10 de marzo de 2017

Emiliano Montesdeoca del Puerto

1daw b

Cesar Manrique – Entornos de Desarrollo

REfactorizacion

# Indice

1. Índice
2. Enunciado.
3. Código original.
4. Código refactorizado.
5. Cambios en el código.

# Enunciado

Aplicar todos los métodos conocidos de refactorización y tratados en el archivo refactoriza.pdf a este código, para obtener un código optimizado, más claro, flexible y fácil de manejar.

Posteriormente elaborar un informe en donde aparezca el código antiguo, el optimizado y la explicación de los cambios efectuados.

# Código original

class Program

{

static void Main(string[] args)

{

int op;

double resu;

int num1;

Console.Clear();

Console.WriteLine("1.- Funcion1");

Console.WriteLine("2.- Funcion2");

Console.WriteLine("3.- Resultado");

Console.WriteLine("0.- Salir");

Console.Write("Opción: ");

op = Convert.ToInt32(Console.ReadLine());

while (op != 0)

{

switch (op)

{

case 1:

resu = funcion1();

Console.WriteLine();

Console.WriteLine("La visualización del resultado

es");

Console.WriteLine("---------------------------------

");

Console.WriteLine("Para ello tenemos que visualizar

los valores");

Console.WriteLine("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", resu);

Console.ReadLine();

break;

case 2:

Console.Write("\nIntroduzca num1: ");

num1 = Convert.ToInt32(Console.ReadLine());

resu = funcion2(num1);

Console.WriteLine();

Console.WriteLine("La visualización del resultado

es");

Console.WriteLine("---------------------------------

");

Console.WriteLine("Para ello tenemos que visualizar

los valores");

Console.WriteLine("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", resu);

Console.ReadLine();

break;

case 3:

resu = (3.1415 \* 2 - 1) / 3.1415;

Console.WriteLine();

Console.WriteLine("La visualización del resultado

es");

Console.WriteLine("---------------------------------

");

Console.WriteLine("Para ello tenemos que visualizar

los valores");

Console.WriteLine("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", resu);

resu = (3.1415 \* 3 - 1) / 3.1415;

Console.WriteLine();

Console.WriteLine("La visualización del resultado

es");

Console.WriteLine("---------------------------------

");

Console.WriteLine("Para ello tenemos que visualizar

los valores");

Console.WriteLine("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", resu);

resu = (3.1415 \* 4 - 1) / 3.1415;

Console.WriteLine();

Console.WriteLine("La visualización del resultado

es");

Console.WriteLine("---------------------------------

");

Console.WriteLine("Para ello tenemos que visualizar

los valores"); Console.WriteLine

("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", resu);

Console.ReadLine();

break;

}

Console.Clear();

Console.WriteLine("1.- Funcion1");

Console.WriteLine("2.- Funcion2");

Console.WriteLine("3.- Resultado");

Console.WriteLine("0.- Salir");

Console.Write("Opción: ");

op = Convert.ToInt32(Console.ReadLine());

}

}

static double funcion1()

{

double resu;

resu = 3.1415 + 3.1415;

return (resu);

}

static int funcion2(int num1)

{

if (num1 < 0)

for (int i = 0; i < 5; i++)

num1 = num1 - i;

else if (num1 == 0 || num1 == 1 || num1 == 2 || num1 == 3 || num1

== 4)

for (int i = 0; i < 5; i++)

num1 = num1 - i;

else if (num1 == 5 || num1 == 6 || num1 == 7)

for (int i = 0; i < 5; i++)

num1 = num1 - i;

else

num1 = num1 \* 2;

return (num1);

}

}

}

# Código refactorizado

namespace Refactorizacion

{

class Program

{

const double pi = 3.1415;

static void Main(string[] args)

{

Calculos();

}

static void Calculos()

{

int a;

do

{

switch (a = MostrarMenu())

{

case 1:

TextoResultado(Funcion1().ToString());

Console.ReadLine();

break;

case 2:

Console.Write("\nIntroduzca num1: ");

TextoResultado(Funcion2(Convert.ToInt32(Console.ReadLine())).ToString());

Console.ReadLine();

break;

case 3:

for (int i = 2; i <= 4; i++)

{

TextoResultado(((pi \* i - 1) / pi).ToString());

}

Console.ReadLine();

break;

}

} while (a != 0);

}

static double Funcion1()

{

return (pi \* 2);

}

static int Funcion2(int num1)

{

int aux = num1;

if (aux < 8)

for (int i = 0; i <= 5; i++)

aux = aux - i;

else

aux = aux \* 2;

return (aux);

}

static void TextoResultado(string txt)

{

Console.WriteLine();

Console.WriteLine("La visualización del resultado es");

Console.WriteLine("---------------------------------");

Console.WriteLine("Para ello tenemos que visualizar los valores");

Console.WriteLine("--------------------------------------------");

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Resu: {0}", txt);

}

static int MostrarMenu()

{

Console.Clear();

Console.WriteLine("1.- Funcion1");

Console.WriteLine("2.- Funcion2");

Console.WriteLine("3.- Resultado");

Console.WriteLine("0.- Salir");

Console.Write("Opción: ");

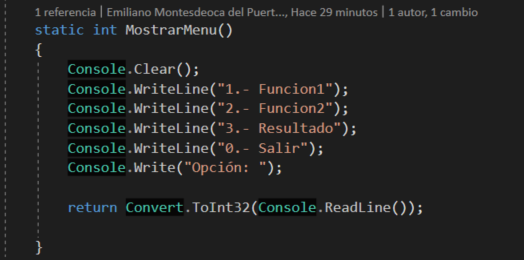
return Convert.ToInt32(Console.ReadLine());

}

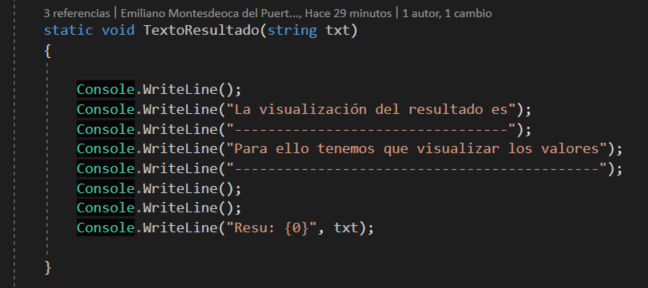
}

}

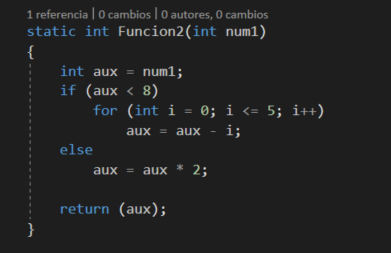
# Cambios en el código



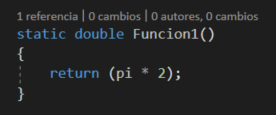
Se ha creado una funciona donde se encapsula el texto del menú y también pregunta sobre la opción que desea realizar, y devuelve un entero con la opción elegida.



Se ha creado un método que muestra la información a la hora de mostrar el resultado y los valores, en este método se recogen por valor el resultado de la operación.

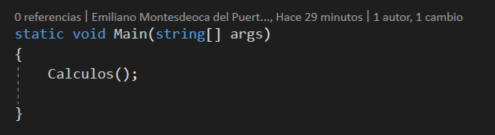


Se ha modificado la función **funcion2**, que tenía más código del necesario, se han eliminado los IF innecesarios y tratado la variable introducida con un auxiliar.

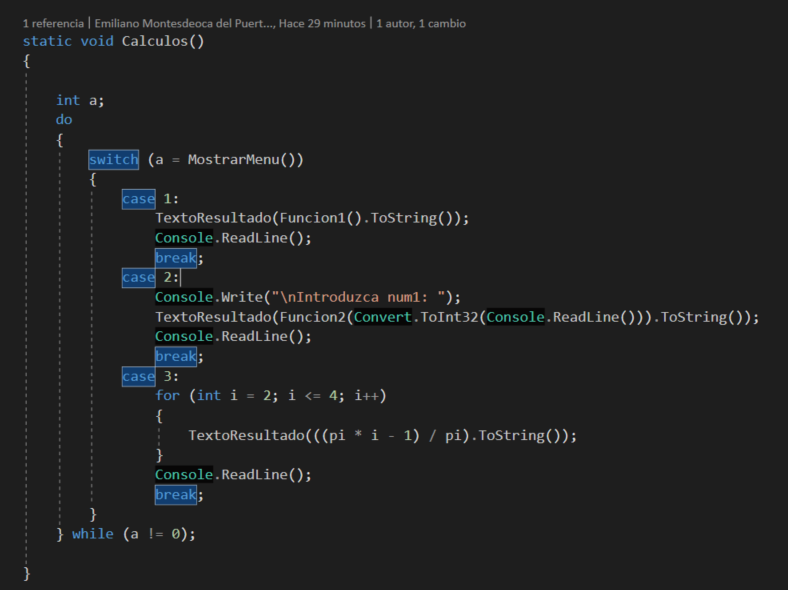




Se ha modificado la función **funcion1**, que tenía más código del necesario, se ha remplazado el doublé por una constante definida en la clase.



Se ha quitado el código principal del método Main y se ha incluido en un método publico llamado **Calculos**.



En la función principal se ha cambiado la forma en la que se utiliza el menú, se ha introducido un **do while** que solo finaliza si la opción introducida a la función **MostrarMenu** devuelve un 0.

En cuanto al caso 1, se ha actualizado llamando a la función **TextoResultado**, pasándole el resultado de la **Funcion1**.

En cuanto al caso 2, se ha actualizado llamando a la función **TextoResultado**, pasándole el resultado de la **Funcion2**, que a su vez le pasamos por parámetro en pantalla lo que introducimos.

En cuanto al caso 3, cambiamos por un bucle **for** que empieza en 2 y acaba en 4. Cada vez que entra llama a la función **TextoResultado** pasándole una operación.