие наступает, когда пользователь делает щелчок (или двойной щелчок) на объект кнопкой мыши.

MouseDown, MouseUp – событие наступает, когда пользователь нажимает или отпускает левую кнопку мыши.

Процедуры и методы:

Функция LoadPicture (имя файла-картинки) устанавливает свойство Picture Окна рисунка или изображения.

Например, Picture1.Picture = LoadPicture («cloud.ico») загружает картинку с облаком в Окно рисунка.

Кроме того, Окно рисунка поддерживает методы Print, Cls, Scale (масштабирование), Pset, Line и прочие графические методы.

Задание 3. Цветовая палитра.



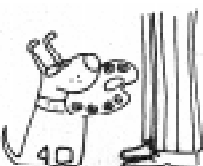
1. Создайте форму, как на рисунке 3, переименуйте ее.

2. Нанесите на форму Окно рисунка и три Кнопки опций. Измените их свойства в соответствии с рисунком.

3. Напишите программу, выполняющую следующие функции:

При нажатии на одну из Кнопок опций в Окне рисунка появляется изображение палитры соответствующего цвета. Программа завершает работу при нажатии на кнопку «Закреть» системного меню.

Выполнение задания 3.



Первая часть – *визуальное программирование.*

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 3, свойству Caption (название) присвоив значение «Цветовая палитра».

2. Нанести на форму Окно рисунка.

3. Нанести на форму Кнопки опций 1, 2 и 3, изменив их свойство Caption на «Красный», «Зеленый» и «Синий», соответственно.

Вторая часть – *написание кода программы.*

4. В данном приложении не будут использоваться никакие глобальные переменные, следовательно, нет описаний в общем разделе объявлений.

5. При загрузке формы не происходит никаких событий.

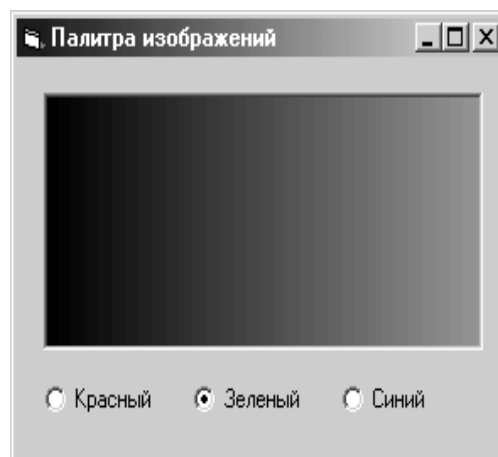


Рисунок 3

6. Событие после загрузки формы – щелчок на одну из Кнопок опций. В процедуре обработки данного события необходимо предусмотреть следующее действие: нарисовать палитру соответствующего цвета. Каким образом это можно осуществить?

- Создать общую процедуру DrawPalette, которая, в зависимости от входного параметра – цвета Base, заполняет окно рисунка линиями, используя метод Line и постепенно увеличивая номера цветов.

```
Sub DrawPalette(Base)
```

```
For i = 0 To 255  
pic.ForeColor = i * Base  
pic.Line (i, 0)-(i, pic.Height)  
Next  
End Sub
```

- При нажатии на Кнопку опций вызвать процедуру DrawPalette с соответствующим номером цвета: красным (&H1), зеленым (&H100) или синим (&H10000).

```
Private Sub optBlue_Click()  
Call DrawPalette(&H10000)  
End Sub
```

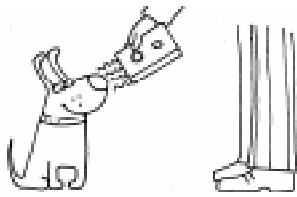
```
Private Sub optGreen_Click()  
Call DrawPalette(&H100)  
End Sub
```

```
Private Sub optRed_Click()  
Call DrawPalette(&H1)  
End Sub
```

- Запустить программу на выполнение, протестировать ее и завершить.

Задание 4.

Просмотрщик изображений.



1. Создайте форму, как на рисунке 4, переименуйте ее.

2. Нанесите на форму Окно изображения и три Кнопки опций. Измените их свойства в соответствии с рисунком.

3. Напишите программу, выполняющую следующие функции:

При нажатии на одну из Кнопок опций в Окне рисунка появляется рисунок в соответствующем графическом формате. Причем, если это иконка формата .ICO, размер ее увеличивается до размеров Окна изображения, если это фотография формата .JPG или рисунок .BMP, размер их уменьшается до размеров Окна. Программа завершает работу при нажатии на кнопку «Закреть» системного меню.

Выполнение задания 4.



Первая часть – *визуальное программирование.*

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 4, свойству Caption (название) присвоив значение «Просмотр изображений».

2. Нанести на форму Окно изображения, свойству Stretch Окна изображения присвоить значение True для возможности масштабирования рисунков.

3. Нанести на форму Кнопки опций 1, 2 и 3, изменив их свойство Caption на «BMP», «ICO» и «JPG», соответственно.

Вторая часть – *написание кода программы.*

4. В данном приложении не будут использоваться никакие глобальные переменные, следовательно, нет описаний в общем разделе объявлений.

5. При загрузке формы можно сделать активной любую из Кнопок опций, например, третью.



Рисунок 4

```
Private Sub Form_Load()  
Option3.Value = True  
End Sub
```

6. Событие после загрузки формы – щелчок на одну из Кнопок опций. В процедуре обработки данного события необходимо предусмотреть следующее действие: загрузить рисунок из файла соответствующего формата с помощью функции LoadPicture().

```
Private Sub Option1_Click()  
img.Picture =  
LoadPicture («earth.bmp»)  
End Sub
```

```
Private Sub Option2_Click()  
img.Picture =  
LoadPicture («skyline.ico»)  
End Sub
```

```
Private Sub Option3_Click()  
img.Picture =  
LoadPicture («flowers.jpg»)  
End Sub
```

Задание 5. Число ПИ.



1. Создайте форму, как на рисунке 5, переименуйте ее.

2. Нанесите на форму Окно рисунка, нарисуйте в нем квадрат и впишите в него окружность, используя соответствующие методы Окна рисунка.

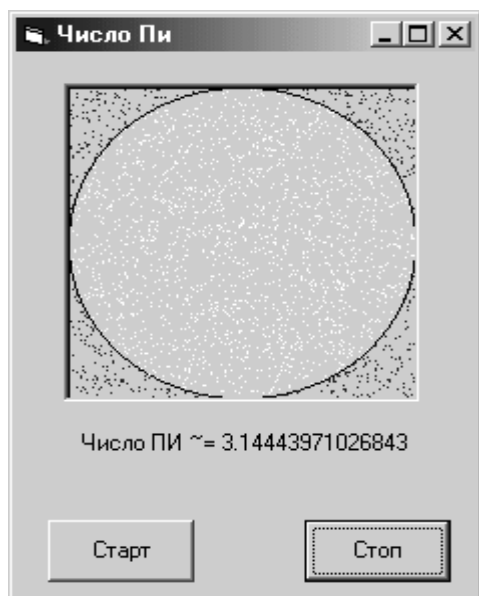


Рисунок 5

3. Нанесите на форму метку для вывода сообщения-результата.

4. Нанесите на форму две Командные кнопки, изменив у них свойство Caption.

5. Напишите программу, вычисляющую число ПИ с помощью метода Монте-Карло и демонстрирующую графически данный метод.

6. Нанесите на форму Таймер.

7. Реализуйте следующие функции:

- При нажатии на кнопку «Старт» квадрат начинает заполняться случайными точками, причем точки, попавшие внутрь круга, изображаются другим цветом.
- При нажатии на кнопку «Стоп» заполнение точками прекращается и вычисляется число ПИ как отношение точек, попавших внутрь круга, к общему количеству точек, попавших в квадрат.
- Выводится значение числа ПИ в метку.

Выполнение задания 5.



Прежде, чем приступить к выполнению данного задания, необходимо немного рассказать о методе Монте-Карло. Метод статистического моделирования, или метод Монте-Карло, назван так в честь столицы кня-

жества Монако, известной своими многочисленными казино, в которых публика растрчивает или увеличивает свои доходы, согласно законам распределения случайных величин.

Данный метод используется для моделирования игровых вероятностных ситуаций (бросание монеты, кубика, блуждания), а также для вычисления площадей фигур неопределенной формы. Для того, чтобы определить площадь фигуры, необходимо вписать ее в геометрическую фигуру, площадь которой легко вычисляется (например, в прямоугольник). Затем бросать в этот прямоугольник случайные точки, подсчитывая количество тех из них, которые попали внутрь фигуры. Произведение площади прямоугольника на отношение количества точек, попавших внутрь фигуры, ко всему количеству точек и дает приближенное значение искомой площади.

Первая часть – *визуальное программирование.*

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 5, свойству Caption (название) присвоив значение «Число ПИ».

2. Нанести на форму Окно рисунка.

3. Нанести на форму метку, изменив ее свойства: Alignment на 2 – Center, Autosize на True, Caption на пустую строку.

4. Нанести на форму Командные кнопки, изменив их свойство Caption на «Старт» и «Стоп».

5. Нанести на форму объект Таймер, присвоив свойству интервал значение 1 (что означает 1 миллисекунда) и Enabled – False.

Вторая часть – *написание кода программы.*

6. В данном приложении будут использоваться переменные, обозначающие количество точек всего, число точек, попавших внутрь круга, число ПИ, следовательно, их следует описать в общем разделе объявлений.

Dim n, m, pi

7. При обработке события Загрузка Формы необходимо включить генератор случайных чисел и изменить масштаб Окна рисунка с помощью метода Scale, указав реальные координаты левого нижнего угла и правого верхнего.

```
Private Sub Form_Load()
    Randomize Timer
    Pic.Scale (-1, -1)-(1, 1)
End Sub
```

8. При обработке события Щелчок на Кнопку «Старт», необходимо запустить Таймер, изменив его Enabled на True. В окне рисунка, используя методы Line и Circle, нарисовать квадрат со стороной 2 и окружность единичного радиуса, вписанную в данный квадрат. Площадь такого квадрата равна 4 кв.ед., а круга – 3,1415926, то есть ПИ.

```
Private Sub Command1_Click()
    Timer1.Enabled = True
    Pic.Line (-1, -1)-(1, 1), , B
    Pic.Circle (0, 0), 1, , , 0.95
End Sub
```

9. Каждую миллисекунду срабатывает событие Timer, в результате которого должна быть «брошена» в квадрат случайная точка. Таким образом, счетчик точек увеличивается на 1, формируются случайные (внутри квадрата) координаты точки и проверяется принадлежность этой точки вписанному в квадрат кругу. Если точка попадает в круг, то увеличивается на 1 другой счетчик. Точки внутри и за пределами круга для наглядности изображаются разным цветом.

```
Private Sub Timer1_Timer()
    n = n + 1
    x = Rnd * 2 - 1
    y = Rnd * 2 - 1
    If (x * x + y * y) <= 1 Then
        m = m + 1
        Pic.PSet (x, y), QBColor(15)
    Else
        Pic.PSet (x, y), QBColor(4)
    End If
End Sub
```

10. Следующее событие – щелчок на Командную кнопку «Стоп». В проце-

дуре обработки данного события необходимо выключить Таймер, вычислить приближенное значение числа ПИ (площади круга единичного радиуса) из соотношений: Общее количество точек n соответствует площади квадрата.

Количество точек внутри круга m соответствует площади круга.

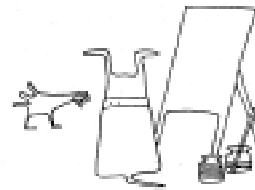
Отсюда следует формула для вычисления ПИ: $(m/n)*4$. Полученный результат представить в метке, изменив ее свойство Caption.

```
Private Sub Command2_Click()
    Timer1.Enabled = False
    pi = m * 4 / n
    Label1.Caption = "Число ПИ ≈ " + Str(pi)
End Sub
```

Следующее задание можно считать контрольной работой по всему ранее изложенному материалу.

Задание 6.

Контрольная работа № 1. Вариант 1.



1. Создайте форму, как на рисунке 6.

2. При запуске программы Окна текста и изображения недоступны.

3. При нажатии на кнопку «Ввести текст» в Текстовом окне появляется фраза: «Ввести текст из файла (1) или с клавиатуры (2)?»



Рисунок 6

4. При выборе Переключателя 1 текст вводится в Текстовое окно из готового файла, при выборе Переключателя 2 – с клавиатуры.

5. При нажатии на кнопку «Отобразить рисунок» в Окне изображения появляется рисунок или фотография.

6. При перемещении движка линейки прокрутки вправо очищается Текстовое окно, влево – Окно изображения.

7. При нажатии на кнопку «Выход» программа завершает работу.

Выполнение задания 6.



Первая часть – визуальное программирование.

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 6, свойству Caption (название) присвоив значение «Контрольная работа № 1. Вариант 1».

2. Нанести на форму две метки, изменив их свойство Caption на строки «Окно изображения» и «Текстовое окно», соответственно, свойство Font в соответствии с требованиями к размеру, цвету и начертанию букв, свойству Alignment присвоить значение 2 – Center.

3. Нанести на форму Текстовое окно, изменив свойство Text на пустую строку, свойству Multiline присвоив значение True, а свойству ScrollBars присвоив значение 2 – Vertical.

4. Нанести на форму Окно изображения, свойству Stretch Окна изображения присвоить значение True для возможности масштабирования рисунков

5. Нанести на форму Кнопки опций 1 и 2.

6. Нанести на форму горизонтальную линейку прокрутки, присвоив свойствам Min, Max и Value значения –1, 1 и 0, соответственно.

7. Нанести на форму Командные кнопки 1, 2 и 3, изменив их свойство Caption на «Ввести текст», «Отобразить рисунок» и «Выход», соответственно.

Вторая часть – написание кода программы.

8. В данном приложении не будут использоваться никакие глобальные переменные, следовательно, нет описаний в общем разделе объявлений.

9. Первое событие – загрузка формы. По условию задания, необходимо сделать недоступными Окно изображения и Текстовое окно, изменив свойство Enabled на False.

```
Private Sub Form_Load()  
    Text1.Enabled = False  
    Image1.Enabled = False  
End Sub
```

10. Второе событие после загрузки формы – щелчок на кнопку «Ввести текст». В процедуре обработки данного события необходимо предусмотреть следующее действие: вывести в текстовое окно сообщение «Ввести текст из файла (1) или с клавиатуры (2)?»

```
Private Sub Command1_Click()  
    Text1.Text = « Ввести текст из  
    файла (1) или с клавиатуры (2)?»  
End Sub
```

11. Следующее событие – щелчок на кнопку «Отобразить рисунок». В процедуре обработки данного события необходимо в окно изображения загрузить рисунок из файла с помощью функции LoadPicture().

```
Private Sub Command2_Click()  
    Image1.Picture =  
    LoadPicture («kr1.jpg»)  
End Sub
```

12. Следующее событие – щелчок на Переключатель 1. В процедуре обработки данного события необходимо в Текстовое окно загрузить текст из файла. Для этого необходимо:

- Открыть для чтения текстовый файл с помощью оператора OPEN.
- Очистить Текстовое окно, присвоив свойству Text пустую строку.
- В цикле WHILE ... WEND, пока не конец файла, считывать текст по одной строке с помощью оператора LINE INPUT и выводить в Текстовое окно с учетом перехода на другую строку (символ CR –

Chr(13) – перевод каретки и символ LF – Chr(10) – переход на другую строку).

• Закрывать файл.

```
Private Sub Option1_Click()  
    Open "kr_1.txt" For Input As #1  
    Text1.Text = ""  
    While Not EOF(1)  
        Line Input #1, a$  
        Text1.Text = Text1.Text + _  
            Chr(13) + Chr(10) + a$  
    Wend  
    Close #1  
    Text1.Enabled = True  
End Sub
```

13. Следующее событие – щелчок на Переключатель 2. В процедуре обработки данного события необходимо подготовить Текстовое окно для ввода с клавиатуры – сделать его доступным и очистить.

```
Private Sub Option2_Click()  
    Text1.Enabled = True  
    Text1.Text = ""  
End Sub
```

14. Следующее событие – это перемещение движка Линейки прокрутки. Проверку на перемещение вправо или влево легко осуществить, сравнивая текущее положение движка Value с нулем, так как нуль – середина Линейки прокрутки. Функция LoadPicture (), вызванная без параметра, очищает Окно изображения или рисунка.

```
Private Sub HScroll1_Change()  
    If HScroll1.Value < 0 Then  
        Image1.Picture = LoadPicture()  
    ElseIf HScroll1.Value > 0 Then  
        Text1.Text = ""  
    End If  
End Sub
```

15. Следующее событие – щелчок на кнопку «Выход». В процедуре обработки данного события необходимо завершить программу оператором END .

```
Private Sub Command3_Click()  
    End  
End Sub
```

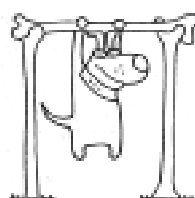
16. Запустить программу на выполнение, протестировать ее и завершить.



Рисунок 7

Задание 7.

Контрольная работа № 1. Вариант 2. (для самостоятельного выполнения)



1. Создайте форму, как на рисунке 7.

2. При запуске программы Окна метки и рисунка недоступны.

3. При нажатии на кнопку «Ввести текст» в Окне метки появляется фраза «Программа приветствует Вас!»

4. При нажатии на кнопку «Отобразить рисунок» в Окне метки появляется фраза «Готовый рисунок из файла (1) или возможность рисования (2)?».

5. При отметке Флажка 1 в Окне рисунка появляется рисунок, загруженный из файла; при выборе Флажка 2 в Окне рисунка появляется окружность.

6. При щелчке на Окно метки оно очищается, и движок Линейки прокрутки перемещается вверх, при щелчке на Окно рисунка оно очищается, и движок Линейки прокрутки перемещается вниз.

7. При нажатии на кнопку «Выход» программа завершает работу.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ LINE

Line (Линия) – это объект, позволяющий создавать простые прямые линии. Модифицируя свойства линий, можно изменять их размер, цвет и стиль.



Основные свойства:

Наряду со стандартными свойствами Name и Visible, Линия обладает еще рядом свойств, описывающих внешний вид этого объекта на экране:

BorderColor (Цвет границы) – определяет цвет линии.

BorderStyle (Тип границы) – определяет вид линии: 0 – Transparent (прозрачный); 1 – Solid (сплошной); 2 – Dash (пунктир); 3 – Dot (точечный); 4 – Dash-Dot (точка-тире); 5 – Dash-Dot-Dot (длинное-короткое типе); 6 – Inside Solid (сплошной внутри).

BorderWidth (ширина границы) – это свойство позволяет регулировать толщину линии, но только для сплошного типа границы.

Основные события:

Объект Линия не реагирует ни на какие события.

ОБЪЕКТ УПРАВЛЕНИЯ И КОНТРОЛЯ SHAPE



Shape (Фигура) – это объект, позволяющий создавать прямоугольники, квадраты, окружности, эллипсы и прямоугольники со скругленными углами. Допускается подбор цвета и закрашивание замкнутых фигур.

Основные свойства:

Наряду со стандартными свойствами Name, Visible, Height, Left, Top и Width Фигура, как и Линия, обладает еще рядом свойств, таких как: BorderColor, BorderStyle, BorderWidth. А кроме этого, у фигуры есть еще и дополнительные свойства:

FillColor (Цвет заполнения) – позволяет с помощью палитры цветов установить цвет внутренней части фигуры;

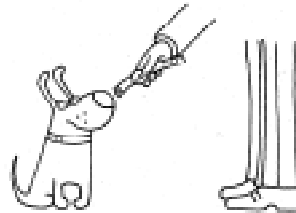
FillStyle (Стиль заполнения) – определяет шаблон заполнения внутренней части

фигуры: 0 – Solid (Сплошное), 1 – Transparent (Прозрачное), 2 – Horizontal Line (Горизонтальными линиями), 3 – Vertical Line (Вертикальными линиями), 4 – Upward Diagonal (Диагональными линиями слева направо), 5 – Downward Diagonal (Диагональными линиями справа налево), 6 – Cross (В клеточку), 7 – Diagonal Cross (В косую клеточку).

Основные события:

Объект Фигура не реагирует ни на какие события.

Задание 8. Часы.



1. Создайте форму, как на рисунке 8, переименуйте ее.

2. Нарисуйте часы, используя следующие объекты управления и контроля:

Фигура для корпуса часов, три объекта Линия для часовой, минутной и секундной стрелок, четыре метки для цифр на циферблате (3, 6, 9, 12).

3. Нанесите на форму два Переключателя, используемые для выбора формы часов – круглые или квадратные.

4. Напишите программу, выполняющую следующие функции:

- При запуске программы часы показывают текущее время и «идут», то есть

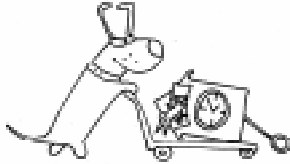


Рисунок 8

стрелки передвигаются в соответствии с реальным временем.

- При выборе соответствующей Кнопки опций изменяется форма часов.

Выполнение задания 8.



Первая часть – визуальное программирование.

1. Необходимо создать форму и переименовать ее в соответствии с рисунком 8, свойству Caption (название) присвоив значение «Часы».

2. Нанести на форму Фигуру, изменив ее свойства FillColor и Shape.

3. Нанести на форму три Линии, изображающие часовую, минутную и секундную стрелки, изменив их свойства X1, X2, Y1, Y2, BorderColor и BorderWidth.

4. Нанести на форму пять меток, изменив их свойство Caption на строки «3», «6», «9», «12» и «Форма часов».

5. Нанести на форму Кнопки опций 1 и 2, изменив их свойство Caption на «Круглые» и «Квадратные».

6. Нанести на форму объект Таймер, присвоив свойству интервал значение 1000 (что означает 1 сек).

Вторая часть – написание кода программы.

7. В данном приложении будут использоваться константы, обозначающие число «пи», «пи пополам» и т.п., а также константы, обозначающие длину всех трех стрелок часов в twip, следовательно, их следует описать в общем разделе объявлений.

```
Const Pi = 3.1416
Const Pi2 = Pi/2
Const cHour = Pi/6
Const cMinute = Pi/30
Const hLen = 500
Const mLen = 700
Const sLen = 900
```

8. Каждую секунду срабатывает событие Timer, в результате которого должны перемещаться стрелки на часах. Функция Time возвращает системное время,

функциями Hour, Minute, Second выделяются из системного времени часы, минуты и секунды, соответственно, переводятся в радианы, начальный угол отсчета смещается на угол $-\pi / 2$ (там, где 12 часов). После этого вычисляются координаты второго конца стрелок (известна длина стрелок и углы поворота), и их значения присваиваются свойствам X2 и Y2 объектов Линия.

```
Private Sub Timer1_Timer()
    Beep
    t = Time
    hh = Hour(t) * cHour - Pi2
    mm = Minute(t) * cMinute - Pi2
    ss = Second(t) * cMinute - Pi2
    lnHour.X2 = lnHour.X1 + _
        hLen * Cos(hh)
    lnHour.Y2 = lnHour.Y1 + _
        hLen * Sin(hh)
    lnMinute.X2 = lnMinute.X1 + _
        mLen * Cos(mm)
    lnMinute.Y2 = lnMinute.Y1 + _
        mLen * Sin(mm)
    lnSecond.X2 = lnSecond.X1 + _
        sLen * Cos(ss)
    lnSecond.Y2 = lnSecond.Y1 + _
        sLen * Sin(ss)
End Sub
```

9. Следующее событие – щелчок на Переключатель 1 или Переключатель 2. В процедурах обработки данных событий необходимо изменить свойство Shape либо на 3 (Circle), либо на 5 (Rounded Square).

```
Private Sub Option1_Click()
    Clock.Shape = 3
End Sub

Private Sub Option2_Click()
    Clock.Shape = 5
End Sub
```

Задание 9. Заставка

(для самостоятельного выполнения).

1. Создайте форму, как на рисунке 9, переименуйте ее.

2. Нанесите на форму Окно рисунка, являющееся недоступным.



3. Нанесите на форму три Командные кнопки, переименуйте их.



Рисунок 9

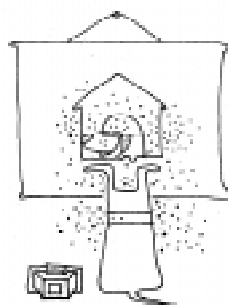
4. Напишите программу, выполняющую следующие функции:

- При нажатии на Кнопку «Броуновское движение» в Окне рисунка появляется возможность с помощью мыши нанести произвольное количество точек-кружочков, которые начинают хаотично двигаться.

- При нажатии на Кнопку «Вращающиеся линии» в Окне рисунка появляется некоторое количество разноцветных отрезков, которые вращаются и меняют цвет.
- При нажатии на Кнопку «СТОП» прекращается движение.

Задание 10. Слайды

(для самостоятельного выполнения).



1. Создайте форму, как на рисунке 4, переименуйте ее в соответствии с новым заданием.

2. Нанесите на форму Командную кнопку «Показать слайды», при нажатии на которую в Окне

изображения с интервалом в полсекунды должны меняться картинки. Повторное нажатие прекращает показ слайдов.

Указание: Для реализации данного задания используйте массив с названиями файлов картинок и объект Таймер.

*Паньгина Нина Николаевна,
преподаватель ОИиВТ
школы-лицея № 8, г. Сосновый Бор*



Наши авторы, 2001.
Our authors, 2001.