المحاضرة الرابعة الشرطي C#

البرمجة 1 – BPG401

SLASH TEAM

برنامج الهندسة المعلوماتية – الجـــــامعة الافتراضية السورية

تقديم	عدي الحلبي
وصف	أكواد الشرط الفائدة منها وتطبيقها. مسائل هامة جدا.
عدد الصفحات	12

الشرط في C#

توصيف المحاضرة

التوصيف:

سنتحدث في هذا الملف عن أكواد الشرط المنطقي وكيف نستخدمها ومتى ولماذا، متضمنين كافة أدوات الشرط في لغة C#. بالإضافة لمسائل هامة تبرز فائدة الشرط وطريقة استخدامه.

طريقة الدراسة:

كما في المحاضرات السابقة التجريب والتطبيق العملي للأكواد هو الطريقة الوحيدة لفهم الغرض من كل أداة وكل كود وحفظ تفاصيله. **من الهم جدا حل المسائل في اخر الملف**.

مقدمة

ما هو الشرط:

إن استخدام الشرط في البرمجة هو مفهوم أساسي ومهم جدا. يوفر لنا البرمجة بإمكانية التحكم في تدفق البرنامج واتخاذ القرارات بناءً على شروط معينة.

لماذا نستخدم الشرط في البرمجة؟

- **التحكم في تدفق البرنامج:** يتيح لنا اتخاذ قرارات وتنفيذ أوامر مختلفة بناءً على ظروف معينة.
- **الاستجابة للبيانات المدخلة:** يمكن للبرنامج الاستجابة بشكل مختلف استنادًا إلى البيانات التي يتلقاها.
- **المرونة والتخصيص:** يسمح لنا بكتابة برامج ديناميكية أكثر تتكيف مع حالات متعددة.

يمكن استخدام الشرط في مجموعة متنوعة من السيناريوهات، من التحقق من صحة المدخلات إلى إدارة مسارات البرنامج المختلفة بناءً على بيانات المستخدم. او وقوع أحداث معينة في مسار عم البرنامج.

IF, ELSE, ELSE IF:

الأمر if:

يُستخدم الأمر if للتحقق من شرط معين. إذا كان هذا الشرط صحيحًا، يتم تنفيذ الكود المحصور بين الأقواس المعقوفة {} وهي كتلة الشرط.

```
int number = 10;

if (number > 5)
{
    Console.WriteLine("The number is greater than 5.");
}
```

الأمر else if:

إذا كان لدينا أكثر من شرط نود التحقق منه، يمكننا استخدام الأمر else if. يُستخدم للتحقق من شرط آخر إذا كان الشرط الأول غير صحيح.

```
int number = 10;

if (number > 15)
{
    Console.WriteLine("The number is greater than 15.");
}
else if (number > 5)
{
    Console.WriteLine("The number is greater than 5 but less than or equal to 15.");
}
```

الأمر else:

إذا كانت جميع الشروط السابقة غير صحيحة، يتم تنفيذ الكود الموجود في الأمر else.

```
int number = 3;

if (number > 15)
{
    Console.WriteLine("The number is greater than 15.");
}
else
{
    Console.WriteLine("The number is less than or equal to 5.");
}
```

باستخدام هذه الأوامر الثلاثة، يمكنك إنشاء برامج تتفاعل بمرونة مع مختلف الحالات والظروف.

مثال عملي عن الشرط:

```
Console.Write("Enter the student's grade: ");
int grade = int.Parse(Console.ReadLine()); // يتم الادخال من قبل المستخدم والتحويل الى رقم مباشرتاً

if (grade >= 90)
    Console.WriteLine("Excellent");

else if (grade >= 80)
    Console.WriteLine("Very Good");

else if (grade >= 70)
    Console.WriteLine("Good");

else if (grade >= 60)
    Console.WriteLine("Satisfactory");

else
    Console.WriteLine("Fail");
```

قوم بالتحقق من قيمة الدرجة المدخلة باستخدام سلسلة من الأوامر الشرطية:

- إذا كانت الدرجة أكبر من أو تساوي 90، يتم طباعة"Excellent".
- إذا كانت الدرجة أكبر من أو تساوي 80 وأقل من 90، يتم طباعة "Very Good".
- إذا كانت الدرجة أكبر من أو تساوي 70 وأقل من 80، يتم طباعة "Good".
- إذا كانت الدرجة أكبر من أو تساوي 60 وأقل من 70، يتم طباعة "Satisfactory".
- إذا كانت الدرجة أقل من 60، يتم طباعة "Fail".

هذا المثال يظهر كيف يمكن استخدام الأوامر الشرطية للتحكم في تدفق البرنامج بناءً على القيم المدخلة.

هل لاحظت اننا لم نضع الأقواس المجعدة لتبين كتلة كل شرط؟

هذا ليس خطأ في حالة وجود كود واحد فقط داخل كتلة الشرط يمكن إزالة الأقواس المجعدة لاختصار وتقليل عدد أسطر الكود. وهذا ما يخرج لنا مفهوم غموض الشرط.

غموض كود الشرط (CONDITIONAL STATEMENT AMBIGUITY):

يشير إلى الحالات التي يمكن فيها أن تكون التعليمات الشرطية غير واضحة أو مفتوحة لتفسيرات متعددة، مما يؤدي إلى نتائج غير متوقعة أو غير صحيحة.

كما لاحظنا في المثال السابق انه يمكن إزالة الأقواس المجعدة {} في حال وجود كود واحد داخل كتلة الشرط.

لكن ماذا لو كان داخل الشرط شرط اخر كما يلي:

```
int age = 20;
if (age > 18)
    if (age == 20)
        Console.WriteLine("You are adult and your age is 20");
```

هل هذا صحيح؟

نعم صحيح حيث ان if وما بداخلها تعامل كأنها كود واحد ويمكنك تجربة ذلك.

لكن الغموض يظهر هنا:

```
int x = 5;
if (x > 10)
if (x == 11)
Console.WriteLine("x equal 11");
else
Console.WriteLine("x less than 10");
```

هل يمكنك تحديد تبعية else؟

هذا ما يسمى بغموض الشرط بشكل عام تتبع التعليمة الشرطية else لأقرب if إلا إذا حددت الأقواس عكس ذلك.

كيفية تجنب الغموض

لتجنب هذا الغموض، يجب علينا استخدام الأقواس المعقوفة {} بوضوح لتحديد الكتل الشرطية بوضوح:

```
int x = 5;
if (x > 10)
{
    if (x == 11)
        Console.WriteLine("x equal 11");
}
else
    Console.WriteLine("x less than 10");
```

الشكل IF ELSE:

يوجد شكل مختصر لكتابة شرط ونقيضه مثل التالي:

```
int x = 6;
if (x > 5) x--; else x++;
```

إذا كانت x أكبر من 5 فيتم إنقاص 1 من قيمتها. والا تكون أصغر ويتم زيادة 1.

الاسناد الشرطي (CONDITIONAL ASSIGNMENT):

هو تقنية تستخدم في البرمجة لتعيين قيمة لمتغير بناءً على شرط معين باستخدام تعبير شرطي مختصر تستخدم كما يلي:

(Condition) ? value : second value;

مثال:

```
int number = 10;
string result = (number % 2 == 0) ? "Even" : "Odd";
Console.WriteLine(result);
```

يتم استخدام هذا الشكل فقط في الاسناد كما في المثال السابق.

الأداة الشرطية SWITCH

مقدمة:

الأداة switch تُستخدم للتعامل مع عدة حالات محتملة لقيمة متغير معين، وتعتبر مناسبة عندما يكون لدينا العديد من الشروط التي تحتاج إلى التحقق بناءً على قيمة واحدة.

مميزات SWITCH مقارنة بـ IF

- **وضوح وسهولة القراءة**: عندما يكون لدينا العديد من الشروط التي تتعلق بنفس المتغير، فإن switch يوفر لنا طريقة أوضح لعرض هذه الشروط مقارنة بـ if.
- **الكفاءة:** في بعض الأحيان، تكون switch أكثر كفاءة من استخدام سلسلة من if، خاصة عندما يكون هناك العديد من الحالات التي تتعلق بنفس المتغير.
- **التنظيم:** باستخدام switch يمكننا تجنب التداخل الزائد والتكرار الذي قد يحدث عند استخدام if.

بنية SWITCH:

```
Switch (variable)
{
    case case_value: // يمكن للمتغير ان يكون نصي وان يتم وضع قيم نصية للحالات
        instructions;
        break; //to end case
    case case_value:
        instructions;
        break; //to end case
    default:
        instructions;
        break; //to end default case
}
```

مثال عملي

دعونا نأخذ مثالاً يوضح كيفية استخدام switch لتحديد اسم اليوم بناءً على رقم اليوم:

```
int dayOfWeek = 3;
switch (dayOfWeek)
{
    case 1:
        Console.WriteLine("Monday");
        break;
    case 2:
        Console.WriteLine("Tuesday");
        break;
    case 3:
        Console.WriteLine("Wednesday");
        break;
    case 4:
        Console.WriteLine("Thursday");
        break;
    case 5:
        Console.WriteLine("Friday");
        break;
    case 6:
        Console.WriteLine("Saturday");
        break;
    case 7:
        Console.WriteLine("Sunday");
        break;
    default:
        Console.WriteLine("Invalid day");
        break;
}
```

شرح الكود:

1. لدينا المتغير dayOfWeek الذي يحتوي على قيمة 3.
2. نستخدم الأداة الشرطية switch للتحقق من قيمة المتغير dayOfWeek.
3. بناءً على القيمة، يتم تنفيذ الكود الموافق للحالة. في هذا المثال، سيتم طباعة "Wednesday" لأن dayOfWeek يساوي 3.
4. نستخدم الأمر break لإنهاء تنفيذ الكود داخل كل حالة ومنع التداخل بين الحالات.
5. الحالة default تُستخدم للتعامل مع أي قيم غير متوقعة.

متى نستخدم SWITCH؟

- عندما يكون لدينا عدة حالات تعتمد على نفس المتغير.
- عندما تكون هناك حاجة لتحسين قابلية قراءة الكود وتفادي التداخل الزائد.
- عند التعامل مع حالات محددة ومعروفة مسبقًا.

مسائل عامة:

المسألة الأولى: تصنيف الأعداد

اكتب برنامجًا يقوم بقراءة عدد صحيح من المستخدم، ثم يحدد ما إذا كان العدد موجبا، سالبا، أو صفرا.

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int number = int.Parse(Console.ReadLine());

        if (number > 0)
        {
            Console.WriteLine("The number is positive.");
        }
        else if (number < 0)
        {
            Console.WriteLine("The number is negative.");
        }
        else
        {
            Console.WriteLine("The number is zero.");
        }
    }
}
```

المسألة الثانية: التحقق من الصلاحية

اكتب برنامجًا يقوم بقراءة عمر المستخدم وتحديد ما إذا كان يمكنه التصويت أم لا (يجب أن يكون العمر 18 أو أكثر للتصويت).

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter your age: ");
        int age = int.Parse(Console.ReadLine());

        if (age >= 18)
        {
            Console.WriteLine("You are eligible to vote.");
        }
        else
        {
            Console.WriteLine("You are not eligible to vote.");
        }
    }
}
```

المسألة الثالثة: تحديد نوع المثلث

اكتب برنامجًا يقرأ أطوال أضلاع مثلث من المستخدم، ثم يحدد نوع المثلث (متساوي الأضلاع، متساوي الساقين، أو مختلف الأضلاع).

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter side a: ");
        int a = int.Parse(Console.ReadLine());

        Console.Write("Enter side b: ");
        int b = int.Parse(Console.ReadLine());

        Console.Write("Enter side c: ");
        int c = int.Parse(Console.ReadLine());

        if (a == b && b == c)
        {
            Console.WriteLine("The triangle is equilateral.");
        }
        else if (a == b || b == c || a == c)
        {
            Console.WriteLine("The triangle is isosceles.");
        }
        else
        {
            Console.WriteLine("The triangle is scalene.");
        }
    }
}
```

المسألة الرابعة: الحسابات البسيطة

اكتب برنامجًا يقرأ رقمين وعملية حسابية (جمع، طرح، ضرب، قسمة) من المستخدم، وينفذ العملية الحسابية بناءً على المدخلات.

```
using System;
class Program
{
    static void Main()
    {
        Console.Write("Enter first number: ");
        double num1 = double.Parse(Console.ReadLine());
        Console.Write("Enter second number: ");
        double num2 = double.Parse(Console.ReadLine());
        Console.Write("Enter an operation (+, -, *, /): ");
        char operation = Console.ReadKey().KeyChar;
        //Enter اضافة من خارج المنهاج هذا الكود يقرا محرف واحد فقط ولا يطلب من المستخدم الضغط على
        Console.WriteLine();
        double result;
        switch (operation)
        {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 * num2;
                break;
            case '/':
                if (num2 != 0)
                {
                    result = num1 / num2;
                }
                else
                {
                    Console.WriteLine("Cannot divide by zero.");
                    return;
                }
                break;
            default:
                Console.WriteLine("Invalid operation.");
                return;
        }
        Console.WriteLine("The result is " + result + ".");
    }
}
```