

Частное учреждение образования  
«Колледж бизнеса и права»

УТВЕРЖДАЮ  
Заместитель директора  
по учебной работе  
\_\_\_\_\_Голубков В.К.  
«\_\_»\_\_\_\_\_2012

Специальность 2-40 01 01: «Программное обеспечение ин- формационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 25.11.2011	

**Лабораторная работа № 25**

**Инструкционно-технологическая карта**

**Тема:** Разработка простого приложения с использованием базовых структур алгоритмов

**Цель работы:**

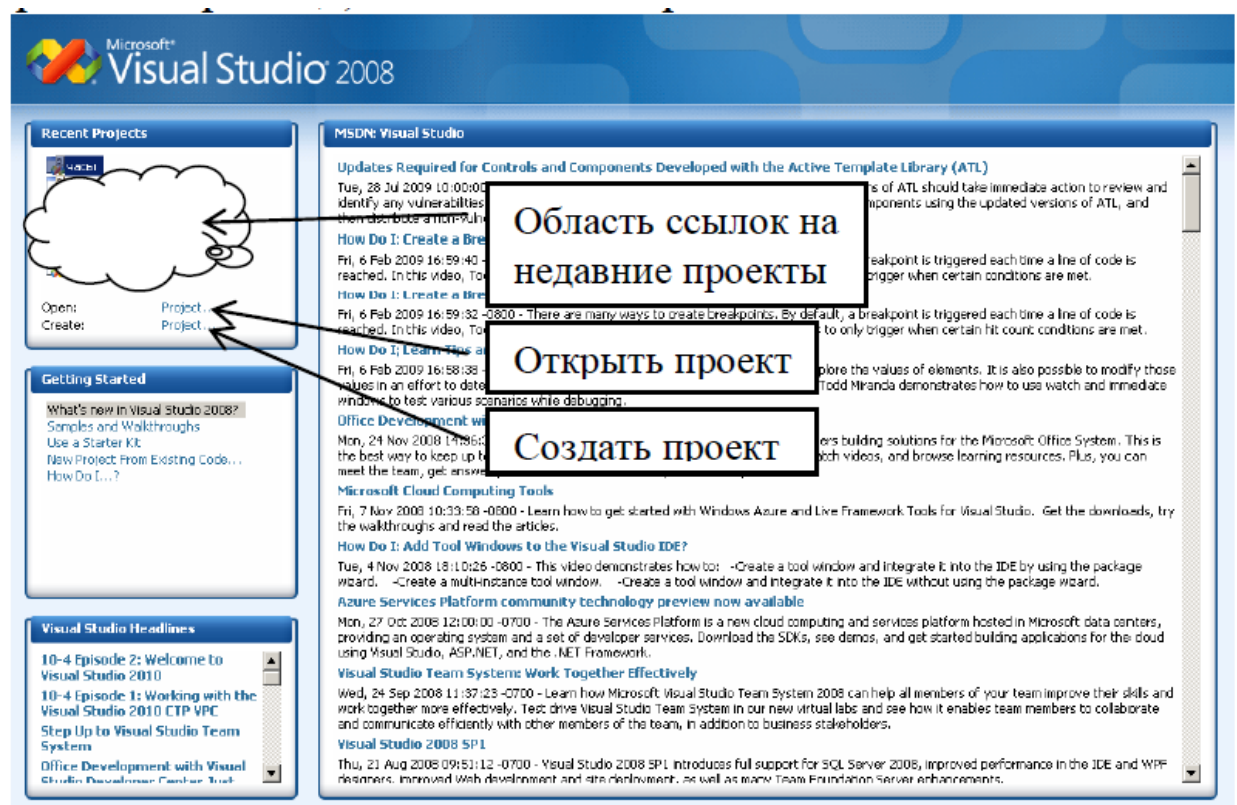
- научиться работать с основными элементами среды проектирования. Познакомиться с основными окнами интегрированной среды разработки проектов;
- научиться разрабатывать программы для реализации базовых структур алгоритмов: следование, разветвление и цикл, с использованием методов стандартного класса Math;
- получить представление об операциях и встроенных математических функциях.

**Время выполнения:** 2 часа

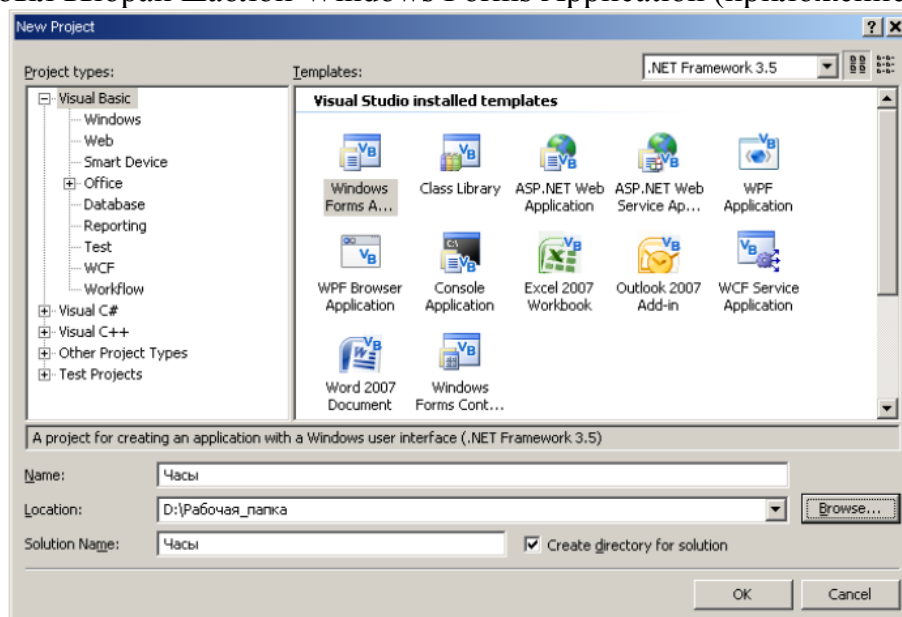
**Краткие теоретические сведения**

Visual Basic представляет собой удобный графический интерфейс в процессе разработки программы (приложения).

Для запуска Visual Studio 2008 щелкните на кнопке Пуск. В последовательно открывающихся окнах выберите Все программы, Microsoft Visual Studio 2008, и еще раз выберите Microsoft Visual Studio 2008. Откроется стартовая страница, показанная на рисунке ниже.

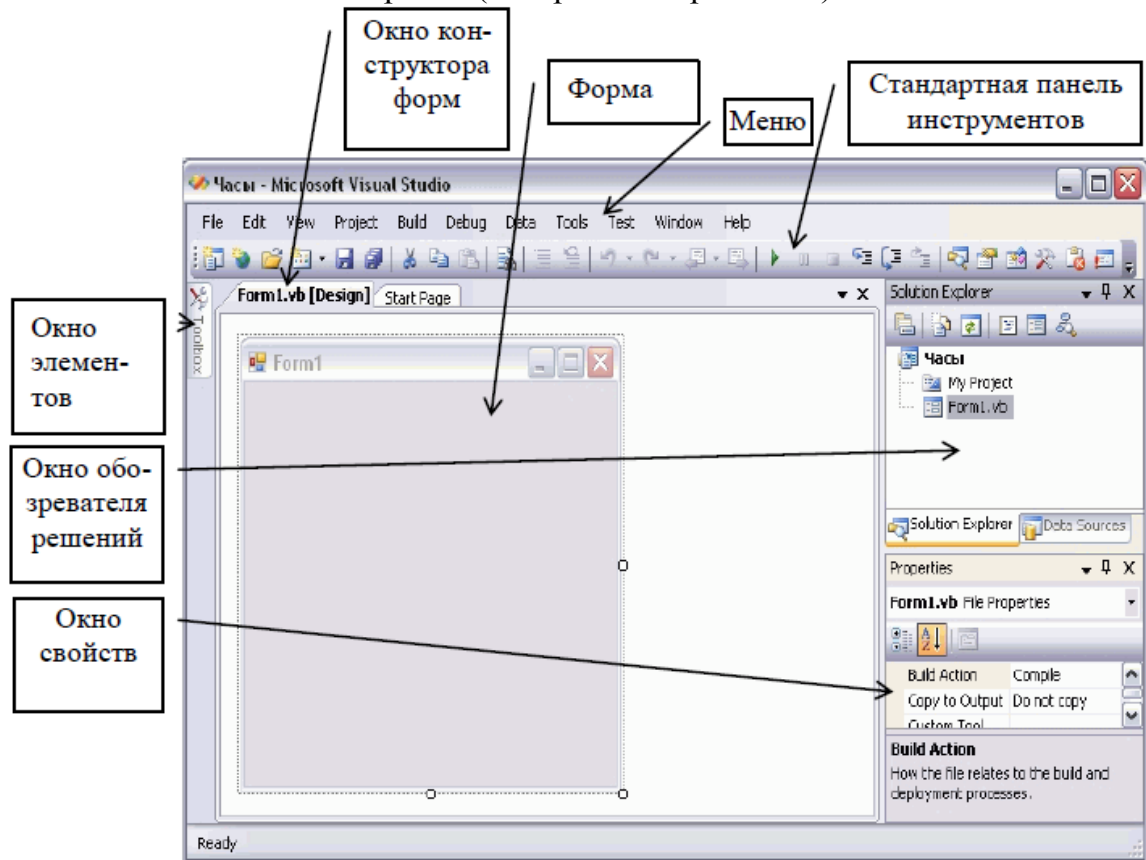


На стартовой странице в окне Recent Project (недавние проекты) находятся ссылки, которые позволяют: открыть один из недавних проектов, открыть существующий проект (Open: Project), или создать новый проект (Create: Project). Для создания нового проекта необходимо щелкнуть в окне Recent Project (недавние проекты) на ссылке Project в предложении Create: Project. В открывшемся окне New Project (новый проект), показанном на рисунке ниже, необходимо проверить, чтобы в окне Project Types (тип проекта) был выбран тип Visual Basic, а также в окне Templates, был выбран шаблон Windows Forms Application (приложение Windows).






Затем в поле Name необходимо заменить предложенное по умолчанию имя проекта необходимым именем, например «Часы» и указать папку, где будут сохранены файлы проекта. Завершение создания и сохранения проекта происходит щелчком по кнопке ОК. После этого на экране монитора отобразиться окно инте-

гированной среды разработки (Integrated Development Environment) Visual Studio 2008 (принятое сокращение IDE). Это окно приведено на рисунке ниже. Если окно Form1.vb (окно конструктора форм) не отобразилось, то щелкните дважды на узле Form1.vb в окне Solution Explorer (обозреватель решений).



Если открыто больше окон, чем требуется, то лишние окна можно закрыть. Если же открыты не все требуемые окна, то недостающие окна следует открыть, пользуясь соответствующими кнопками стандартной панели инструментов:

- ✓ окно обозревателя решений – кнопкой  Solution Explorer;
- ✓ окно свойств – кнопкой  Properties Window;
- ✓ окно элементов – кнопкой  Toolbox.

Окно конструктора форм можно открыть кнопкой View Designer, расположенной на панели инструментов окна обозревателя решений. Если эта кнопка там не отображена, то для ее отображения следует в окне обозревателя решений дважды щелкнуть на компоненте проекта Form1.vb.

Перед рассмотрением вопроса проектирования интерфейса приложения на Visual Basic, необходимо представлять, из чего вообще состоит этот проект?

В Visual Basic любой проект состоит из следующих файлов:

- ✓ файл каждой формы (расширение frm). Это обычный ASCII текстовый файл, в котором записан весь код, помещённый в форму, а также свойства всех помещённых на форму элементов управления и самой формы тоже;
- ✓ файл каждой формы, содержащий бинарную информацию (например картинку в PictureBox) (расширение frx);
- ✓ файл проекта, содержащий информацию о проекте (расширение vbp);
- ✓ информация о рабочей области проекта (workspace) (расширение vbw).

Это необходимый минимум. Хотя, бывают и исключения, например, когда в проекте не используются формы. Тогда вместо `frm` файла, будет `bas` файл.

Далее перечислим дополнительные файлы, которые могут быть подключены к проекту:

- ✓ файл каждого модуля (расширение `bas`) Это текстовый файл;
- ✓ файл каждого модуля классов (расширение `cls`). Это текстовый файл;
- ✓ файл каждого дополнительного элемента управления (расширение `ctl`) Это тоже текстовый файл;
- ✓ файл ресурсов (расширение `res`);
- ✓ другие файлы (`osx`, `tlb`, и т.д...).

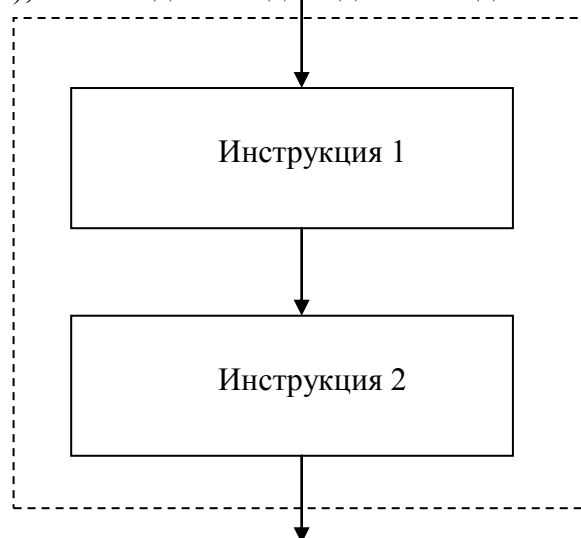
### Базовые структуры алгоритмов

Алгоритм – это правило получения решения некоторой задачи, выраженное в виде совокупности конечного числа элементарных действий. Известны различные способы записи алгоритма. Любая программа на языке VB – это тоже запись соответствующего алгоритма. Однако наибольшей наглядностью обладает изображение алгоритма в виде блок-схемы. В сложных случаях, чтобы правильно разработать алгоритм, иногда полезно перед его записью средствами VB изобразить его блок-схему.

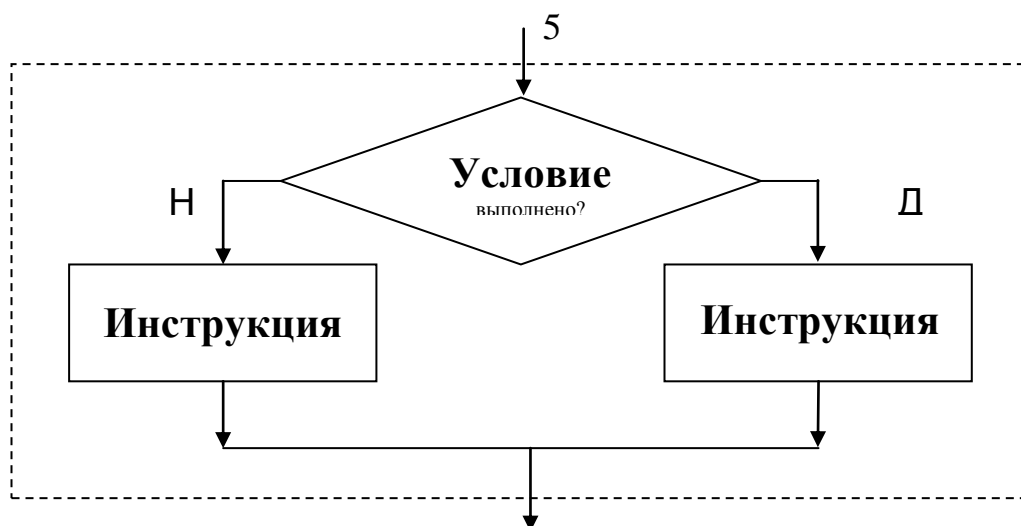
При разработке алгоритма необходимо применять такие технологические рекомендации, при соблюдении которых алгоритм получался бы наиболее понятным, а ошибки при записи алгоритма в виде программы были бы наименее вероятны. В соответствии с современными воззрениями в сегодняшней технологии разработки программ, которые представляют собой одну из сторон метода структурного программирования, алгоритм должен быть структурирован. Он может включать только базовые структуры, которых всего три: следование, разветвление и цикл.

### Следование

Эта структура, изображенная на рисунке ниже, предполагает последовательное выполнение входящих в нее инструкций. Существенно, что структура следование, рассматриваемая как единое целое (на рисунке ниже она заключена в пунктирный прямоугольник), имеет один вход и один выход.



Разветвление, блок-схема которого приведена на рисунке ниже, предполагает проверку некоторого условия. В зависимости от того выполняется это условие или нет, выполняется либо инструкция 1, либо инструкция 2.

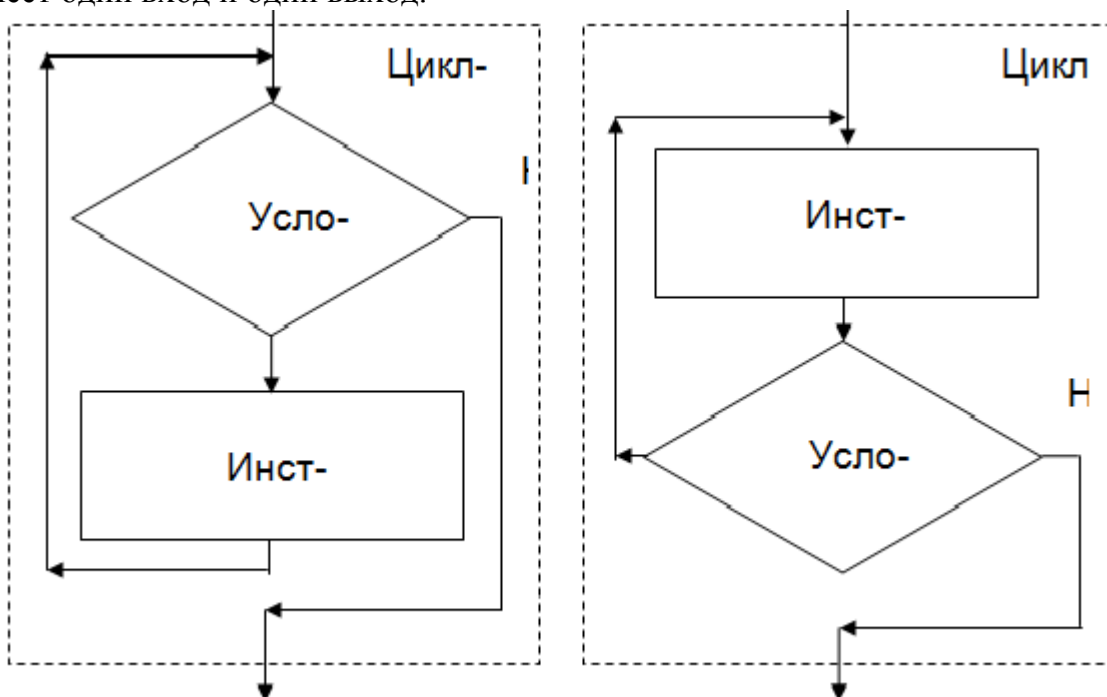


Если на момент проверки условие оказалось выполнено, то будет выполнена инструкция 1, а инструкция 2 игнорируется. Если же оказывается, что условие не выполнено, то будет выполнена инструкция 2, а инструкция 1 игнорируется. Разветвление также имеет один вход и один выход.

### Цикл

Цикл предполагает повторение выполнения некоторой инструкции, а также проверку некоторого условия необходимости повторения этой инструкции. Различают два вида базовых циклов в зависимости от порядка выполнения этих действий: сначала проверка условия необходимости выполнения инструкции, а затем ее выполнение (цикл – пока), или сначала выполнение инструкции, а затем проверка условия необходимости повторения ее выполнения (цикл – до).

На рисунке ниже изображена блок-схема цикла – пока и блок-схема цикла – до. Если условие выполняется (выход по стрелке с надписью да), то инструкция будет выполнена. Если же условие не выполняется (выход по стрелке с надписью нет), то инструкция не будет выполнена и произойдет выход из цикла. Базовый цикл имеет один вход и один выход.



Программирование на VB структуры – следование не встречает затруднений. Для программирования базовой структуры – разветвление применяется условная инструкция **If...End If**.

### Инструкции цикла в VB

Инструкция цикла представляет собой указание, из которого ясно, какой набор инструкций (этот набор инструкций называется телом цикла) нужно выполнять многократно и каково правило окончания выполнения цикла.

#### Параметрический цикл For... Next

Этот цикл управляется параметром, который при повторении выполнения тела цикла изменяет с заданным шагом свое значение от заданного начального значения до заданного конечного значения. Такой цикл используют в случае, когда заранее известно, сколько раз необходимо совершить выполнений тела цикла.

Синтаксис цикла:

**For** ИПЦ = НЗПЦ **To** КЗПЦ [**Step** ШИПЦ]

Тело цикла (одна или несколько инструкций)

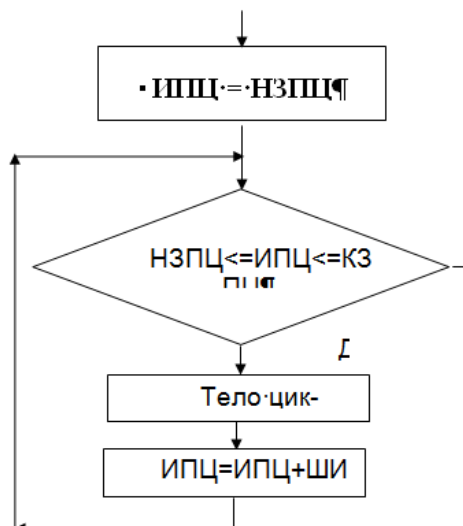
**Next** [ИПЦ]

Слова **For** (для), **To** (до), **Step** (шаг), **Next** (затем) являются зарезервированными. Применены обозначения:

- ✓ **ИПЦ** – имя параметра цикла (переменная любого числового типа);
- ✓ **НЗПЦ** – начальное значение параметра цикла (выражение любого числового типа), которое параметр цикла будет иметь при первом выполнении тела цикла;
- ✓ **КЗПЦ** – конечное значение параметра цикла (выражение любого числового типа), с которым сравнивается текущее значение параметра цикла;
- ✓ **ШИПЦ** – шаг изменения параметра цикла (выражение любого числового типа) – необязательная часть инструкции цикла.

Числовые значения НЗПЦ и КЗПЦ задают интервал, в котором будет изменяться параметр цикла. Необязательный параметр ШИПЦ задает шаг изменения счетчика цикла на каждом проходе. По умолчанию, если он отсутствует, то принимается равным 1. Наконец, после инструкций, составляющих тело цикла, следует команда, обозначающая границу действия цикла. В случае вложенных циклов (в тело цикла входит инструкция цикла) полезно указывать, какой из них закрывается. Это достигается добавлением после слова **Next** имени параметра цикла.

Процесс выполнения инструкции For... Next для положительного шага иллюстрирует рисунок ниже.



Инструкцию **For...Next** можно завершить досрочно с помощью инструкции **Exit For**. Выполнение этой инструкции приводит к немедленному выходу из цикла.

### Инструкция цикла **Do While...Loop** или **Do..Loop While**

Здесь **While** (пока) и **Loop** (цикл) – зарезервированные слова. Циклы типа **While** предназначены для ситуаций, когда количество повторений тела цикла (итераций) заранее не известно. Вот синтаксис двух разновидностей цикла **While**:

**Do While** УсловиеПовторения

Группа инструкций

**Loop**

**Do**

Группа инструкций

**Loop While** УсловиеПовторения

Различие между ними заключается в том, что УсловиеПовторения (условие повторения выполнения тела цикла) проверяется в первом случае до выполнения тела цикла (цикл – пока), а во втором случае – после выполнения тела цикла (цикл – до).

Инструкцию **Do...Loop** можно завершить досрочно с помощью инструкции **Exit Do**.

### Инструкция цикла **Do Until...Loop** или **Do..Loop Until**

**Until** (до) – зарезервированное слово. По своей логике цикл **Until** подобен циклу **While** с той лишь разницей, что выполнение условия означает необходимость выхода из цикла. Как и в случае цикла **While**, проверка условия выхода в цикле **Until** может осуществляться перед очередным проходом или после него. Вот синтаксис этих двух вариантов:

**Do Until** УсловиеВыхода

Группа инструкций

**Loop**

**Do**

Группа инструкций

**Loop Until** УсловиеВыхода

Инструкцию **Do...Loop** можно завершить досрочно с помощью инструкции **Exit Do**.

### Операции и функции

В расположенной ниже таблице перечислены операции, которые могут быть применены к числовым данным.

Арифметические операции: Операция	Функция
[операнд1] + [операнд2]	Сложение
[операнд1] - [операнд2]	Вычитание
- операнд	Перемена знака
[операнд1] * [операнд2]	Умножение
[операнд1] / [операнд2]	Деление
[операнд1] \ [операнд2]	Целочисленное деление. Результат – целая часть отношения операндов без остатка

Арифметические операции: Операция	Функция	
[операнд1] MOD [операнд2]	Деление по модулю. Деление выполняется до получения целой части. Остаток возвращается как результат.	
[операнд1] ^ [операнд2]	Возведение в степень	
Операции отношения: [операнд1] < [операнд2]	Меньше	Результат True, если отношение выполняется, или False, если отношение не выполняется
[операнд1] > [операнд2]	Больше	
[операнд1] <= [операнд2]	Меньше или равно	
[операнд1] => [операнд2]	Больше или равно	
[операнд1] = [операнд2]	Равно	
[операнд1] <> [операнд2]	Не равно	

Если в выражении использовано несколько операций, то в первую очередь выполняются операции, имеющие наивысший приоритет. Если приоритет операций одинаковый, то они выполняются слева направо.

Таблица приоритетов: Приоритет	Операция	Приоритет	Операция
1	Вызов функции, скобки.	5	\
2	^	6	MOD
3	-	7	+, -
4		*, /	

Функция Visual Basic .NET	Описание
Abs(x)	Возвращает абсолютное значение заданного числа x
Atan(x)	Возвращает значение arctg x типа Double.
Cos(x)	Возвращает значение cos x типа Double
Exp(x)	Возвращает значение типа Double, равное числу e (основание натуральных логарифмов) в заданной степени x
Log(x)	Возвращает значение типа Double, равное натуральному логарифму заданного числа x. Может возвращать и логарифм числа x по заданному основанию
Round(x)	Возвращает значение типа Double, равное ближайшему к заданному числу x целому значению
Sign(x)	Возвращает значение (1, 0, -1) типа Integer, указывающее знак числа
Sin(x)	Возвращает значение типа Double, равное sin x



Функция Visual Basic .NET	Описание
Sqrt(x)	Возвращает значение типа Double, равное квадратному корню числа x
Tan(x)	Возвращает значение типа Double, равное tg x
Int(x)	Целая часть
Fix(x)	Целая часть
Rnd	Случайное число

Обязательный аргумент число представляет значение типа Double. Обе функции Int и Fix отбрасывают дробную часть числа и возвращают целое значение. Различие между функциями Int и Fix состоит в том, что для отрицательного значения аргумента число функция Int возвращает ближайшее отрицательное целое число, меньшее либо равное указанному, а Fix ближайшее отрицательное целое число, большее либо равное указанному.

Функция Rnd возвращает значение типа Single, содержащее случайное число, меньшее 1 и большее или равное нулю.

Перед первым вызовом функции Rnd надо использовать инструкцию Randomize без аргумента для инициализации генератора случайных чисел. Для получения случайных целых чисел в заданном диапазоне используйте следующую формулу:

**CInt(Int((ВерхнееЗначение - НижнееЗначение + 1) \* Rnd() + НижнееЗначение))**

Здесь функция CInt преобразует значение к типу Integer.

Для форматирования чисел, дат и времени может быть применена функция Format. Эта функция преобразует число, дату или время в строку символов.

Вот упрощенный синтаксис функции Format:

**Format(Выражение, Формат)**

Ниже приводятся примеры преобразования чисел и дат функцией Format:

Обращение к функции	Результат
Format (315.4,"00000.00")	00315,40
Format (315.4,"#####.##")	315,4
Format (6315.4,"##,##0.00")	6 315,40
Format (315.4,"\$##0.00")	\$315,40
Format ("09.10.2001","dd.mm.yy")	09.10.01
Format ("09.10.2001","dddd, dd mmmm, уууу года")	вторник, 09 октября, 2001 года

### **Применение текстового поля TextBox для вывода значений данных.**

Для вывода значений данных полезно использовать текстовое поле.

Отметим некоторые особенности.

Константа vbTab создает некоторый интервал при выводе между значением переменной a и значением переменной b. Константа vbCrLf обеспечивает перевод строки и возврат каретки (иначе говоря, делает текущей следующую строку текстового окна). Константы vbTab и vbCrLf позволяют управлять расположением выводимых результатов.

Метод AppendText делает текущую строку текстового окна равной его аргументу (строке, помещенной в скобках после имени этого метода).

Чтобы текстовое окно могло содержать не одну, а несколько строк, его свойство Multiline должно иметь значение True.

### Порядок выполнения работы

1. Изучить теоретический материала.
2. Задано аналитическое выражение для некоторой функции  $f(x)$ . Составьте проект для получения таблицы значений функции  $f(x)$  при  $n$  значениях аргумента  $x$ , изменяющегося от начального значения  $a$  до конечного значения  $b$  с постоянным шагом  $dx=(b-a)/(n-1)$ , например, в виде показанном ниже:

Аргумент	Функция
0.00	-1.000000
0.10	-0.895004
0.20	-0.780067
0.30	-0.655336
0.40	-0.521061
0.50	-0.377583
0.60	-0.225336
0.70	-0.064842
0.80	0.103293
0.90	0.278390
1.00	0.459698

Вид  $f(x)$ , значения  $a$  и  $b$  возьмите из таблицы вариантов, представленной ниже. При выполнении проекта задайте  $n = 11$ .

Номер варианта	Заданная функция $f(x)$	Отрезок $[a, b]$	Значение $f(a)$
1	$\frac{3,8 - 3 \sin \sqrt{x}}{0,35} - x$	[2; 3]	0,3905776
2	$\frac{1}{3 + \sin 3,6x} - x$	[0; 0,85]	0,3333333
3	$\cos \sqrt{1 - 0,3x^3} - x$	[0; 1]	0,5403023
4	$\sin \sqrt{1 - 0,4x^2} - x$	[0; 1]	0,841471
5	$0,25x^3 - x - 1,2502$	[2; 3]	-1,2502
6	$0,1x^2 - x \ln x$	[1; 2]	0,1
7	$3x - 4 \ln x - 5$	[2; 4]	-1,772588
8	$e^x - e^{-x} - 2$	[0; 1]	-2
9	$x + \sqrt{x} + \sqrt[3]{x} - 2,5$	[0,4; 1]	-0,7307382
10	$tgx - \frac{tg^3 x}{3} + \frac{tg^5 x}{5} - \frac{1}{3}$	[0; 0,8]	-0,3333333
11	$\cos \frac{2}{x} - 2 \sin \frac{1}{x} + \frac{1}{x}$	[1; 2]	-1,099089
12	$\sin(\ln x) - \cos(\ln x) + 2 \ln x$	[1; 3]	-1
13	$\ln x - x + 1,8$	[2; 3]	0,4931472
14	$0,4 + \arctg \sqrt{x} - x$	[1; 2]	0,1853982

15	$xtgx - \frac{1}{3}$	[0,2; 1]	-0,2927913
16	$\text{tg}(0,55x+0,1) - x^2$	[0; 1]	0,1003347
17	$2 - \sin \frac{1}{x} - x$	[1,2; 2]	0,0598231
18	$1 + \sin x - \ln(1+x) - x$	[0;1, 5]	1
19	$-\cos(x^{0,52+2}) - x$	[0,5; 1]	0,4029458
20	$\sqrt{\ln(1+x)+3} - x$	[2; 3]	0,024503
21	$e^x + \ln x - 10x$	[3; 4]	-8,815851
22	$3x - 14 + e^x - e^{-x}$	[1; 3]	-8,649598
23	$2\ln^2 x + 6\ln x - 5$	[1; 3]	-5
24	$2x \sin x - \cos x$	[0,4; 1]	-0,6095263
25	$\frac{-x^2}{\cos x - e^{\frac{x^2}{2}} + x - 1}$	[1; 2]	-0,0662284
26	$\sqrt{1-x} - \text{tg} x$	[0; 0,9]	1
27	$\sin^2 x + \cos^2 x - 10x$	[0; 1]	1
28	$e^x + \sqrt{1+e^{2x}} - 2$	[-1; 0]	-0,5665994
29	$\sqrt{1-x} - \cos \sqrt{1-x}$	[0; 0,9]	0,4596977
30	$\text{tg} \frac{x}{2} - \text{ctg} \frac{x}{2} + x$	[1; 2]	-0,2841852
31	$x - \cos x$	[0,5; 2,5]	-0,3775826

3. Ответить на контрольные вопросы.

### Контрольные вопросы

1. Из каких файлов состоит любой проект Visual Basic?
2. Определите понятие алгоритма.
3. Какие структуры алгоритмов относят к базовым структурам?
4. Приведите пример следования.
5. Приведите пример разветвления.
6. Приведите пример цикла-до.
7. Приведите пример цикла-пока.
8. Поясните синтаксис и порядок выполнения инструкции **For**.
9. Поясните синтаксис и порядок выполнения инструкции **While**.
10. Поясните синтаксис и порядок выполнения инструкции **Until..**

Преподаватель

С.В. Бурко

Рассмотрено на заседании цикловой  
комиссии программного обеспечения  
информационных технологий №10  
Протокол № \_\_\_\_\_ от « \_\_\_\_\_ » \_\_\_\_\_ 201\_  
Председатель ЦК \_\_\_\_\_ С.В. Бурко