

VISUAL BASIC

- История развития
- Переменные
- Операторы

ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ VB

Beginner's All-purpose Symbolic Instructional Code

Многоцелевой символьный командный код для начинающих.

- ❑ Язык программирования **Basic** был создан в **1964** году двумя профессорами из Dartmouth College – **Джоном Кенеми** и **Томасом Куртцом** для обучения студентов навыкам программирования.
- ❑ Язык получился настолько простым и понятным, что через некоторое время его начали применять и в других учебных заведениях.

ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ VB

Beginner's All-purpose Symbolic Instructional Code

Многоцелевой символьный командный код для начинающих.

- ❑ В 1975 году **Билл Гейтс и Пол Аллен**, основатели **Microsoft**, создали новую версию Basic для первых компьютеров "Альтаир" (MITS Altairs), способную работать в 4КБ оперативной памяти. Со временем именно эта версия и превратилась в один из самых популярных языков программирования в мире..

ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ VB

Beginner's All-purpose Symbolic Instructional Code

Многоцелевой символьный командный код для начинающих.

- При разработке новой версии языка сохранялась **совместимость** с прежними версиями и программа, написанная для практически первого Basic, вполне (с незначительными изменениями) могла бы работать и в последующих версиях этого языка.

ВИЗУАЛЬНАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ VISUAL BASIC

- ❑ В начале 90-х появляется операционная система **Microsoft Windows** с новым **графическим интерфейсом** пользователя. И в **1991** году появилась первая версия нового инструментального средства **Microsoft Visual Basic**.
- ❑ Система программирования, созданная разработчиками Visual Basic, позволяла "отстраниться" от сложнейшей внутренней структуры Windows и создавать программы из "**кубиков**", как в детском конструкторе.
- ❑ Меню, окна, списки, кнопки, поля ввода текста и другие элементы интерфейса Windows добавлялись в программу с помощью простейших операций **drag&drop**. Свою первую программу VB-программисты создавали уже через несколько минут после начала изучения этого языка

MICROSOFT ВЫБИРАЕТ BASIC

- ❑ в **1994** году выходит **Visual Basic for Applications**. Именно в это время, после включения VBA в состав Microsoft Office, Basic начинает превращаться в один из основных стандартов программирования для **Windows**.
- ❑ **Microsoft** оставила за собой **QuickBasic** - среду разработки программ на **Basic**, отказавшись, от дальнейшей работы над языком программирования **Pascal**, оставив его фирме **Borland**.

ГРАММАТИЧЕСКАЯ ПРОСТОТА MICROSOFT BASIC

- ❑ Популярность **Visual Basic** обусловлена двумя причинами: **относительной простотой и продуктивностью.**
- ❑ Программы на VB работают **медленнее** своих аналогов на C/C++, но все же они достаточно быстры для многих деловых целей и **требуют гораздо меньше времени на разработку.**

ГРАММАТИЧЕСКАЯ ПРОСТОТА MICROSOFT BASIC

- ❑ Выражения в VB ограничиваются одной строкой кода, а переменные объявляются и инициализируются в отдельных строках кода.
- ❑ Операторы **присваивания и сравнения используют один и тот же символ.**
- ❑ **Отсутствуют указатели** — требование Билла Гейтса, начиная с первых версий Microsoft Basic. Хотя указатели полезны, так как разрешают прямой доступ к памяти по любому адресу, их использование сопряжено с ошибками в том случае, если они применяются неаккуратно.

ПЕРЕМЕННЫЕ

тип Visual Basic	Структура типов среды CLR	Номинальное выделение хранилища	Диапазон значений	Суффикс
Объект	Object (класс)	4 байта на 32-разрядной платформе 8 байт на 64-разрядной платформе	Любой тип может храниться в переменной типа Object	
Boolean	Boolean	Зависит от реализации платформы	True или False	
Строка (переменная длина)	String (класс)	Зависит от реализации платформы	От 0 до приблизительно 2 миллиардов символов Юникода	\$
Byte	Byte	1 байт	От 0 до 255 (без знака)	
SByte	SByte	1 байт	-128–127 (подписан)	
Char (один символ)	Char	2 байта	От 0 до 65535 (без знака)	
Short (короткое целое число)	Int16	2 байта	От -32 768 до 32 767 (подписан)	
UShort	UInt16	2 байта	От 0 до 65 535 (без знака)	
Integer	Int32	4 байта	-2,147,483,648–2,147,483,647 (подписан)	
UInteger	UInt32	4 байта	От 0 до 4 294 967 295 (без знака)	
Long (длинное целое число)	Int64	8 байт	-9 223 372 036 854 775 808 по 9 223 372 036 854 775 807 (9,2...E+18 †) (подписанный)	
ULong	UInt64	8 байт	От 0 до 18 446 744 073 709 551 615 (1,8...E+19 †) (без знака)	&

ПЕРЕМЕННЫЕ

тип Visual Basic	Структура типов среды CLR	Номинальное выделение хранилища	Диапазон значений	Суффикс
Single (одинарной точности с плавающей запятой)	Single	4 байта	-3.4028235E+38--1.401298E-45 [†] для отрицательных значений; 1.401298E-45--3.4028235E+38 [†] для положительных значений	!
Double (двойной точности с плавающей запятой)	Double	8 байт	-1.79769313486231570E+308--4.94065645841246544E-324 [†] для отрицательных значений; 4.9406564541246544E-324--1.79769313486231570E+308 [†] для положительных значений	#
Десятичное число	Decimal	16 байт	От 0 до +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) [†] без десятичной запятой; От 0 до +/-7.9228162514264337593543950335 с 28 местами справа от десятичного числа; наименьшее ненулевое число — +/-0,000000000000000000000001 (+/-1E-28) [†]	
Дата	DateTime	8 байт	0:00:00 (полночь) 1 января 0001 по 11:59:59 вечера 31 декабря 9999 г.	
Определяемые пользователем (структура)	(наследует от ValueType)	Зависит от реализации платформы	Каждый член структуры имеет диапазон, определенный его типом данных, и не зависит от диапазонов других элементов.	

ПЕРЕМЕННЫЕ

Требования к имени переменной:

- имя должно содержать не более **255 символов**
- имя может содержать **любые буквы, цифры и символ подчеркивания**, при этом **первый символ в имени должен быть буквой или символом подчеркивания**
- в имени должны **отсутствовать пробелы и знаки пунктуации**
- имя не должно быть ключевым словом**

ПЕРЕМЕННЫЕ

Явное и неявное объявление переменных

```
Const limit As Integer = 33
```

```
Dim m As Integer = 45
```

```
Dim a, b As Integer
```

```
Dim a, b As String
```

```
a = "String variable assignment"
```

```
Dim r, s, t As Boolean
```

```
r = True
```

```
Dim n=45 'System.Int32
```

ПРЕОБРАЗОВАНИЕ ДАННЫХ

Преобразования в VB могут быть: **явные и неявные**

Неявные расширяющее преобразование

```
Dim k As Integer
```

```
Dim q As Double
```

```
' Integer к Double Неявные расширяющее преобразование
```

```
k = 432
```


```
q = k
```

ПРЕОБРАЗОВАНИЕ ДАННЫХ

Преобразования в VB могут быть: **явные и неявные**

Неявные сужающее преобразование

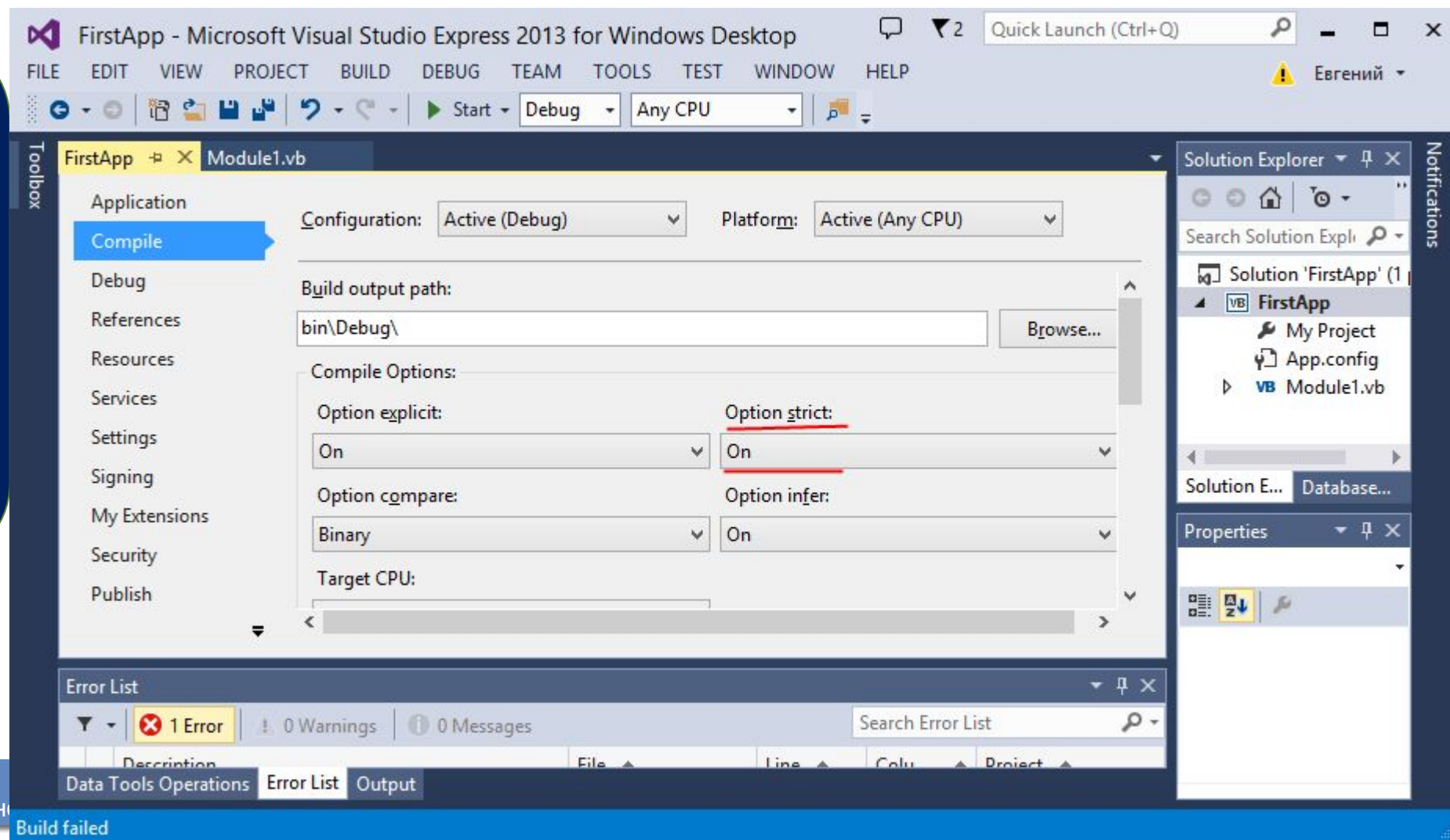
```
Dim num1 As Integer = 100
Dim num2 As Integer = 200
Dim res As Byte = num1 + num2
```



ПРЕОБРАЗОВАНИЕ ДАННЫХ

Преобразования в VB могут быть: **явные и неявные**

Запретить или разрешить неявное сужающее преобразование



Введение в специальность
Лектор Паршина Галина Иван

ОПЕРАТОРЫ ПРЕОБРАЗОВАНИЯ

ЯВНЫЕ

```
Dim num As Integer = 45  
Dim s As String = str(num)
```

```
Dim s As String = "15"  
Dim num As Integer = val(s)
```

ПРЕОБРАЗОВАНИЕ ДАННЫХ

Явные

Ключевое слово преобразования типов	Преобразует выражение в тип данных	Допустимые типы данных выражения для преобразования
<code>CBool</code>	Логический тип данных	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечислимые типы), <code>String</code> <code>Object</code>
<code>CByte</code>	Тип данных <code>Byte</code>	Любой числовой тип (включая <code>SByte</code> и перечислимые типы), <code>Boolean</code> , <code>String</code> <code>Object</code>
<code>CChar</code>	Тип данных <code>Char</code>	<code>String</code> , <code>Object</code>
<code>CDate</code>	Тип данных <code>Date</code>	<code>String</code> , <code>Object</code>
<code>Cdbl</code>	Тип данных <code>Double</code>	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String</code> <code>Object</code>
<code>CDec</code>	Тип данных <code>Decimal</code>	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String</code> <code>Object</code>
<code>CInt</code>	Целочисленный тип данных	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String</code> <code>Object</code>
<code>CLng</code>	Тип данных <code>Long</code>	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String</code> <code>Object</code>
<code>CObj</code>	<code>Object Data Type</code>	Любой тип

ПРЕОБРАЗОВАНИЕ ДАННЫХ

Явные

<code>CByte</code>	Тип данных SByte	Любой числовой тип (включая <code>Byte</code> и перечислимые типы), <code>Boolean</code> , <code>String Object</code>
<code>CShort</code>	Тип данных Short	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String Object</code>
<code>CSng</code>	Тип данных Single	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String Object</code>
<code>CStr</code>	Тип строковых данных	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечислимые типы), <code>Boolean</code> , <code>Char Char</code> массив, <code>Date Object</code>
<code>CType</code>	Тип, указанный после запятой (,)	<p>При преобразовании в <i>базовый тип данных</i> (включая массив элементарной типы), те же типы, что и для соответствующего ключевого слова преобразования.</p> <p>При преобразовании в <i>составной тип данных</i> интерфейсы, которые он реализует, и классы, от которых он наследует.</p> <p>При преобразовании в класс или структуру, на которую вы перегружены <code>CType</code>, этот класс или структура</p>
<code>CUInt</code>	Тип данных UInteger	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String Object</code>

ПРЕОБРАЗОВАНИЕ ДАННЫХ

ЯВНЫЕ

<code>CULng</code>	Тип данных ULong	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String Object</code>
<code>CUShort</code>	Тип данных UShort	Любой числовой тип (включая <code>Byte</code> , <code>SByte</code> и перечисляемые типы), <code>Boolean</code> , <code>String Object</code>

```
Dim n As Long = 15  
Dim num As Integer = CInt(n)
```

```
Imports System.Console
```

```
Dim q$ = "5"  
Dim k% = CType(q, Integer)  
WriteLine(k * 5)
```

ОПЕРАТОРЫ ВВОДА/ВЫВОДА, РАБОТА С КОНСОЛЬЮ

Beep(): подает звуковой сигнал

Clear(): очищает окно консоли

WriteLine(): выводит строку текста с переводом на новую строку

Write(x): выводит строку текста без перевода на новую строку

ReadLine(): считывает введенный пользователем текст

Read() считывает одиночный введенный символ в виде числового кода данного символа

ReadKey(): считывает нажатую клавишу клавиатуры

```
Dim key As ConsoleKeyInfo = Console.ReadKey()
```

```
ReadKey.Key = ConsoleKey.A
```

ОПЕРАТОРЫ ВВОДА/ВЫВОДА

ReadLine()

Read()

WriteLine(string)

Write(string)

```
Imports System.Console
```

```
Module Program
```

```
Sub Main(args As String())
```

```
Write("What is your name? ")
```

```
Dim name As String = ReadLine()
```

```
WriteLine("Hello World and " & name & "!")
```

```
WriteLine($"Hello World and {name}!")
```

```
Dim a As Integer = ReadLine()
```

```
ReadKey()
```

```
End Sub
```

```
End Module
```

Задача 1

Написать программу вывода
таблицы

a = a + 2	a += 2
a = a - 2	a -= 2
a = a * 2	a *= 2
a = a / 2	a /= 2
a = a ^ 2	a ^= 2
a = a \ 2	a \= 2
a = a Mod 2	

Задача 2

Написать программу вывода
таблицы

x	y	Not x	x Or y	x And y	x Xor y
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

ОПЕРАЦИИ ЯЗЫКА VISUAL BASIC.NET

Арифметические операции

- + сложение двух чисел
- - вычитание двух чисел
- * умножение
- ^ возведение в степень
- / обычное деление
- \ целочисленное деление двух чисел
- **Mod** Получение остатка от деления двух чисел

```
Dim x1 As Integer = 6 + 7
```

```
'Результат равен 13
```

```
Dim x2 As Integer = 6 - 7
```

```
'Результат равен -1
```

```
Dim x3 As Integer = 6 * 7
```

```
'Результат равен 42
```

```
Dim x4 As Integer = 12 / 6
```

```
'Результат равен 2
```

```
Dim x5 As Integer = 13 \ 6
```

```
'Результат равен 2, а остаток отбрасывается
```

```
Dim x6 As Integer = 13 Mod 6
```

```
'Результат (он же остаток от деления 13 на 6) равен 1
```

```
Dim x7 As Integer = 6 ^ 2
```

```
'Результат равен 36
```

ОПЕРАЦИИ ЯЗЫКА VISUAL BASIC.NET

Арифметические операции

$a+=1$

$a-=1$

$a*=1$

$a^=2$

$a/=1$

$a\=1$

ОПЕРАЦИИ ЯЗЫКА VISUAL BASIC.NET

Операции сравнения

- ❑ **>** Больше
- ❑ **>=** Больше или равно
- ❑ **<** Меньше
- ❑ **<=** Меньше или равно
- ❑ **=** Равно (в данном случае используется как знак сравнения на равенство двух значений)
- ❑ **<>** Не равно

Логические операции

- ❑ **And** Логическое умножение (логическое И)
- ❑ **Or** Логическое сложение (логическое ИЛИ)
- ❑ **Xor** Исключающее "или"
- ❑ **Not** Логическое отрицание
- ❑ **AndAlso** Сокращенный оператор And
- ❑ **OrElse** Сокращенный оператор Or

ОПЕРАЦИИ ЯЗЫКА VISUAL BASIC.NET

Операции сравнения и Логические операции

Dim x As Boolean = 6 > 2 And 2 < 4

Dim y As Boolean = 6 > 2 And 2 > 4

Dim x1 As Boolean = 6 > 2 Or 2 < 4

Dim y1 As Boolean = 6 > 2 Or 2 > 4

Dim y2 As Boolean = 6 < 2 Or 2 > 4

Dim x2 As Boolean = 6 > 2 Xor 2 < 4

Dim y2 As Boolean = 6 > 2 Xor 2 > 4

Dim x3 As Boolean = Not 2 < 4

Dim y3 As Boolean = Not 2 > 4



ОПЕРАЦИИ ЯЗЫКА VISUAL BASIC.NET

Операции сравнения и Логические операции

`Dim x As Boolean = 6 > 2 And 2 < 4` 'Результат True, так как и первое выражение и второе выражение истинны
`Dim y As Boolean = 6 > 2 And 2 > 4` 'Результат False, так как только одно выражение истинно
`Dim x1 As Boolean = 6 > 2 Or 2 < 4` 'Результат True, так как хотя бы одно выражение истинно
`Dim y1 As Boolean = 6 > 2 Or 2 > 4` 'Результат True, так как опять же одно выражение истинно
`Dim y2 As Boolean = 6 < 2 Or 2 > 4` 'Результат False, так как оба выражения ложны
`Dim x2 As Boolean = 6 > 2 Xor 2 < 4` 'Результат False, так как оба выражения истинны
`Dim y2 As Boolean = 6 > 2 Xor 2 > 4` 'Результат True, так как одно выражение истинно, а другое ложно
`Dim x3 As Boolean = Not 2 < 4` 'Результат False, так как выражение истинно
`Dim y3 As Boolean = Not 2 > 4` 'Результат True, так как выражение ложно

```
Dim x As Boolean = 6 < 2 And 2 < 4
Dim x As Boolean = 6 < 2 AndAlso 2 < 4
```

ОПЕРАТОР ВЫБОРА IF

```
If condition [ Then ]  
    [ statements ]  
[ ElseIf elseifcondition [ Then ]  
    [ elseifstatements ] ]  
[ Else  
    [ elsestatements ] ]  
End If
```

```
Dim R As New Random()  
Dim num1 As Integer = R.Next(10)  
If num1 = 1 Then  
    WriteLine("Число num1 равно 1")  
Elseif num1 = 2 Then  
    WriteLine("Число num1 равно 2")  
Elseif num1 = 3 Then  
    WriteLine("Число num1 равно 3")  
Else  
    WriteLine($"Число {num1} больше 3")  
End If  
ReadKey()
```

ТЕРНАРНЫЙ ОПЕРАТОР ВЫБОРА IF

`Variable = If` (condition, statement `Then`, statement `Else`)

```
Dim num1 As Integer = ReadLine()  
Dim num2 As Integer = ReadLine()  
WriteLine($"Max = {if (num1>num2, num1, num2)}")  
ReadKey()
```

ОПЕРАТОРЫ ВВОДА/ВЫВОДА, РАБОТА С КОНСОЛЮ

BackgroundColor: цвет фона консоли

ForegroundColor: цвет шрифта консоли

BufferHeight: высота буфера консоли

BufferWidth: ширина буфера консоли

Title: заголовок консоли

WindowHeight и **WindowWidth**: высота и ширина консоли соответственно

ОПЕРАТОРЫ ВВОДА/ВЫВОДА, РАБОТА С КОНСОЛЬЮ

' установка красного цвета шрифта

```
Console.ForegroundColor = ConsoleColor.Red
```

```
Console.WriteLine("Введите цвет консоли (синий или зеленый):")
```

```
Dim color As String = Console.ReadLine()
```

```
If (color = "синий") Then
```

```
    Console.ForegroundColor = ConsoleColor.Blue
```

```
Elseif (color = "зеленый") Then
```

```
    Console.ForegroundColor = ConsoleColor.DarkGreen
```

```
End If
```

```
Console.WriteLine("Выбран цвет: {0}", color)
```

```
Console.ReadLine()
```

ОПЕРАТОР МНОЖЕСТВЕННОГО ВЫБОРА SELECT CASE

```
Select [ Case ] testexpression  
  [ Case expressionlist  
    [ statements ] ]  
  [ Case Else  
    [ elstatements ] ]  
End Select
```

```
WriteLine("Введите число от 1 до 10")  
Dim number%=ReadLine()  
Select Case number  
  Case 1 To 5  
    WriteLine("Between 1 and 5, inclusive")  
    ' The following is the only Case clause that evaluates to True.  
  Case 6, 7, 8  
    WriteLine("Between 6 and 8, inclusive")  
  Case 9 To 10  
    WriteLine("Equal to 9 or 10")  
  Case Else  
    WriteLine("Not between 1 and 10, inclusive")  
End Select
```

КОНСТРУКЦИЯ GoTo

```
GoTo Label  
    [ statements ]  
Label:  
    [ elsestatements ] ]
```

```
Dim time As String = "Night" "day"  
If time = "day" Then  
    GoTo M1  
Else  
    GoTo M2  
End If  
M1: Console.WriteLine("Hello World")  
M2: Console.WriteLine("Good Bye World")
```

КОНСТРУКЦИЯ GoTo

```
GoTo Label  
    [ statements ]  
Label:  
    [ elstatements ] ]
```

```
Dim time As String = "day"  
If time = "day" Then  
    GoTo M1  
Else  
    GoTo M2  
End If  
M1: Console.WriteLine("Hello World")  
    GoTo M3  
M2: Console.WriteLine("Good Bye World")  
M3:
```

VISUAL BASIC

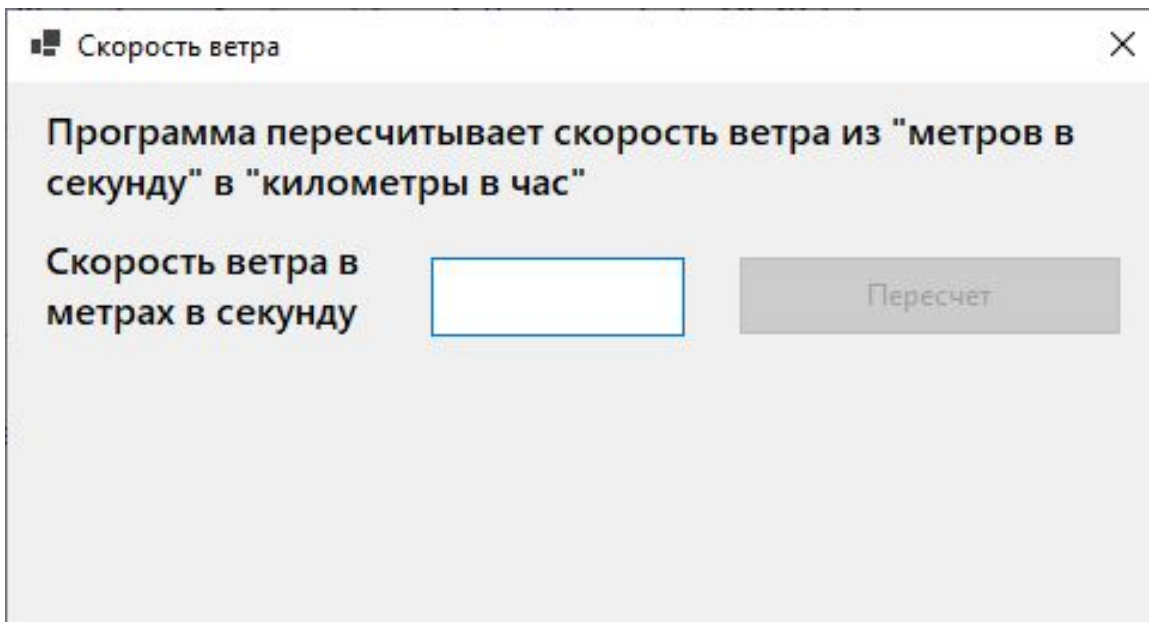
БАЗОВЫЕ КОМПОНЕНТЫ

	Label
	TextBox
	Button
	CheckBox
	RadioButton
	ListBox
	ComboBox

- поле вывода текста;
- поле ввода/редактирования текста;
- командная кнопка;
- независимая кнопка выбора;
- зависимая кнопка выбора;
- список выбора;
- комбинированный список выбора.

ПРИМЕР 1. ПРОГРАММА ПЕРЕСЧЕТА СКОРОСТИ ВЕТРА

Вариант 1. Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час"

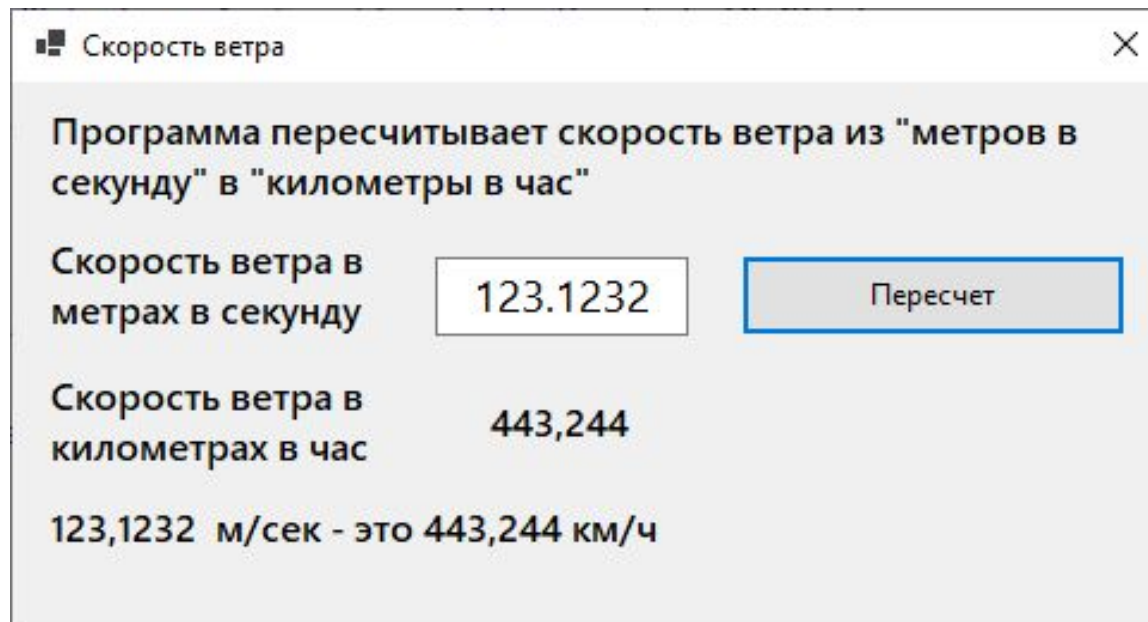


Скорость ветра

Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час"

Скорость ветра в метрах в секунду

Пересчет



Скорость ветра

Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час"

Скорость ветра в метрах в секунду

Пересчет

Скорость ветра в километрах в час **443,244**

123,1232 м/сек - это 443,244 км/ч

ПРИМЕР 1. ПРОГРАММА ПЕРЕСЧЕТА СКОРОСТИ ВЕТРА

Вариант 1. Программа пересчитывает скорость ветра из "метров в секунду" в "километры в час"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim ms As Integer ' скорость м/сек
    Dim kmh As Single ' скорость км/ч
    ms = Val(TextBox1.Text) ' ВВОД ИСХОДНЫХ ДАННЫХ
    kmh = Math.Round(ms * 3.6,3) ' пересчет
    ' ВЫВОД РЕЗУЛЬТАТА
    Label3.Text = $"{ms:f3} м/сек - это {kmh:f3} км/ч"
End Sub
```


ПРИМЕР 1. ПРОГРАММА ПЕРЕСЧЕТА СКОРОСТИ ВЕТРА

Вариант 2. Программа выводит результат только в том случае, если в поле ввода есть данные.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim ms As Integer ' скорость м/сек
    Dim kmh As Single ' скорость км/ч
    ms = Val(TextBox1.Text) ' ввод исходных данных
    If ms = 0 Then Exit Sub
    kmh = Math.Round(ms * 3.6,3) ' пересчет
    ' вывод результата
    Label3.Text = $"{ms:f3} м/сек - это {kmh:f3} км/ч"
End Sub
```

ПРИМЕР 1. ПРОГРАММА ПЕРЕСЧЕТА СКОРОСТИ ВЕТРА

Вариант 3. Кнопка Пересчет доступна только в том случае, если в поле ввода есть данные.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Button1.Enabled = False
End Sub
```

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
    TextBox1.TextChanged
    ' проверим, есть ли в поле исходные данные
    If Len(TextBox1.Text) = 0 Then
        Button1.Enabled = False ' кнопка Пересчет недоступна
    Else
        Button1.Enabled = True ' кнопка Пересчет доступна
    End If
End Sub
```

ПРИМЕР 1. ПРОГРАММА ПЕРЕСЧЕТА СКОРОСТИ ВЕТРА

Вариант 4. Пользователь может ввести в поле Скорость только целое положительное число.

```
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles  
    TextBox1.KeyPress  
        e.KeyChar = If("0123456789".Contains(e.KeyChar) OrElse e.KeyChar = vbBack,  
e.KeyChar, Nothing)  
    End Sub
```

```
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles  
    TextBox1.KeyPress  
        If "1234567890".Contains(e.KeyChar) OrElse e.KeyChar = vbBack  
            OrElse (e.KeyChar = "." And Not TextBox1.Text.Contains(".")) Then  
            e.KeyChar = e.KeyChar  
        Else e.KeyChar = Nothing  
        End If  
    End Sub
```

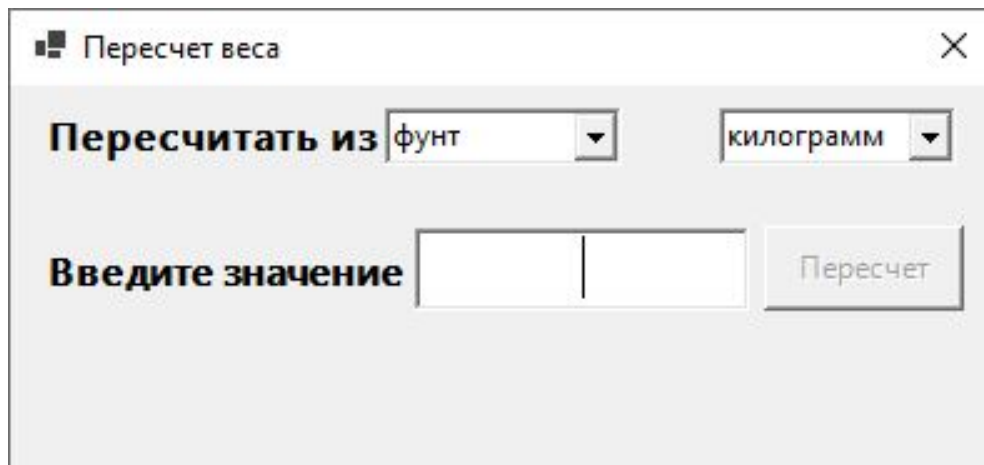
САМОСТОЯТЕЛЬНАЯ РАБОТА

Задание 1

Написать программу, пересчитывающую вес из фунтов в килограммы (1 фунт – 0,4536 килограмм).

Программа составить таким образом, чтобы кнопка Пересчет была доступна только в том случае, если пользователь ввел исходные данные.

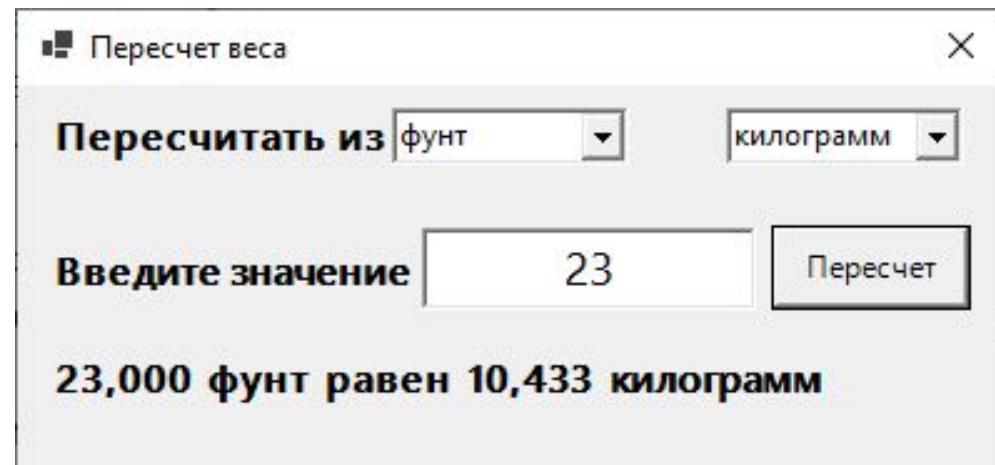
Тип данных Single



Пересчет веса

Пересчитать из

Введите значение



Пересчет веса

Пересчитать из

Введите значение

23,000 фунт равен 10,433 килограмм

ДОМАШНЯЯ РАБОТА

Задание 1

Написать программу, вычисляющую скорость, с которой бегун пробежал дистанцию. Рекомендуемый вид экрана во время выполнения программы приведен ниже.

Вычисление скорости бега

Введите длину дистанции (метры)

Введите время (минут.секунд)

Расчет

Вычисление скорости бега

Введите длину дистанции (метры)

Введите время (минут.секунд)

Расчет

Время: 12 мин 24 сек = 744 сек

Вы бежали со скоростью 0,13 м/сек
или 0,48 км/ч

`Math.Sqrt(d1)`

`Math.Round(0.12335, 4) '0,1234'`

ДОМАШНЯЯ РАБОТА

Задание 2

Ввести a, b, c и рассчитать корни квадратного уравнения

Расчет корней квадратного уравн...

$$a \cdot x^2 + b \cdot x + c = 0$$

a

b

c

Расчитать