Умовні кострукції

|  |  |  |
| --- | --- | --- |
| **№ Уроку:** | 3 | **Курс:** |

C#

|  |
| --- |
| **Засоби навчання :** |

Компьютер з програмою Visual Studio

**Огляд, мета і призначення уроку**

Розглянемо оператои розгалуження для побудови умовних конструкцій.

**Вивчивши матеріал цього занняття, студент зможе:**

* Розуміти роботу операторів розгалуження.
* Використовувати умовні конструкції: if-else Використовувати тернарний оператор
* Використовувати оператор багатозначного вибору switch-case.

**Зміст уроку**

1. Розглянемо поняття розгалуження в програмуванні.
2. Розглянемо оператори розгалуження.
3. Розглянемо випадки застосування умовних конструкцій.
4. Розглянемо приклад: Умовна конструкція - if (з однією гілкою).
5. Розглянемо приклади: Умовна конструкція - if-else (з двома гілками).
6. Розглянемо приклад: Умовна конструкція - if-else (з декількома гілками). Каскад умовних операторів.
7. Розглянемо приклад: тернарної умовної операція ( ? : ).
8. Розглянемо приклад: Обмеження, пов'язані з безпекою типів.
9. Розглянемо приклад: Вкладенмх тернарних операторів.
10. Розглянемо приклад: Оператор багатозначного вибору - switch-case (перемикач).
11. Розглянемо приклад: Провалення в перемикачах.

**Резюме**

* Оператор розгалуження (умовна конструкція, умовний оператор) - конструкція мови програмування, що забезпечує виконання певної команди (набору команд) тільки за умови істинності деякого логічного виразу, або виконання однієї з кількох команд (наборів команд) в залежності від значення деякого виразу.
* У C # існує три основні форми умовних конструкцій:

1. умовний оператор (if-else)
2. тернарний оператор (?:)
3. оператор багатозначного вибору (перемикач, switch-case).

* Умовний оператор if реалізує виконання певних команд за умови, що логічний вираз в умові, приймає значення true.
* Якщо використовується конструкція if-else, і результатом умови було значення true, то виконається тільки тіло оператора if, а тіло блоку else залишиться не виконаним.
* Після виконання оператора if управління передається наступному оператору.
* Оператор, що виконується після перевірки умови, може бути будь-якого типу, включаючи інший оператор if, вкладений в оригінальний оператор if.
* Якщо тіло блоку if або else складається з одного виразу, то операторні дужки можна опустити
* Тернарна умовна операція (записується як ?:) - операція, яка повертає свій другий або третій операнд в залежності від значення логічного виразу, заданого першим операндом.
* Тернарний оператор [? :], Є скороченою формою конструкції if ... else.
* Тернарний оператор складається з наступних операндів:

(Умова) ? (Блок true) : (блок false);

* Алгоритм роботи тернарного оператора:
* (Логічний вираз) ? вираз 1 : вираз 2

1. Обчислюється логічнИЙ вираз (умова).

2. Якщо логічнИЙ вираз істинний, то обчислюється значення виразу 1 (блоку true), в іншому випадку - значення виразу 2 (блоку false).

3. Обчислене значення повертається.

* Тернарний оператор обов'язково повинен повертати значення, інакше нічого не вийде.
* Або блок true і блок false повинні бути однакового типу, або має існувати неявне перетворення з одного типу в інший.
* Конструкція перемикача switch-case має кілька (дві або більше) гілок. Перемикач виконує одну задану гілку в залежності від значення обчислюваного ключового висловлювання. Принциповою відмінністю цієї конструкції від умовного оператора є те, що вираз, що визначає вибір виконуваної гілки, допускає використання не логічних значень.
* Для порожніх операторів case дозволено "провалювання" від одного оператора до іншого.
* У кожному операторі case вказується постійне значення. Виконується тіло того оператора case, постійне значення якого, відповідає значенню виразу селектора оператора switch.
* Якщо постійний вираз оператора case не містить відповідного значення, виконується блок default, якщо такий є. Якщо блок default відсутній, відбувається вихід за межі оператора switch.
* Кожен блок case, як і блок default, в якому містяться виконувані оператори, повинен завершуватися оператором переходу break, або gotocase, return або throw
* Виконання порівняння значення виразу селектора з постійними значеннями операторів case починаються з першого оператора і триває по списку, зазвичай до досягнення оператора переходу. У цій точці управління передається за межі оператора switch або переходить до іншого оператора case, якщо оператори переходу або тіло оператора case були відсутні, і так до того оператора case, у якого буде присутнє тіло і оператор переходу. Така техніка називається проваленням.
* Блок default може бути створений в будь-якому місці тіла перемикача switch-case. Винятком є ​​тіло операторів case.
* Використання Convert.To ... () перетворює значення одного базового типу даних до іншого базового типу даних.
* Методи класу Convert використовуються для підтримки перетворення в базовий тип даних і з нього в .NET Framework. Підтримуються наступні типи даних: Boolean, Char, SByte, Byte, Int16, Int32, Int64, UInt16, UInt32, UInt64, Single, Double, Decimal, DateTime і String.

**Закріплення матеріалу**

* Чи обов'язково оператор if повинен використовуватись разом з оператором else?
* Чи обов'язково створювати блок default в перемикачі switch?
* Чи припустимо вкладення тернарних операторів?
* Значення якого типу можна передавати як параметр if ()?
* Чи обов'язково в перемикачі switch-case використовувати оператор переходу break?
* Для яких цілей можна використовувати метод Convert.To ... ()?
* Чи може switch-case мати тільки блок default?
* Що таке техніка провалювання в операторі switch-case?

**Дз**

Завдання 1

Використовуючи Visual Studio, створіть проект за шаблоном ConsoleApplication.

Напишіть програму українсько-англійський перекладач. Програма знає 10 слів. Потрібно, щоб користувач вводив слово українською мовою, а програма давала йому переклад цього слова на англійській мові. Якщо користувач ввів слово, для якого відсутній переклад, то слід вивести повідомлення, що такого слова немає.

Завдання 2

Вивчіть основні конструкції і поняття, розглянуті на уроці.

Завдання 3

Використовуючи Visual Studio, створіть проект за шаблоном Console Application.

Напишіть програму - консольний калькулятор.

Створіть дві змінні з іменами operand1 і operand2. Задайте змінним деякі довільні значення. Запропонуйте користувачеві ввести знак арифметичної операції. Прийміть значення введене користувачем і помістіть його в стрічкову змінну sign.

Для організації вибору алгоритму обчислювального процесу, використовуйте перемикач switch.

Виведіть на екран результат виконання арифметичної операції.

У разі використання операції ділення, організуйте перевірку спроби поділу на нуль. І якщо така є, то відмініть виконання арифметичної операції та повідомте про помилку користувачеві.

Завдання 4

Використовуючи Visual Studio, створіть проект за шаблоном Console Application.

Напишіть програму визначення, чи потрапляє вказане користувачем число від 0 до 100 в числовий проміжок [0 - 14] [15 - 35] [36 - 50] [50 - 100]. Якщо так, то вкажіть, в який саме проміжок. Якщо користувач вказує число, яке не входить ні в один з наявних числових проміжків, то виводиться відповідне повідомлення.

Завдання 5

Зайдіть на сайт MSDN.

Використовуючи пошукові механізми MSDN, знайдіть самостійно опис теми по кожному наприклад, який був розглянутий на уроці, так, як це представлено нижче, в розділі «Рекомендовані ресурси», опису даного уроку. Збережіть посилання і дайте їм короткий опис.

**Рекомендовані ресурси**

MSDN: Умовна конструкція if-else (Довідник C #)

http://msdn.microsoft.com/ru-ru/library/5011f09h.aspx

MSDN: Тернарний оператор (? :).

http://msdn.microsoft.com/ru-ru/library/ty67wk28.aspx

MSDN: Оператор множинного вибору (перемикач) switch-case.

http://msdn.microsoft.com/ru-ru/library/06tc147t.aspx