

Тема.Основные понятия VB.NET.

<https://msdn.microsoft.com>

Структура программы Visual Basic

Программа Visual Basic построена из стандартных блоков.

Решение состоит из одного или нескольких проектов. *Проект* в свою очередь может содержать одну или несколько сборок. Каждая *сборка* компилируется из одного или нескольких исходных файлов.

Исходный файл включает в себя определения и реализацию классов, структур, модулей и интерфейсов, и в конечном счете содержит весь код.

Элементы программирования уровня файла

При создании проекта или файла и открытии редактора кода, пользователю предоставляется уже готовая часть кода, расположенная в правильном порядке. Любой код должен создаваться в следующей последовательности:

Операторы **Option**

Операторы **Imports**

Операторы **Namespace** и элементы уровня пространства имен

Размещение операторов в иной последовательности может вызвать ошибки компиляции.

Элементы программирования уровня файла

Операторы **Option** устанавливают основные правила для последующего кода, способствуя предотвращению синтаксических и логических ошибок.

Оператор **Imports** позволяет ссылаться на классы и другие типы, определенные в импортированном пространстве имен, без их уточнения.

Библиотека классов .NET Framework имеет иерархическую структуру с единым корнем и состоит из пространств имен (namespaces). Каждое пространство имен содержит данные, которые можно использовать в программе. Пространство имен используется для определения области видимости (scope). Два класса с одним и тем же именем могут использоваться в программе при условии, что они принадлежат различным пространствам имен. Имя пространства имен является частью полного имени объекта, имеющего синтаксис namespace.typename.

Все пространства имен, поставляемые Microsoft начинаются либо с System либо Microsoft.

Например.

```
System.Console.WriteLine("Hello, World")
```

В данном примере System является пространством имен, Console – классом, WriteLine – методом класса Console.

Элементы программирования уровня пространства имен

Классы, структуры и модули содержат весь код исходного файла. Они являются элементами *уровня пространства имен* и могут отображаться в пространстве имен или на уровне исходного файла. Они содержат описания всех других элементов программирования.

Оператор Class (Visual Basic)

Оператор Structure

Оператор Module

Оператор Interface (Visual Basic)

Элементы данных на уровне пространства имен являются перечислениями и делегатами.

Элементы программирования уровня модуля

Процедуры, операторы, свойства и события являются единственными элементами программирования, которые могут содержать исполняемый код (то есть выполняются во время выполнения). Они являются элементами *уровня модуля* программы.

Оператор Function (Visual Basic)

Оператор Sub (Visual Basic)

Оператор Declare

Оператор Operator

Оператор Property

Оператор Event

Элементами данных на уровне модуля являются переменные, константы, перечисления и делегаты.

Процедура Main

Процедура Main выполняется первой при загрузке приложения. Процедура Main служит начальной точкой программы и осуществляет общее управление работой приложения:

```
Sub Main()
```

Операторы в Visual Basic

Оператор в Visual Basic — это полная инструкция. В инструкциях содержатся ключевые слова, операторы, переменные, константы и выражения. Каждая инструкция принадлежит к одной из следующих двух категорий:

Операторы объявления, которые объявляют переменную, константу или процедуру, и могут также указывать тип данных.

Выполняемые операторы, которые инициируют действия. Эти операторы могут вызывать метод или функцию, а также способны выполнять цикл или ветвь, состоящие из блоков кода. Выполняемые операторы содержат **Операторы присваивания**, которые присваивают значение или выражение переменной или константе.

Операторы в Visual Basic

Оператор в Visual Basic — это полная инструкция. В инструкциях содержатся ключевые слова, операторы, переменные, константы и выражения. Каждая инструкция принадлежит к одной из следующих двух категорий:

Операторы объявления, которые объявляют переменную, константу или процедуру, и могут также указывать тип данных.

Выполняемые операторы, которые инициируют действия. Эти операторы могут вызывать метод или функцию, а также способны выполнять цикл или ветвь, состоящие из блоков кода. Выполняемые операторы содержат **Операторы присваивания**, которые присваивают значение или выражение переменной или константе.

Тип данных программного элемента определяет данные, которые могут содержаться в нем, и способы их хранения. Типы данных применяются ко всем значениям, которые могут храниться в памяти компьютера или участвовать в вычислении выражения. Каждая переменная, литерал, константа, перечисление, свойство, параметр процедуры, аргумент процедуры и возвращаемое значение процедуры имеют тип данных.

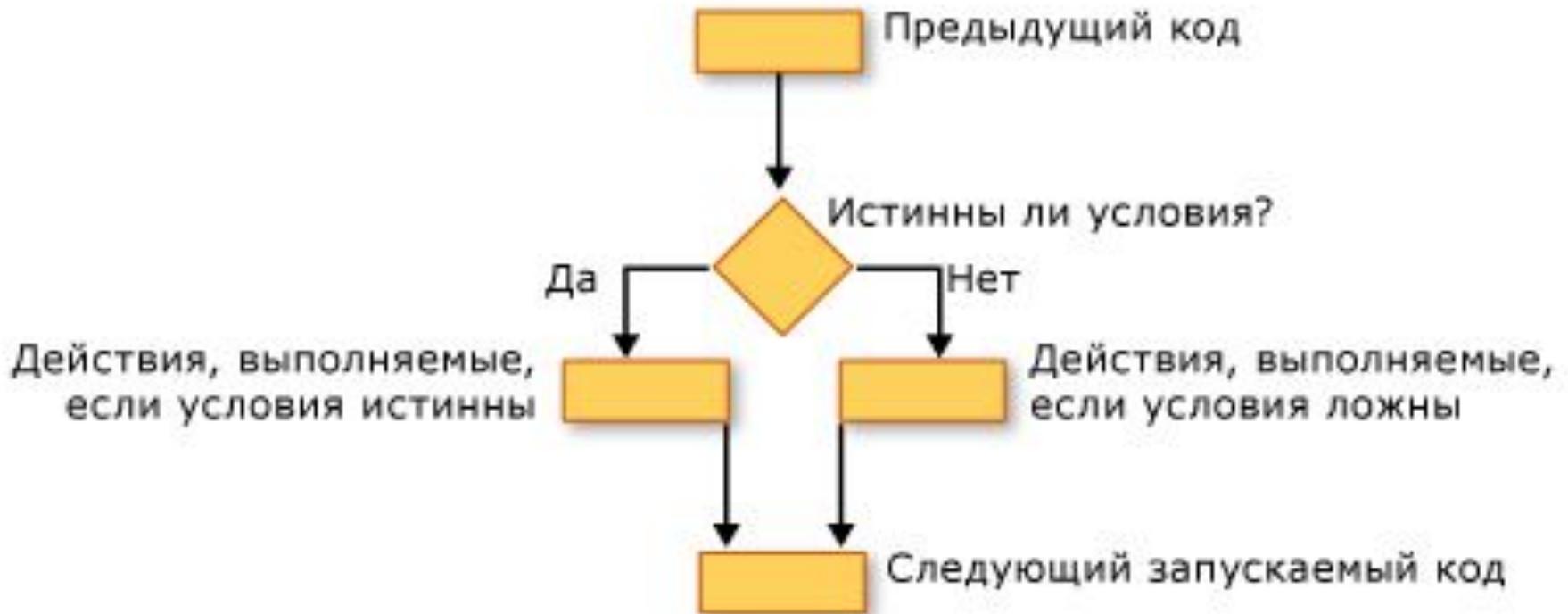
Сводка типов данных (Visual Basic)
(см. файл word)

Структуры решений

Visual Basic позволяет проверять условия и выполнять различные операции в зависимости от результатов такой проверки. Можно проверять истинность или ложность условия для различных значений выражений или исключений, выдаваемых при выполнении последовательности операторов.

Следующий пример показывает структуру решений, которая проверяет условие и выполняет различные действия в зависимости от того, выполняется оно или нет.

Структуры решений



Выполняет различные действия, когда условие имеет значение true или false

Конструкции **If...Then...Else** позволяют проверить одно или несколько условий и выполнить один или несколько операторов для каждого условия. Можно проверить условия и выполнить действия следующими способами:

Если условие **True**, выполнить один или несколько операторов

Если условие **False**, выполнить один или несколько операторов

Если условие **True**, выполнить некоторые операторы, а другие — если условие **False**

Проверить дополнительное условие, если предыдущее условие **False**

Управляющая структура, обеспечивающая все эти возможности — Оператор **If...Then...Else**. Можно использовать простую версию при наличии только одного условия и одного оператора. Если имеется более сложный набор условий и действий, можно использовать составную версию.

Конструкция **Select...Case**

Конструкция **Select...Case** позволяет вычислить выражение один раз и выполнить различные наборы операторов на основе различных значений выражения.

Конструкция **Try...Catch...Finally**

Конструкции **Try...Catch...Finally** позволяют выполнять набор операторов в среде, которая сохраняет управление, если какой-либо из операторов вызовет исключение. Можно выполнять различные действия для различных исключений. Можно указать блок кода, который будет выполняться перед выходом из всей конструкции **Try...Catch...Finally** независимо от того, что произошло.