



# Программирование на языке ВЫСОКОГО уровня.

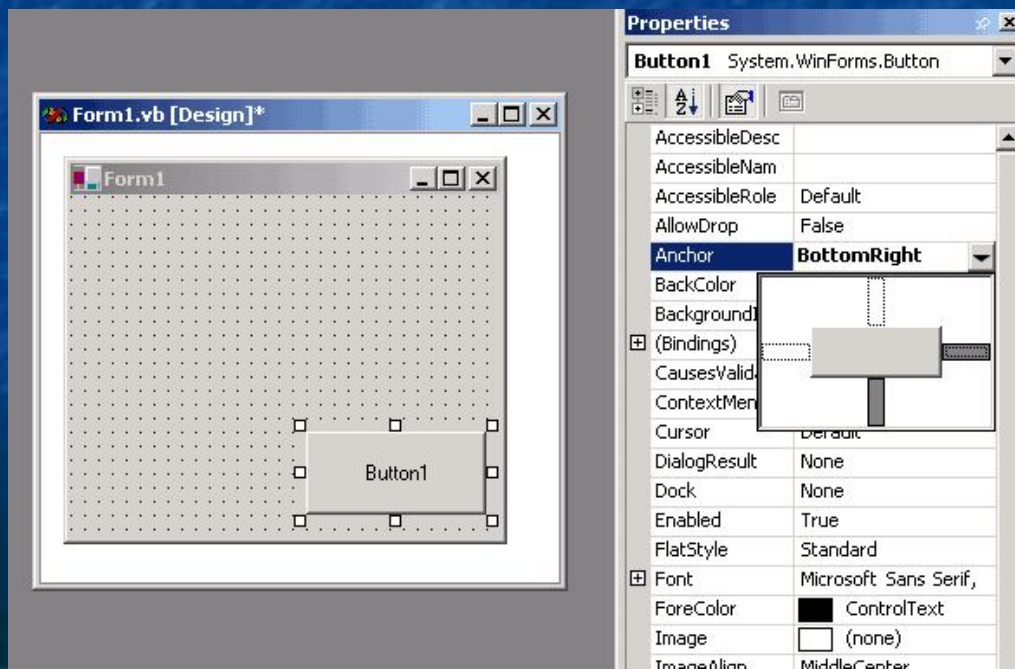
Бужинский В.А. ктн доцент  
[bva2516@mail.ru](mailto:bva2516@mail.ru)

# Вебинар № 3/2.

## Визуализация процесса программирования

### 1. Принципы визуализации

### 2. Инструменты



## Основная литература

Ашарина И.В. Объектно-ориентированное программирование в С++ [Электронный ресурс]: учебное пособие/ Ашарина И.В.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 320 с.— Режим доступа: <http://www.iprbookshop.ru/12008>.

Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual С# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2011.— 180 с.— Режим доступа: <http://www.iprbookshop.ru/19258>.

Агапов В.П. Основы программирования на языке С# [Электронный ресурс]: учебное пособие/ Агапов В.П.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2012.— 128 с.— Режим доступа: <http://www.iprbookshop.ru/16366>.

Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Высокоуровневые языки программирования были разработаны для платформенной независимости сути алгоритмов. Зависимость от платформы перекладывается на инструментальные программы — трансляторы, компилирующие текст, написанный на языке высокого уровня, в элементарные машинные команды (инструкции). Поэтому, для каждой платформы разрабатывается платформенно-уникальный транслятор для каждого высокоуровневого языка, например, переводящий текст, написанный на Delphi в элементарные команды микропроцессоров семейства x86.

Так, высокоуровневые языки стремятся не только облегчить решение сложных программных задач, но и упростить портирование программного обеспечения. Использование разнообразных трансляторов и интерпретаторов обеспечивает связь программ, написанных при помощи языков высокого уровня, с различными операционными системами программируемыми устройствами и оборудованием, и, в идеале, не требует модификации исходного кода (текста, написанного на высокоуровневом языке) для любой платформы.

Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков. Одним из следствий этого стало добавление поддержки того или иного языка низкого уровня (язык ассемблера) в ряд современных профессиональных высокоуровневых языков программирования.

Примеры: C++, C#, Java, JavaScript, Python, PHP, Ruby, Perl, Паскаль, Delphi, Лисп. Языкам высокого уровня свойственно умение работать с комплексными структурами данных. В большинстве из них интегрирована поддержка строковых типов, объектов, операций файлового ввода-вывода и т. п.

Первым языком программирования высокого уровня считается компьютерный язык Plankalkül, разработанный немецким инженером Конрадом Цузе ещё в период 1942—1946 годах. Однако транслятора для него не существовало до 2000 года. Первым в мире транслятором языка высокого уровня является ПП (Программирующая Программа), он же ПП-1, успешно испытанный в 1954 году. Транслятор ПП-2 (1955 год, 4-й в мире транслятор) уже был оптимизирующим и содержал собственный загрузчик и отладчик, библиотеку стандартных процедур, а транслятор ПП для ЭВМ Стрела-4 уже содержал и компоновщик (linker) из модулей. Однако, широкое применение высокоуровневых языков началось с возникновением Фортрана и созданием компилятора для этого языка (1957).

Visual Basic (VB) — является языком программирования третьего поколения (событийный язык программирования) и среда разработки от Microsoft для модели программирования COM.

Этот язык был получен из BASIC и допускает быструю прикладную разработку (RAD) графического интерфейса пользователя (GUI), доступ к базам данных при помощи DAO, RDO, ADO, создание элементов управления ActiveX и объектов. Языки сценариев (VBA, VBScript) синтаксически подобны Visual Basic, но отличаются от него.



Язык Visual Basic унаследовал дух, стиль и отчасти синтаксис своего предка — языка Бэйсик, у которого есть немало диалектов.

В то же время Visual Basic — современный язык программирования, сочетающий процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования.

Среда разработки VB включает инструменты для визуального конструирования пользовательского интерфейса.

Программист может создавать приложение, используя компоненты предоставленные в Visual Basic. Программы, написанные в Visual Basic, могут также использовать Windows API (требует объявление внешних функций).

Visual Basic легко использовать для изучения и использования. Он позволяет разрабатывать не только простые приложения с графическим интерфейсом, но и создавать сложные приложения.

Программирование в VB представляет собой сочетание визуальных компонентов и контролов, определение атрибутов и действий для компонентов, написание дополнительного кода для расширения функциональных возможностей.

Определенные по умолчанию значения и действия для компонентов позволяют создать простую программу без написания кода программистом. Ранние версии имели определенные проблемы с производительностью программ, но с использованием современных компьютеров и компиляцией собственного кода данная проблема стала менее значимой. Компиляция программ была введена в Visual Basic 5, но при этом исполняемые программы до сих пор требуют наличия определенных библиотек для запуска. Эти библиотеки включены в Windows, начиная с Windows 2000, но для более ранних версий Windows они должны были поставляться вместе с программой.

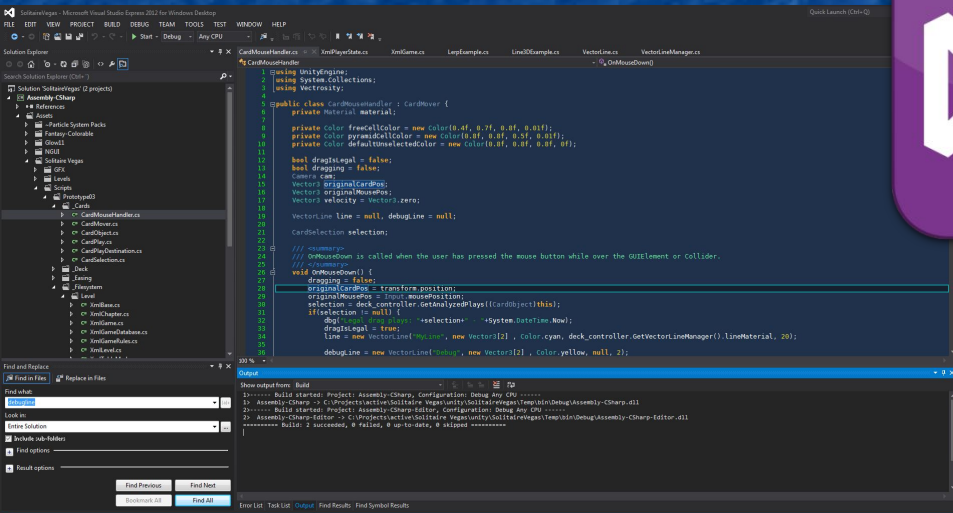


Формы создаются при помощи технологии «перетащи и брось» (drag and drop). Элементы управления просто размещаются на форме (например, поля ввода, кнопки, и т.д.). Элементы управления имеют свои атрибуты и обработчики событий. Многие атрибуты могут быть изменены во время исполнения программы, что позволяет создавать программы, динамически реагирующие на действия пользователя.

Visual Basic может создать исполняемые программы (EXE файлы), элементы управления ActiveX, библиотеки DLL, но, прежде всего, используется для разработки Windows приложений. Диалоговые окна с ограниченными функциональными возможностями могут использоваться для обеспечения подсказок. Элементы управления обеспечивают основные функциональные возможности приложения, а программист может расширить логику программы при помощи обработчиков событий. Например, выпадающий список автоматически отображает список и позволяет пользователю выбрать элемент. Обработчик событий вызывается для исполнения дополнительного кода в зависимости от выбранного элемента.

Язык использует справочную информацию для сбора мусора, имеет большую библиотеку сервисных объектов, объектно-ориентированную разработку. В отличие от многих других языков программирования, Visual Basic не зависит от регистра. Сравнение строк зависит от регистра, но может быть выполнено без учета регистра.

Компилятор Visual Basic поставляется вместе с другими языками Visual Studio (C, C++), но ограничения в интегрированной среде разработки не позволяют создание некоторых видов приложений.



Visual Basic обладает следующими чертами:

булевская постоянная «истина» имеет числовое значение  $-1$  и «ложь» значение  $0$ , потому что тип Boolean хранится как 16-разрядное целое число со знаком. Они взаимосвязаны между собой через логическую операцию Not иначе говоря Истина = Не Ложь.

логические и битовые операторы объединены. Этим он непохож ни на один C подобный язык (Java, Perl).

массивы объявляются с определением верхней и нижней границ, как это делается в Pascal и Fortran. Возможно, при использовании оператора Option Base, установить нижнюю границу по умолчанию, но это приводит к запутанности программ. Нижняя граница массивов не ограничена  $0$  или  $1$  и можно устанавливать более низкие границы. В Visual Basic .NET и VBScript нижняя граница фиксирована.

сильная интеграция с операционной системой Windows и COM. целые числа автоматически преобразуются в вещественные при использовании деления (/). Оператор (\) производит деление с усечением дробной части.

для переменных, которые были объявлены без указания типа, используется по умолчанию тип variant. Но при использовании оператора DefType можно установить иной тип по умолчанию ( DefInt, DefBool, DefVar, DefObj, DefStr и др.).

Тип по умолчанию может быть отменен для переменной при использовании специального символа-суффикса в имени переменной.

Корпорация Microsoft разработала разновидности Visual Basic для создания сценариев. Со временем оригинальный Visual Basic был заменен версией .NET.

Классический Visual Basic (версии 5-6) — язык очень сильно привязан к своей среде разработки и к операционной системе Windows, являясь исключительно инструментом написания Windows-приложений. Привязка к среде заключается в том, что существует большое количество средств, предназначенных для помощи и удобства программирования: встроенный отладчик, просмотр переменных и структур данных на лету, окно отладки, всплывающая подсказка при наборе текста программы. Все эти преимущества делают бесполезным и даже невозможным использование Visual Basic вне среды разработки, например в обычном текстовом редакторе

Visual Basic for Applications (VBA) — средство программирования, практически ничем не отличающееся от классического Visual Basic, которое предназначено для написания макросов и других прикладных программ для конкретных приложений. Наибольшую популярность получил благодаря своему использованию в пакете Microsoft Office. Широкое распространение Visual Basic for Applications в сочетании с изначально недостаточным вниманием к вопросам безопасности привело к широкому распространению макровирусов

Visual Basic Scripting Edition (VBScript) — скриптовый язык, являющийся несколько усечённой версией обычного Visual Basic. Используется в основном для автоматизации администрирования систем Windows, а также для создания страниц ASP и сценариев для Internet Explorer

Visual Basic .NET — преемник Visual Basic 6.0 и часть платформы Microsoft .NET

File Edit View Search Run Debug Options Window Help

[1] CLOCK.FRM

```
' You have a royalty-free right to use, modify, repr
' and distribute the sample applications and toolkit
' Visual Basic for MS-DOS (and/or any modified versi
' in any way you find useful, provided that you agre
' Microsoft has no warranty, obligations or liabilit
' any of the sample applications or toolkits.
'-----
' Include file containing declarations for called pr
'$INCLUDE: 'clock.bi'

CONST FALSE = 0
CONST TRUE = NOT FALSE
CONST ALARMSOUND = "MBT255L1606C04GEDCBT255L1606C04GED

DIM SHARED AlarmTime AS STRING      ' Alarm ti
DIM SHARED TimeFmt AS STRING        ' Time dis
```

← →

F1=Help | Display version number, copyright, and owner information

Index  
Contents

Keyboard

Topic: F1  
Using Help Shift+F1  
Tutorial

About...



```

Option Explicit
Private Sub Form_Load()
    Me.Show
    lblHello.Caption = "Hello World!"
End Sub
Private Sub cmdClose_Click()
    End
End Sub

```

Project - Hello\_World

- Forms
  - frmHello (frmHello)

Properties - frmHello

frmHello Form

Alphabetic	Categorized
(Name)	frmHello
Appearance	1 - 3D
AutoRedraw	False
BackColor	<input type="checkbox"/> &H8000000F&
BorderStyle	1 - Fixed Single
Caption	Hello World!
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	<input checked="" type="checkbox"/> &H00000000&

Hello\_World - frmHello (Form)

Hello World!

**Hello World!**

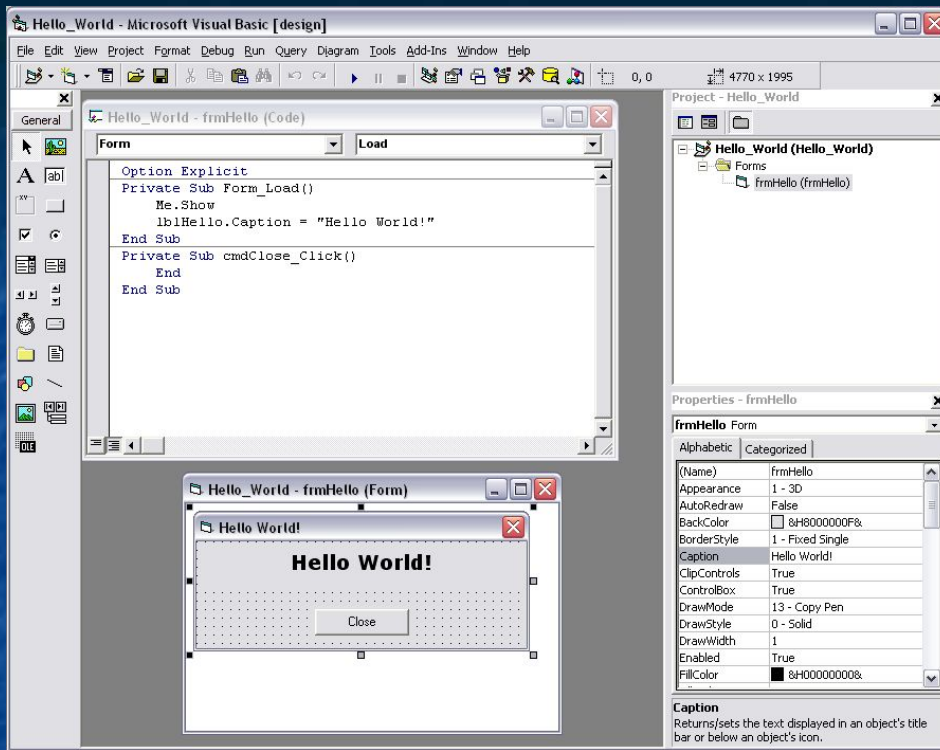
Close

**Caption**  
Returns/sets the text displayed in an object's title bar or below an object's icon.

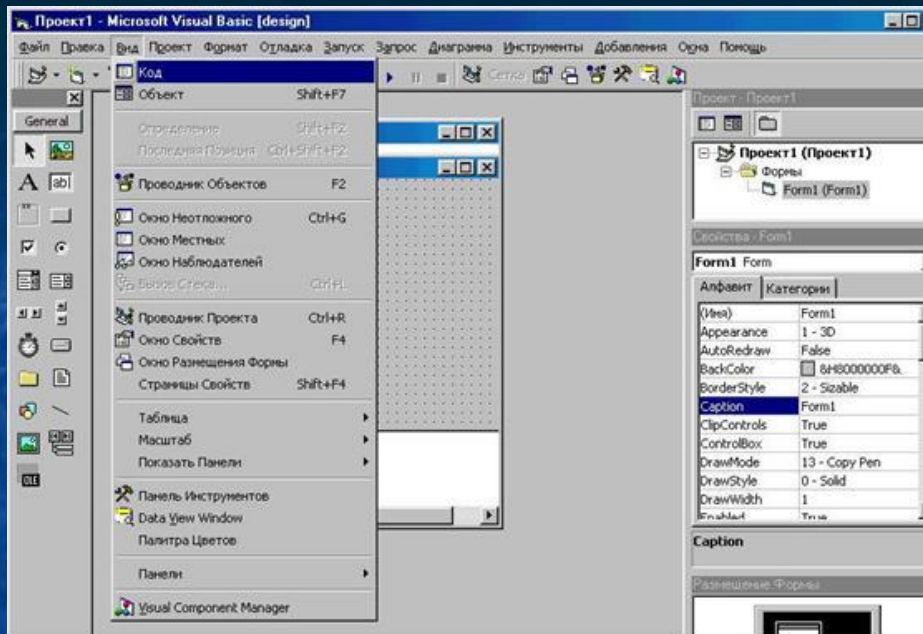


Факториал вычисляется итеративно; значения накапливаются в строковой переменной и затем выводятся в одном окне. & — оператор конкатенации, CStr преобразует число в строку (несмотря на то, что факториал хранится как число с плавающей точкой, выводится он без дробной части), Chr возвращает символ с заданным ASCII-кодом.

```
Sub Factorial()  
    Dim f As Double  
    Dim res As String  
    f = 1  
    For i = 0 To 16  
        res = res & CStr(i) & "! = " & CStr(f) & Chr(10)  
        f = f * (i + 1)  
    Next i  
    MsgBox (res)  
End Sub
```



Теперь мы найдем место, куда будем заносить собственно программу. Для этого в верхнем ряду окна нажимаем кнопку "Вид". В выпавшем меню первая строка и есть та, которая нам нужна: "Код".



Вверху этого окна у нас имеется два выпадающих списка. Слева - список доступных объектов, справа – события для этих объектов. Объектов у нас сейчас два: (General) и Form. Выбираем строку Form. В поле окна кода появились две строки. Что это значит. Первая строка: **Private Sub Form\_Load()** говорит о том, что мы начинаем подпрограмму (Private Sub) или правильней процедуру, которая будет выполняться при загрузке (\_Load) нашей формы (Form). Загрузка - это возникающее событие, на которое и реагирует эта процедура. Последняя строка **End Sub** просто говорит о том, что этой строкой процедура Form\_Load завершается.

Процедура:

```
Private Sub Form_Load()
```

```
Form1.Print "Я стану программистом"
```

```
End Sub
```

Строка, которую мы только что вписали в процедуру, представляет собой оператор, в котором первое слово Form1 представляет собой объект с которым мы работаем (не даром Visual Basic - это объектно-ориентированный язык). В нашем случае объект - это форма.

Последний этап – создание выполняемого (т.е. EXE) файла, то ради чего мы это все делали. Нажми “Файл – Создать Проект1.exe...”, сохраняй и готово.

Теперь можно закрыть VB и запускать свой EXE-файл.

### **Некоторые возможности работы с формой\*\*\*\*\***

**Private Sub Form\_Load()**

**Form1.AutoRedraw = True** 'включает перерисовку формы 1

**Form1.BackColor = vbBlack** 'меняет цвет формы 1 на черный

**Form1.ForeColor = &HFF&** 'меняет цвет текста формы 1 на красный

**Form1.BorderStyle = 3** 'меняет рамку (бордюр) формы 1 на неизменяемую

**Form1.Font.Size = 18** 'меняет размер шрифта формы 1 на 18

**Form1.Font.Underline = True** 'подчеркивает текст в форме 1

**Form1.Caption = "Моя первая программа"** 'устанавливает заголовок формы 1

**Form1.Print "Я стану программистом"** 'выводит текст на форму 1

**Form2.Show** 'делает видимой форму 2

**Form2.Width = 9000** 'устанавливает ширину формы 2 равной 9000

**ТВИПОВ**

**Form2.WindowState = 1** 'сворачивает форму 2

**End Sub**