

№ 23 **Курс:** Процедурное программирование на языке C#
урока:

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

В этом уроке мы с вами рассмотрим условные конструкции, которые могут оказаться полезными, когда вам потребуется в своей программе учитывать условие, в зависимости от которого могут выполняться или не выполняться определенные блоки.

Изучив материал данного занятия, учащийся сможет:

- Понимать смысл операции ветвления.
- Уметь составлять нелинейный алгоритм, выполняющийся при определенном условии.
- Понимать и уметь применять операторы ветвления.

Содержание урока

1. Рассмотрение практической задачи
2. Условная конструкция if
3. Условная конструкция if на практике
4. Условная конструкция if-else

Резюме

- **Алгоритм** – это последовательность действий, команд, необходимых для решения определенной задачи.
- Алгоритмы, в которых действия выполняются последовательно друг за другом, называются - **линейные алгоритмы**.
- Алгоритмы, в которых действия выполняются или нет, в зависимости от условия, называются - **алгоритмами ветвления**.
- **Ветвление** — это алгоритмическая конструкция, в которой та или иная последовательность действий выполняется в зависимости от выполнения условия.
- Структура алгоритма, который содержит ветвление, называется разветвляющейся.
- Операторы ветвления или, как их еще называют, условные операторы, это такие конструкции языка, которые позволяют нам строить нелинейные алгоритмы.
- Условная конструкция **if**, состоит из двух частей. Первая часть – это тело, которое представляет собой, обычный, знакомый нам, блок кода. Вторая часть условной конструкции **if** – это само условие, которое приписывается к телу, то есть к блоку.

- Если условие удовлетворяет истинности, то мы входим в тело условной конструкции и выполняем его. Иначе, если условие не удовлетворяет истинности, то мы в тело условной конструкции не заходим, а продолжаем выполнение программы со следующей строки, которая находится за телом условной конструкции **if**.
- Такая конструкция алгоритма, при которой блок ветвления содержит только оператор **if**, называется **ветвлением в неполной форме**.
- Конструкция **if-else** — это расширенная конструкция **if**. И расширенная она, за счёт добавления блока **else** (иначе).
- При использовании конструкции **if-else**, уже не получится сделать так, чтобы не выполнялся какой-либо один из двух блоков, как это было с использованием чистой конструкции **if**.
- Следует отметить, что блок **else**, не может быть оторван от конструкции **if**, и использоваться сам по себе. Блок **else**, это не самостоятельный блок, блок **else** — это всего лишь дополнение, или расширение конструкции **if**.
- Такая конструкция алгоритма, при которой блок ветвления содержит и оператор **if**, и оператор **else**, называется - **ветвлением в полной форме**.
- **Явное сравнение** - это действие в блоке **if**, когда логическая переменная сравнивается с логическим значением явно - **if (logicVariable == true)**.
- **Неявное сравнение** - это действие в блоке **if**, когда значение логической переменной неявно сравнивается со значением **true** - **if (logicVariable)**.
- Стив Макконнелл рекомендует, по возможности, использовать «неявное сравнение».

Закрепление материала

- Что такое ветвление?
- Чем отличается линейный алгоритм и алгоритм ветвления?
- Какая есть рекомендация по использованию сравнения в блоке **if** и почему?
- Какая есть рекомендация по расположению константы и переменной при операции сравнения между ними и в чем смысл рекомендации?
- В чем отличие ветвления в полной и неполной формах?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Напишите программу, в которой в теле метода Main примите от пользователя число. С помощью конструкции if-else реализуйте алгоритм ветвления, в котором: Если число больше 3 – нужно увеличить значение переменной, хранящей число, на 10. Иначе – увеличить эту переменную в 10 раз. Вывести значение переменной.

- Задание 3

Напишите программу, в которой в теле метода Main примите от пользователя три стороны треугольника. Далее:

1. Проведите проверку – возможен ли треугольник с такими сторонами.
2. Если возможен – проверьте, является ли треугольник прямоугольным, равносторонним и равнобедренным.
3. Выведите информацию о треугольнике на экран.

Рекомендуемые ресурсы

[https://ru.wikipedia.org/wiki/Ветвление_\(программирование\)](https://ru.wikipedia.org/wiki/Ветвление_(программирование))

<https://forum.itvdn.com/t/urok-4-uslovnye-konstrukcii-v-c/3038>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/if-else>