# Comentarios

// (doble slach) sierve para realizar comentarios en una sola línea

/\* (slach multiplicación signo) sirve para comentar una o mas líneas

Comentar sirve para que al leer el algoritmo un tercero este sepa que se realiza en tal línea o bloque de codigo

# Identificadores

Son los nombres que se usan para identificar los elementos del programa:

* namespaces
* clases
* métodos
* variables
* constantes

# Reglas o convenciones

* Solo se pueden usar:
  + letras (mayúsculas o minúsculas)
  + números
  + guiones bajos (“\_”)

ej.

Primera\_Aplicacion (correcto)

Primera-Aplicación (incorrecto)

* las definiciones de variables, clases, namespaces, contastantes deben comenzar por una letra o un guion bajo

ej.

PrimeraAplicacion2 (correcto)

2primeraApliacion (incorrecto)

* no se deben utilizar palabras claves o reservadoas para nombrar variables, namespaces, clases, contantes

***def palabras claves o reservadas: son las que ocupa el lenguaje para sus propios propósitos***

ej.

Class namespaces (incorrecto)

Class algo (correcto)

# Buenas prácticas

1. lo ideal es dejar un espacio vertical entre bloques, es decir un espacio entre cada línea de código.
2. Tabular el código por bloques

## Variables

1. No comenzar el nombre de una variable con un guion bajo
2. No crear más de una variable que se diferencien por una sola letra.

Ej. Edad, edad, EDAD

1. Comenzar el nombre de una variable por minúsculas
2. Si el nombre de la variable está compuesto por más de una palabra es recomendado usar la estructura “CAMEL CASE”. EJ radioDelCirculo
3. No utilizar notación húngara, esto se refiere a no definir las variables de la siguiente manera “bedad” que es la mezcla del tipo de dato byte y el nombre de una variable en este caso la variable edad

# Tipos de datos

Existen dos grandes tipos de datos estos son los por valor y los por referencia

Por valor tenemos a los siguientes tipos de datos

Por valor:

* Enteros:

Estos a su vez se dividen en:

* + Sin signo:
    - Byte
    - Ushort
    - Uint
    - Ulong
  + Con signo:
    - Sbyte
    - Short
    - Int
    - Long
* Reales

Son todos los números que incluyan una parte decimal

* + float
  + double
  + decimal
* booleanos:

valores lógicos para toma de decisiones dentro del programa

* + True : para valores verdaderos
  + False: para valores falsos
* Letras

Son como dice su nombre todos los caracteres de tipo str o char (UTF 16)

# Variables

La declaración de una variable en C# se realiza de la siguiente manera y siempre se termina la ninea con un punto y coma “;”

Tipo de dato + el nombre de la variable;

Ejemplo

Int edad;

Al declarar una una variable esto no quiere decir que esta inicializada para realizar esto bien se puede realizar en la misma line que se define o en algún otro lugar del programa.

Ej

Declaración e inicialización

Int edad = 28;

Declaración y posterior inicialización

Int edad;

Edad = 28;

***Consideración: en C# no se pueden utilizar variables no inicializadas.***

# Operadores aritméticos

Operaciones básicas

* Suma (+) puede ser utilizado tanto para números como para str
* Resta (-) puede ser utilizado tanto para números como para str
* Multiplicación (\*)
* División (/)
* Modulo (%)

Incremento (++)

Decremento (--)

Conjugaciones de símbolos:

* Suma (+) 🡪 +=
* Resta (-) 🡪 -=

Nota:

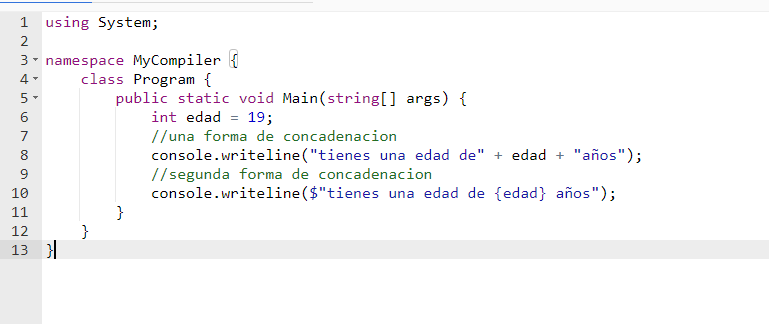
Al dividir el en C# siempre entregara el resultado en el tipo de dato esta la variable o el número.

Ej al dividir 5 en 2 debería dar 2.5 pero en C# nuestra por consola 2, esto debido a que ambos son números enteros por tanto el lenguaje respeta esto, a diferencia de otros lenguajes

Entero/entero = entero

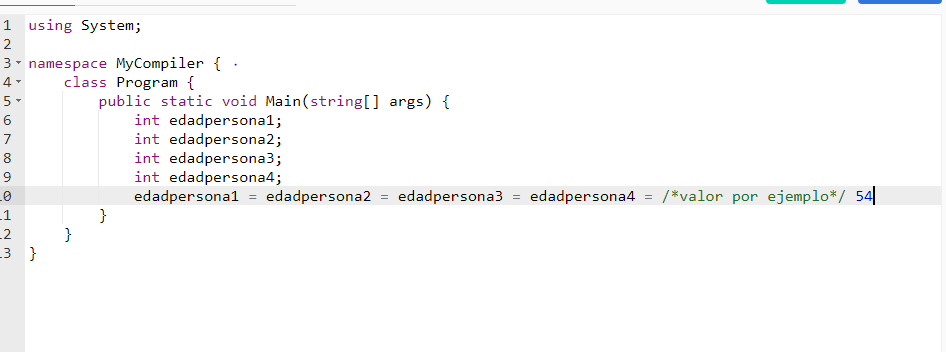
Entero/decimal = decimal

Concatenación de string



Comportamiento “=”

Puede servir para asignar un valor a múltiples variables



No se recomienda hacer esto de esta forma ya que conduce a errores

Int edadpersona1, edadpersona2, edadpersona3, edadpersona4 = 27;

Ya que la única variable que se inicializa es edadpersona4 y las otras 3 no

# Declaración implícita