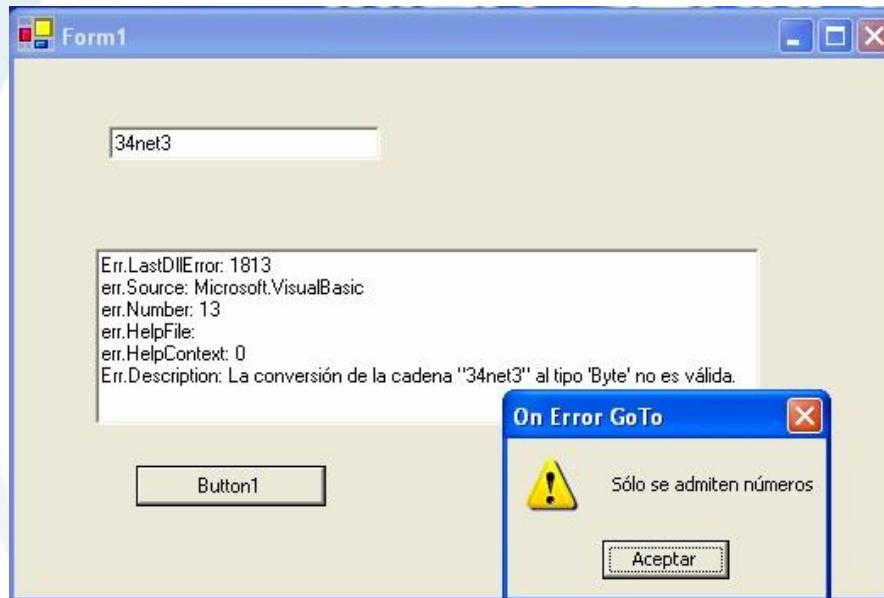


# Excepciones



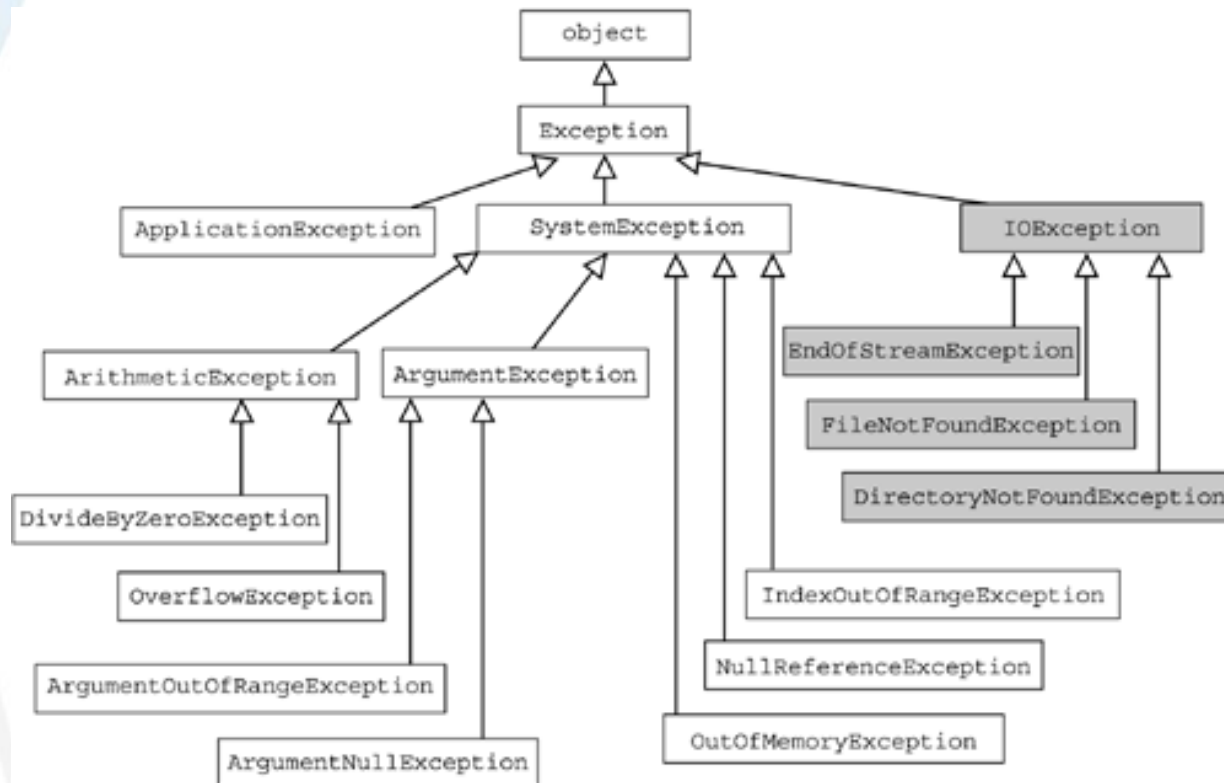
# Manejo de excepciones

- El manejo de excepciones es una técnica de programación que permite afrontar cualquier situación inesperada o excepciones que produce un programa cuando se ejecuta.



# Tipo de excepciones

- Las clases derivadas de **Exception** representan estas excepciones. Esta clase identifica el tipo de excepción y contiene las propiedades que albergan detalles acerca de esta.



# Lista de excepciones

Excepción	Causa
System.OutOfMemoryException	Se lanza cuando falla un intento de reservar memoria mediante el operador <i>new</i> .
System.StackOverflowException	Se lanza cuando la pila de ejecución se llena por tener demasiadas llamadas a métodos pendientes. Normalmente ocurre cuando son métodos recursivos profundos.
System.NullReferenceException	Se lanza cuando se pretende tener acceso mediante a una referencia <i>null</i> aun objeto.
System.TypeInitializationException	Se lanza cuando un constructor <i>static</i> lanza una excepción y no existe una clausula <i>catch</i> que la capture.
System.InvalidCastException	Se lanza cuando falla en tiempo de ejecución una conversión de un tipo base o interface a un tipo derivado.
System.ArrayTypeMismatchException	Se lanza cuando se intenta almacenar un elemento de un tipo no adecuado en un array.
System.IndexOutOfRangeException	Se lanza cuando se intenta acceder a un array mediante un índice menor que cero o mayor que el limite del array
System.MulticastNotSupportedException	Se lanza cuando se intenta combinar dos delegates no nulos, debido a que el tipo delegate no tiene como tipo de retorno <i>void</i>
System.ArithmeticException	En la clase base de las excepciones que se dan durante las operaciones aritméticas, tales como <b>DivideByZeroException</b> y <b>OverflowException</b>
System.DivideByZeroException	Se lanza cuando se intenta dividir un valor de tipo integral por cero.
System.OverflowException	Se lanza cuando una operación aritmética causa un desbordamiento en un contexto <i>"checked"</i> .

# Try, Catch, Finally

- El manejo de excepciones usan las palabras reservadas ***try***, ***catch*** y ***finally*** permiten la manipulación de excepciones logrando ejecutar un camino alternativo o mostrar algún mensaje en caso de ocurrir un error en tiempo de ejecución

```
static void TestCatch2()
{
    System.IO.StreamWriter sw = null;
    try
    {
        sw = new System.IO.StreamWriter(@"C:\test\test.txt");
        sw.WriteLine("Hello");
    }

    catch (System.IO.FileNotFoundException ex)
    {
        // Put the more specific exception first.
        System.Console.WriteLine(ex.ToString());
    }

    catch (System.IO.IOException ex)
    {
        // Put the less specific exception last.
        System.Console.WriteLine(ex.ToString());
    }
    finally
    {
        sw.Close();
    }

    System.Console.WriteLine("Done");
}
```