

2023

Programación II

Control de errores

CONTROL DE ERRORES

Una excepción es un objeto derivado de System.Exception que se genera cuando en tiempo de ejecución se presenta un error y contiene información detallada del mismo.

El control de errores se realiza a través de la instrucción **try**, la cual permitirá capturar cualquier excepción presentada al intentar ejecutar el bloque de código correspondiente y brindarle un tratamiento adecuado.

Existen una cantidad considerable de clases derivadas de System.Exception que le permitirán al desarrollador tomar medidas específicas de acuerdo al tipo de error detectado, algunas de ellas son:

Excepción	Situación ante la cual se presenta
DivideByZeroException	División por cero
IndexOutOfRangeException	Índice de acceso a elemento de tabla fuera del rango válido (menor que cero o mayor que el tamaño de la tabla)
OverflowException	Desbordamiento
StackOverflowException	Desbordamiento de la pila, generalmente debido a un excesivo número de llamadas recurrentes.

Instrucción Try	
Sintaxis	<pre> try { //Bloque de código que se intentará ejecutar } catch (Excepcion1) { //Bloque de código a ejecutar ante el surgimiento de un error //de tipo Excepcion1 en el bloque try } catch (Excepcion2) { //Bloque de código a ejecutar ante el surgimiento de un error //de tipo Excepcion2 en el bloque try } . . . finally { //Opcional: //Bloque de código a ejecutar luego de los bloques try/catch } </pre>
Ejemplo	<pre> Try { Console.WriteLine("Resultado es: {0}", n1/n2); } catch (DivideByZeroException E) { Console.WriteLine("Error: {0}", E.Message); } finally { Console.WriteLine("Operación Finalizada."); } </pre>

Existen situaciones en las que es necesario generar una excepción propia del módulo que se está desarrollando. Para lanzar una excepción se recurre a la utilización de la instrucción **throw**.

El siguiente ejemplo se presenta solo con fines explicativos. En el mismo se genera una excepción vinculada con la División por Cero, siendo que CSharp cuenta con una excepción para esta situación denominada DivideByZeroException.

Ejemplo

```
static void Main(string[] args)
{
    string Cadena;
    float Num1, Num2;
    char Oper;    float
    Resultado;

    Console.WriteLine("Ingreso Primer Numero");
    Cadena = Console.ReadLine();
    Num1 = float.Parse(Cadena);

    Console.WriteLine("Ingreso Segundo Numero");
    Cadena = Console.ReadLine();
    Num2 = float.Parse(Cadena);

    Console.WriteLine("Ingreso Operador");
    Cadena = Console.ReadLine();
    Oper = char.Parse(Cadena);

    switch (Oper)
    {
        case '+':
        {
            Resultado = Sumar(Num1, Num2);
            Console.WriteLine("El Resultado es: {0}", Resultado);
            break;
        }
        case '-':
        {
            Resultado = Restar(Num1, Num2);
            Console.WriteLine("El Resultado es: {0}", Resultado);
            break;
        }
        case '*':
        {
            Resultado = Multiplicar(Num1, Num2);
            Console.WriteLine("El Resultado es: {0}", Resultado);
            break;
        }
        case '/':
        {
            try
            {
                Resultado = Dividir(Num1, Num2);
                Console.WriteLine("El Resultado es: {0}", Resultado);
            }
            catch (Exception e)
            {
            }
        }
    }
}
```

```
        Console.WriteLine("Error: {0}", e.Message);
        Console.WriteLine("Por ayuda recurrir a: {0}", e.HelpLink);
        Console.WriteLine("Software que genero el error: {0}", e.Source);
    }
    break;
}
}
Console.ReadKey();
}

static float Sumar(float N1, float N2)
{
    return N1 + N2;
}

static float Restar(float N1, float N2)
{
    return N1 - N2;
}

static float Multiplicar(float N1, float N2)
{
    return N1 * N2;
}

static float Dividir(float N1, float N2)
{
    if (N2 != 0)
    {
        return N1 / N2;
    }
    else
    {
        Exception Excep = new Exception("Intento de division por cero.");
        Excep.HelpLink = "Consultar Fundamentos Matematicos.";
        Excep.Source = "Programa Calculadora";
        throw Excep;
    }
}
```