**Урок-лекция**

**Предмет - ИКТ – 9 класс (2ч)**

**Тема: Среда программирования Visual Basic. Окна, события.**

**Цель:** знакомство со средой программирования Visual Basic на примерах решения конкретных задач, дополняя основное решение возможными визуальными эффектами на форме. Предварительно познакомить учащихся с основными терминами и элементами ООП, повторить основные алгоритмические структуры и рассмотреть правила их описания на ООП Visual Basic, составить блок–схемы решения задач, описываемых основными алгоритмическими конструкциями.

**Ход урока**

1. Организационный момент
2. Изучение нового материала в виде лекции, сопровождаемой презентацией

**Объектно-ориентированное программирование (ООП)** — это метод программирования, при использовании которого главными элементами программ являются объекты. В языках программирования понятие объекта реализовано как совокупность свойств (структур данных, характерных для данного объекта), методов их обработки (подпрограмм изменения их свойств) и событий, на которые данный объект может реагировать и, которые приводят, как правило, к изменению свойств объекта. Объединение данных и свойственных им процедур обработки в одном объекте, называется **инкапсуляцией** и является одним из важнейших принципов ООП.

Другим фундаментальным понятием является класс. **Класс** — это шаблон, на основе которого может быть создан конкретный программный объект, он описывает свойства и методы, определяющие поведение объектов этого класса. Каждый конкретный объект, имеющий структуру этого класса, называется экземпляром класса.

Важнейшими принципами ООП являются наследование и полиморфизм. **Наследование** предусматривает создание новых классов на базе существующих и позволяет классу потомку иметь (наследовать) все свойства класса – родителя.

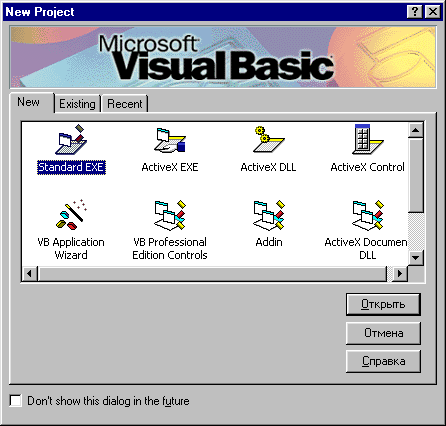
**Полиморфизм** означает, что рожденные объекты обладают информацией о том, какие методы они должны использовать в зависимости от того, в каком месте цепочки они находятся.

Другим важнейшим принципом ООП является **модульность** – объекты заключают в себе полное определение их характеристик, никакие определения методов и свойств не должны располагаться вне его, это делает возможным свободное копирование и внедрение одного объекта в другие.

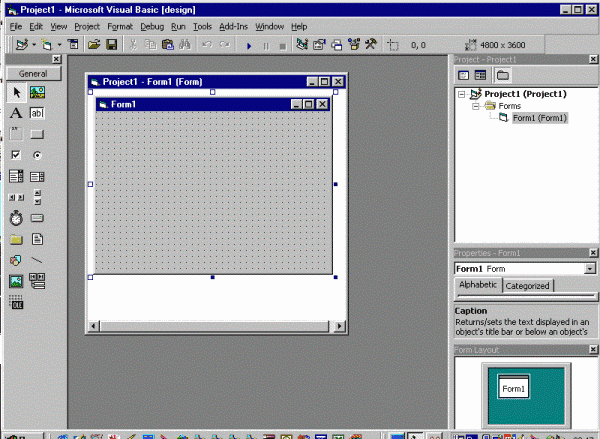
С середины 90-х годов многие объектно–ориентированные языки реализуются как системы визуального проектирования, в которых интерфейсная часть программного продукта создается в диалоговом режиме, практически без написания программных операторов. К объектно–ориентированным системам визуального проектирования относятся [Visual Basic](http://bourabai.kz/einf/vb5/index.htm), [Delphi](http://bourabai.kz/einf/Delphi/index.htm), [C++ Builder](http://bourabai.kz/C-Builder/index.htm), Visual C++.

    Сегодня на уроке Вы начнете знакомиться со средой программирования Visual Basic .

При входе в Visual Basic на экране появится окно. (слайд1)



    Выберите вариант получения в Visual Basic программы в режиме "Standard.exe". Далее подтвердите сделанный выбор, нажав на кнопку "Открыть" и вы увидите следующее окно



В появившемся окне 1-ая строка содержит наименование решаемой Вами задачи (или создаваемой программы), 2-ая строка - меню команд, 3-ья строка - стандартную панель инструментов. Ниже следует рабочее поле Visual Basic. В левой части рабочего поля - окно инструментов, реализующее доступ к инструментальным средствам Visual Basic.

Создание приложения в Visual Basic включает следующие два базовых шага:  
1). Построение пользовательского интерфейса приложения.  
2). Написание программного кода, который отвечает на действия, предпринимаемые в пользовательском интерфейсе.

**Первый шаг** при создании приложения заключается в проектировании форм, которые будут использоваться в приложении. А также в проектировании меню и панелей инструментов. Обычно пользователям нравятся такие приложения, которые просты в освоении и ощутимо повышают продуктивность их труда. Поэтому проектировать формы нужно очень тщательно, ибо это главный компонент интерфейса. Нужно все время взаимодействовать с пользователем, чтобы найти приемлемую и оптимальную для него компоновку форм и интерфейса.   
    Используя окно инструментов, разработчик программы рисует или помещает в форму элементы управления, создавая визуальные компоненты приложения. Рассматриваемая нами панель элементов (окно инструментов) содержит встроенные элементы управления Visual Basic и подключенные элементы ActiveX или любые внедряемые объекты, которые разработчик добавил к проекту.   
    Если панель элементов закрыта, то есть отсутствует на рабочем поле окна программы Visual Basic, то ее можно открыть. Для этого нужно выбрать команду ToolBox в меню View или щелкнуть соответствующую кнопку на стандартной панели инструментов (вторая строка окна программы Visual Basic). Данная панель инструментов (окно инструментов) доступна только во время разработки программы.

    Чтобы построить первое приложение в Visual Basic, разработчику потребуется всего несколько минут. Он создает интерфейс пользователя, "рисуя" элементы управления, например, текстовые поля или кнопки управления, на форме. Далее он устанавливает свойства для формы и для элементов управления, чтобы задать, например, цвет и размер. Затем он пишет программный код на языке Visual Basic, и приложение начинает работать. В дальнейшем на последующих уроках мы рассмотрим основные шаги, которые нужно выполнять при создании Вашего первого приложения. Эти шаги будут означать необходимые принципы, которым надо следовать при разработке Вашего приложения. 

|  |  |
| --- | --- |
| http://ipg.h1.ru/lessons/basic_v/les02.files/win_contents01.gif | http://ipg.h1.ru/lessons/basic_v/les02.files/win_form1.gif |

**Окно (windows)** - это прямоугольная область со своими собственными границами. Например, окно Проводника в Windows или окно текстового редактора Word или окно табличного процессора Excel или, наконец, окно программы Microsoft Paint.   
  
    Это наиболее известные примеры. На самом же деле в Windows существует много всевозможных различных типов окон. Та же кнопка управления или кнопка раздела "Файл" во второй строке любого из выше названных окон тоже являются окнами. Значки, текстовые поля, переключатели (option buttons) и меню - все они являются окнами.   
  
    Операционная система Windows управляет всеми этими типами окон. Каждому типу окон назначается уникальный идентификационный номер (id number), (windows handle или hWnd). Система постоянно следит за каждым окном, отмечает его активность и регистрирует события в нем.   
  
    **События** - возникают в результате действий пользователя. Например, когда он щелкает мышкой или нажимает клавишу на клавиатуре. Еще событие может возникать в ответ на программное управление или как результат действия другого окна.   
  
    **Сообщение** - это ответ системы на некоторое событие. Сообщение посылается операционной системе. Система обрабатывает это сообщение и передает его другим окнам. Каждое окно затем может предпринять некоторое действие. Такое действие основано на собственных инструкциях окна относительно этого вполне конкретного сообщения. Например, окно может снова нарисоваться на экране из под закрывавших его окон.   
  
Вы чувствуете, как на самом деле сложно управлять всеми возможными комбинациями взаимодействия окон, событий и сообщений. Но Visual Basic просто освобождает программиста от подобных проблем, связанных с обработкой всех возникающих сообщений нижнего уровня. Для удобства пользователя одни из сообщений автоматически обрабатываются Visual Basic, другие же распознаются и обрабатываются **процедурами обработки событий** (events procedures). Все это позволяет быстро создавать мощные, яркие приложения, не обращая внимания на несущественные детали.

**Рассмотрим основные алгоритмические структуры на конкретных примерах решения задач.**

1. **Моделирование алгоритмов линейной структуры**

Составить блок-схему и написать программу на языке Visual Basic линейной структуры.

Задача. С клавиатуры вводится время Т секунд. Скорость страуса 22 м/с, а скорость скворца 20 м/с. Какой путь проделает за это время страус? Скворец? На сколько метров путь страуса окажется больше пути скворца за время Т ?

РЕШЕНИЕ. Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

Т

V1:=22

V2:=20

S1:=v1\*T

S2:=v2\*T

S:=S1-S2

S1, S2, S

**Блок-схема Программный код задачи:**

Private Sub Command2\_Click()

Dim t, v1, v2, s1, s2, s As Integer

t = InputBox("введите время в секундах t=")

v1 = 22

v2 = 20

s1 = v1 \* t

s2 = v2 \* t

s = s1 - s2

Print "Путь страуса "; s1; "м"

Print "Путь скворца "; s2; "м"

Print "Путь страуса за "; t; " сек больше пути скворца на "; s; "м"

End Sub

Private Sub Command1\_Click()

End

End Sub

1. **Моделирование алгоритмов разветвленной структуры**

Составить блок-схему и написать программу на языке Visual Basic разветвляющейся структуры, с использованием оператора IF.

Задача. С клавиатуры вводится х.

Вычислить F(x) = 1,2х2 - 3х -9, если х >3,

, если х ≤ 3.

РЕШЕНИЕ. Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

**Блок-схема Программный код задачи:**

Private Sub Command1\_Click()

X

F:= 1,2\*х\*x– 3\*х -9

F

F:=12,1/()



X> 3

\_

+

x = InputBox("введите число х=")

If x > 3 Then f = 1.2 \* x \* x - 3 \* x - 9 Else f = 12.1 / (2 \* x \* x + 1)

Print "f ("; x; ") ="; f

End Sub

Private Sub Command2\_Click()

End

End Sub

1. **Моделирование алгоритмов разветвленной структуры**

Составить блок-схему и написать программу на языке VBasic разветвляющейся структуры, с использованием оператора IF.

Задача. С клавиатуры вводятся катеты двух прямоугольных треугольников а1,b1,a2,b2. Определить какой из треугольников имеет наибольшую гипотенузу.

РЕШЕНИЕ. Для сравнения гипотенуз сравним квадраты их длин.

Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

**Блок-схема**

a1,b1,a2,b2

С1=a1\*a1+b1\*b1

«1»

C1=C2

C1>C2

C2=a2\*a2+b2\*b2

«РАВНЫ»

«2»

+

\_

\_

+

**В блок- схеме:**

**С1 – квадрат гипотенузы первого треугольника**

**С2 – квадрат гипотенузы второго треугольника**

**«1» - означает, что гипотенуза 1 треугольника больше;**

**«равны» - гипотенузы треугольников равны;**

**«2» - гипотенуза второго треугольника больше.**

**Программный код задачи:**

Private Sub Command1\_Click()

Dim a1, b1, a2, b2 As Integer

Dim c1, c2 As Double

a1=InputBox("введите катет первого треугольника")

b1 = InputBox("введите второй катет первого треугольника")

a2 = InputBox("введите катет второго треугольника")

b2 = InputBox("введите второй катет второго треугольника")

c1 = a1 \* a1 + b1 \* b1

c2 = a2 \* a2 + b2 \* b2

If c1 > c2 Then Print "Наибольшая гипотенуза у первого треугольника"

Else If c1 = c2 Then Print "у треугольников равные гипотенузы "

Else Print "Наибольшая гипотенуза у второго треугольника"

End Sub

Private Sub Command2\_Click()

End

End Sub

1. **Моделирование алгоритмов с использованием структуры Выбор**

Составить блок-схему и написать программу на языке VBasic разветвляющейся структуры, с использованием оператора SELECT CASE.

Задача. С клавиатуры вводится время суток W [1:24] часа. Вывести сообщение «Вы уже проснулись?» (W<10), «Не пора ли обедать?» (W=12), «Еще не вечер!» (W ≥19), «Как работается?» для остальных значений W.

РЕШЕНИЕ. Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

**Блок-схема**

«ЕЩЕ НЕ ВЕЧЕР!»

W<10

W

\_

+

«ВЫ УЖЕ ПРОСНУЛИСЬ?»

+

\_

W≥19

«КАК РАБОТАЕТСЯ?»

\_

«НЕ ПОРА ЛИ ОБЕДАТЬ?»

W=12

+

**Программный код задачи:**

Private Sub Command1\_Click()

Dim w As Integer

w = InputBox("Введите время суток w=")

Select Case w

Case Is < 10

Print "Вы уже проснулись?"

Case Is = 12

Print "Не пора ли обедать?"

Case Is >= 19

Print "Еще не вечер!"

Case Else

Print "Как работается?"

End Select

End Sub

Private Sub Command2\_Click()

End

End Sub

1. **Моделирование алгоритмов циклической структуры**

Составить блок-схему и написать программу на языке VBasic циклической структуры. Использовать оператор цикла FOR:

Задача. Составить таблицу перевода веса из фунтов в килограмм для значений от 10 до 40 фунтов с шагом 3. 1 фунт = 450 г

РЕШЕНИЕ.

**Блок-схема Программный код задачи:**

Private Sub Form\_Activate()

f:=10, 40, 3

g:= f\*450/1000

f (ф), g(кг)

Dim f, g As Integer

Print " f(фунт) ! g (кг) "

For f = 10 To 40 Step 3

g = 450 \* f

Print " "; f; " ! "; g / 1000

Next

End Sub

Private Sub Command2\_Click()

End

End Sub

1. **Моделирование алгоритмов циклической структуры**

Составить блок-схему и написать программу на языке VBasic циклической структуры. Использовать оператор цикла DO WHILE:

Задача. С клавиатуры по очереди вводятся данные о росте N учащихся класса. Определить средний рост учащихся класса.

РЕШЕНИЕ. Для решения задачи необходимо запросить число учащихся в классе к. Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

**Блок – схема: Программный код задачи:**

Private Sub Command1\_Click()

Dim к, i As Integer

n

к

Dim n, s As Integer

I:=1

к = InputBox("Учащихся в классе k=")

i = 1

I <=k

\_

Do While i < = к

+

n = InputBox("Введите рост ученика n=")

s = s + n

i = i + 1

S:=S+n

Loop

I:=i+1

Print "Средний рост учащихся класса "; s / к

End Sub

Private Sub Command2\_Click()

s/k

End

End Sub

1. **Моделирование алгоритмов циклической структуры**

Составить блок-схему и написать программу на языке VBasic циклической структуры. Использовать оператор цикла DO… LOOP UNTIL.

Задача. В Сбербанк вкладывают сумму денег, равную s1. Через сколько месяцев она возрастет до суммы s2, если процент годовых начислений равен p%?

РЕШЕНИЕ. Нанесу на форму две командные кнопки: для ввода данных и выхода из программы.

**Блок – схема:**

S1, S2, P

P:= p/100/12

S:=S1

i:=1

S:=S\*(1+p)

i:=i+1

S≥S2

-

+

S, i

**Программный код задачи:**

Private Sub Command1\_Click()

Dim s1, s2, i As Integer

Dim p As Single

Dim s As Double

s1 = InputBox("Начальная сумма вклада =")

s2 = InputBox("Конечная сумма вклада =")

p = InputBox("Годовая процентная ставка =")

p = p / 100 / 12

s = s1

I = 1

Do

s = s \* (1 + p)

I = i+1

Loop Until s > = s2

Print "Конечная сумма"; Round(s, 1); " будет через "; i; " месяцев"

Print "или через "; Round(i / 12, 1); " лет"

End Sub

Private Sub Command2\_Click()

End

End Sub