C# Summery part-1 By Mahmoud Badawy

-----------------------------------------------------

ملخص سى شارب الجزء الاول

الملخص يحتوى على

- Basics

- Some C# Built in Functions

الفيديوهات المستخدمه فى الشرح   
<https://www.youtube.com/channel/UCgRFf3_D5H1Qi8pvw2Czyzg/playlists>

MB

##################################################################

اول حاجه كده هى اذاى احمل البرنامج الى هشتغل عليه و اذاى هسطبه ؟  
اولا لازم نبقى متفقين ان اللغه الى بنكتبها دى عشان تترجم لكلام مفهوم للكمبيوتر و يعرف يحولها لبرامج   
لازم يبقى عندى برنامج بكتب قىه الاكواد و هو الى بيحولها لبرامج بقه تمام   
طيب احنا بقه هنشتغل مع الفيجوال بيزك و مع السى شارب و مع السى بلاس بلاس و مع لغات تانيه كتير ب برنامج واحده اسمه

### Visual Studio

هو ده البرنامج الى هنشتغل بيه حمل اى اصدار منه عندك من اول ما اخترعوه لحد اصدار السنه دى و ياريت يبقى   
Ultimate   
Professional   
Enterprise

عشان دى بتبقى اصدارات كامله

لينكات تحميل مباشر اهه يا جماعه عشان الى هيحمل و يشتغل بقه

و لو انى بحب 2012 عشان حجمه صغير و هتلاقوا سريالات (مفاتيح) ليه على النت كتير جدا   
--------------------------------------------------------------------------------------

**Visual Studio 2012**

المفضل بالنسبالى هو ده الالتميت ده و هتلاقولوا سيريالات بالكوم على النت و كامل و الحمد لله و مساحته معقوله  
<http://download.microsoft.com/download/d/b/0/db03922c-ff91-4845-b7f2-fc68595ab730/vs2012_ult_enu.iso>  
  
  
او  
  
<http://download.microsoft.com/download/7/8/8/78863D92-FAA5-4692-8B51-381901E9BE7F/VS2012.2.iso>  
  
  
من الاخر الصفحه دى فيها كل اصدارات 2012   
<http://www.unlockwindows.com/download-visual-studio-2012-rc/>  
 الى تحب تحمله دوس على   
Image ISO  
 و هتلاقيه اشتغل تحميل

--------------------------------------------------------------------------------------

**Visual Studio 2013**

<http://www.myhow2.net/wp/2014/04/microsoft-visual-studio-2013-professional-ultimate-version-dvd-iso-free-download/>

--------------------------------------------------------------------------------------  
و ده الاكسيبريس برضوا اصدار كويس بس ناقص حاجات كتير هى حاجات معظمها مش هتفيدنا بس لو حد عاوز يشتغل بيه عشان مساحته قليله جدا و خفيف مفيش مانع   
  
[https://extr3metech.wordpress.com/2012/10/09/microsoft-visual-studio-2012-express-offline-install/](https://extr3metech.wordpress.com/2012/10/09/microsoft-visual-studio-2012-express-offline-install/%20)   
  
و معاه كمان شرح اذاى يتسطب يعنى حاجه ذى الفل و الله ههههههههه

وده 2015 كمان جميع الاصدارات

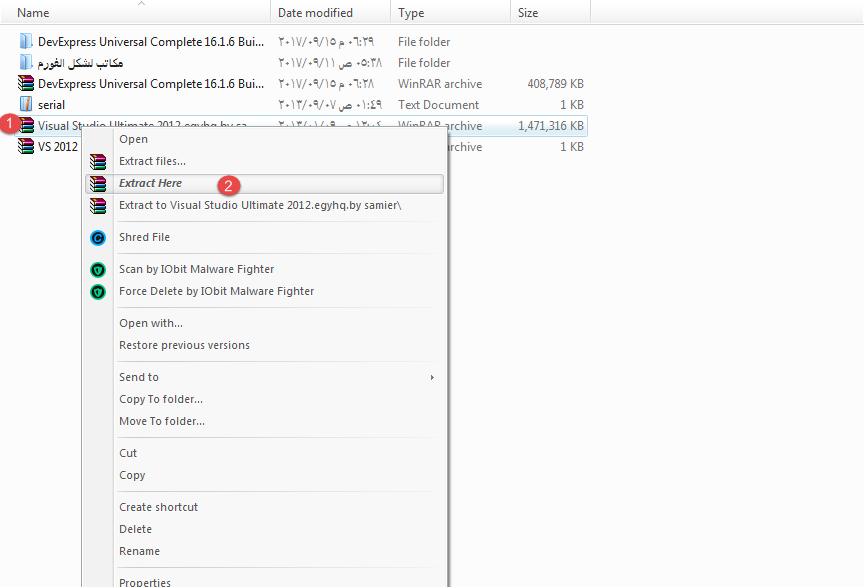
<http://www.kunal-chowdhury.com/2015/07/download-visualstudio-2015.html>

MB

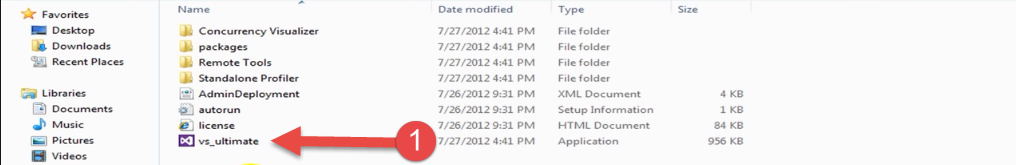
#######################################################################

اذاى بقه نسطبه

1. اول حاجه هتفك الضغط بتاعه هيظهرلك مجموعه ملفات

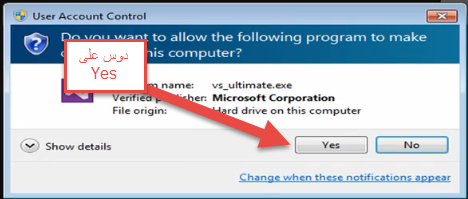


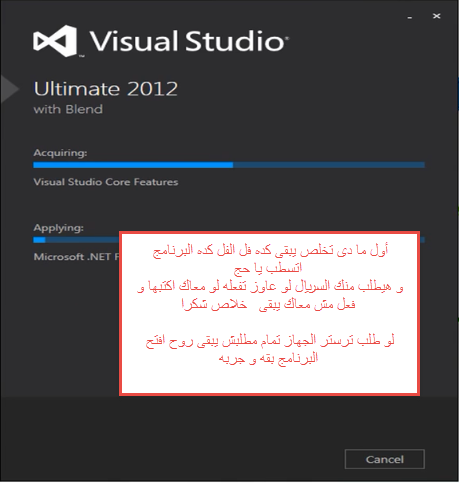
1. تانى حاجه نبدا نسطب بقه ندوس على تسطيب الابليكيشن بقه

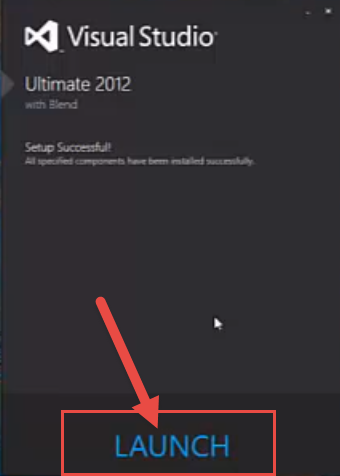


1. يبدأ من هنا التسطيب الفعلى و ده هنشرحه بالصور الى جايه بقه   
     
   



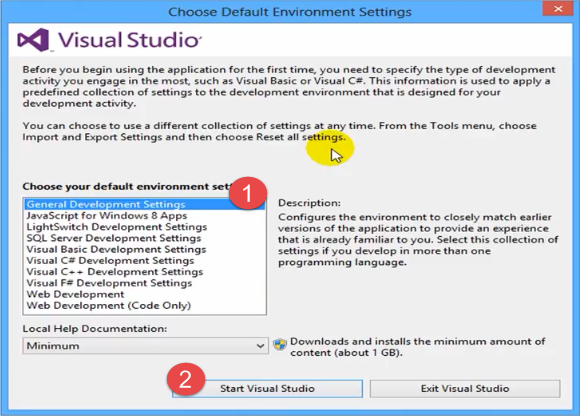


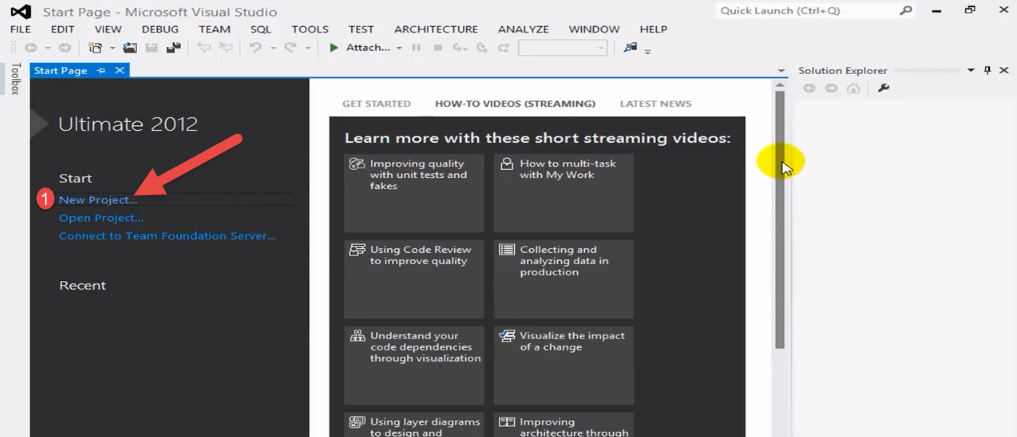


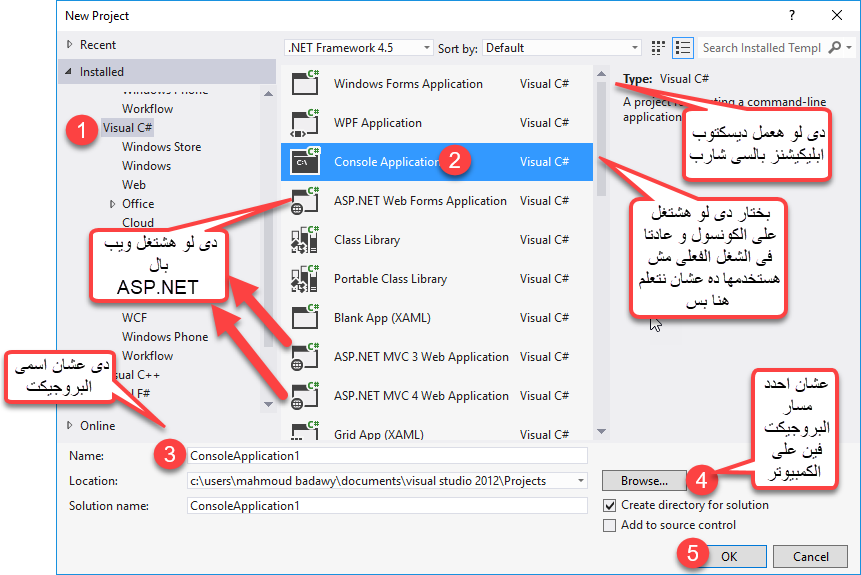


هيفتحلك قائمه كده تحط فيها السيريال نمبر  
 لو معاك لو مش معاك دوس   
Cancel   
 معاك حطها و دوس   
Next



روح افتح البرنامج بقه هيظهرلك اول مره اختيارات  






MB

########################################################################

البنيه الاساسيه لاى برنامج فى ال C#

-------------------------------------

using System;

namespace CSharpConsAppTest

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello Mahmoud Badawy");

}

}

}

Using System;  
دى الليبرارى او الدى ال ال او النيم سبيس لمشروع تانى الى انا بستخدمها فى المشروع ده من بره و هى عباره عن مجموعه من الكلاسات او الانترفيسيس او الدليجيكتس بداخلها فانكشنز و بنستدعيهم فى البروجيكت هنا وقت ما بنحب و بستخدمها عشان انا هستخدم كلاس فيها اسمه كونسول فيه فانكشن اسمها رايت لاين عشان اطبع على الكونسول

namespace CSharpConsAppTest  
دى بقه عباره عن المكان الى بيحتوى على مجموعه من كلاسيس او انترفيسيس او ميثودس او سب نيم سبيسيس كمان

Console.WriteLine("Hello Mahmoud Badawy");

كونسول دى عباره عن كلاس داخل النيم سبيس الى اسمها سيستم و جواها مجموعه من الفانكشنز و الى الرايت لاين واحده منهم و الفانكشن دى الخاصه بطباعه النص على الكونسول و بتعمله فى سطر لوحده

* class Program

دى ياعمنا بقه الكلاس الى انا شغال فيها حاليا

- static void Main(string[] args)

دى بقه المين فانشكن الى هى الفانكشن الاساسيه فى البروجيكت و الى منها بيبدأ البروجيكت يشتغل و استحاله يبدأ شغل من مكان تانى غير من فانكشن المين دى

MB

########################################################################

output in Console

-----------------------

عن طريق الكونسول دوت رايت

Console.Write("Hello World");

دى بقه عامل ذى السى اوت فى السى بلاس بلاس

عن طريق الكونسول دوت رايت لاين اقدر اعمل الاوت بوت فى الكونسول

Console.WriteLine("Hello World");

خد بالك ان رايت لاين بيحجز سطر كامل يعنى لو عملت رايت لاين كمان هينزل السطر الى بعده يعنى عامل ذى السى اوت الى اخره اند لاين

ملحوظه هامه

الران العادى فى الفيجوال ستوديو بستارت او F5

الديباج بقه ب CTRL + F5 و دى الى بفضلها عشان توقفلك الشاشه ( مهم جدا ) و متختفيش بسرعه

MB

########################################################################

Input in Console

-----------------------

اولا بتشوف انت هتدخل اينبوت نوعه ايه و ليكن استرينج بتروح معرف فاريبول استرينج عشان تخزن فيه الاسترينج الى انت هتكتبه ده و بعد كده بتقوله ان الفاريبول ده بيساوى كونسول دوت ريد لاين

string Name = Console.ReadLine();

و ده معناه اقرأ الى اتكتب فى الكونسول و هاتوا حطه فى الميمورى فى الفاريبول الى من نوع سترينج الى اسمها Name

طب اشرح الكود ده

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Your name");

string Name = Console.ReadLine();

Console.WriteLine("Hello : " + Name);

}

}

بص يا سيدى هنا بيقوله دخل اسمك فراح فعلا كاتب اسمه راح البرنامج قاريه من الكونسول و حطه فى فاريبيول

من نوع سترينج الى هو نفس النوع الى انا محتاج ادخله و لما كاتبه فى الفاريبول بقه احتاج يكتب ازيك يا حج كذا   
فيعملها اذاى دى الطريقه الاولى الى هى عن طريق الكونكاتيونيشن الى بتجمع الفاريبول بالاسترينج عن طريق   
علامه + و فى الطريقه التانيه الى هى عن طريق الاسترينج فورمات و شكلها بيبقى كانك بتحط انديكسيس و كل الى بيحصل ان بيتم التعويض عنها بالقيم الى بتديهاله بعد الكوما نفس فكره الاسترينج فورمات فى الفيجوال بيزيك

Console.WriteLine("Hello : {0} " , Name);

MB

########################################################################

Data Type for Varibales   
نوع البيانات للمتغيرات   
----------------------------

عندك أنواع كتير بس انا هشرح الاساسى بس   
اول حاجه

Bool

و دى ذى ما كلنا عارفين الى هى بوليان و دى بتحظن فيها حاجه من اتنين يا ترو يا اما فولس يا صح يا غلط   
True / False

Int  
بتخزن فيها الارقام الصحيحه فقط   
Float  
بتخزن فيها الارقام العشريه و الكسور و كده   
String

بتحزن فيها الكلام و الجمل   
Char   
بتحزن فيها الحروف   
يمكن معرفه كل شئ عن الداتا تايب بالتفصيل الممل من هنا   
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/built-in-types-table>

<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/types-and-variables>

طيب اذاى اكتب تعريف فاريبول و اذاى اكتب تعريفه و اديله قيمه فى نفس الوقت   
بص تعرفه بس فى سطر و تديله فيمه فى سطر كده

string Name ;

Name = "Mahmoud";

طيب تعرفه و تديله القيمه فى سطر واحد اذاى

string Name = "Mahmoud";

فيه بقه حاجه اسمها var بحط فيها اى حاجه انا عاوزها و هى بتفهمها

var Name = "Mahmoud"; // string

var Inum = 0; //int

var Fnum=0.0; // double or float

دلوقتى هو مش عارف هو فلوت ولا دوبل بس عمتا الاتنين ذى بعض الفرق فى المساحات انا بقه فى حاجه ذى دى علشان مشغلش دماغى بيها فى حروف بتتكتب بعد الارقام بتعرف ال var الرقم نوعه ايه بالظبط

namespace Test

{

class Program

{

static void Main(string[] args)

{

var intNum = 0; // int = int32

var FloatNum = 0f; // float

var DoubleNum = 0d; // double

var DecimalNum = 0m; // decimal

var LongNum = 0L; //long = int64

var uintNum = 0u; // Uint = Uint32

var uLongNum = 0ul; //Ulong = Uint64

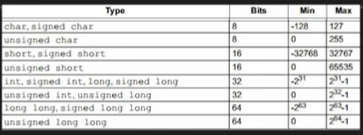
Console.ReadKey();

}

}

}

الفروقات بين data types



MB

########################################################################

Value Type Vs Refrence Type  
---------------------------------

عندنا بقه انواع من ال Data type من نوع ال Value Type و نوع تانى من نوع Refrence Type

طيب ايه الفرق بينهم ؟؟

ال Value Type ذى مثلا ال ال

Bool , byte , char, decimal, double , enum , float , int , long , sbyte , short , struct , uint , ulong , ushort

و دى يا سيدى الميكانيزم بتعها ان اسمها و قيمتها بيتسجلوا فى ال Stack

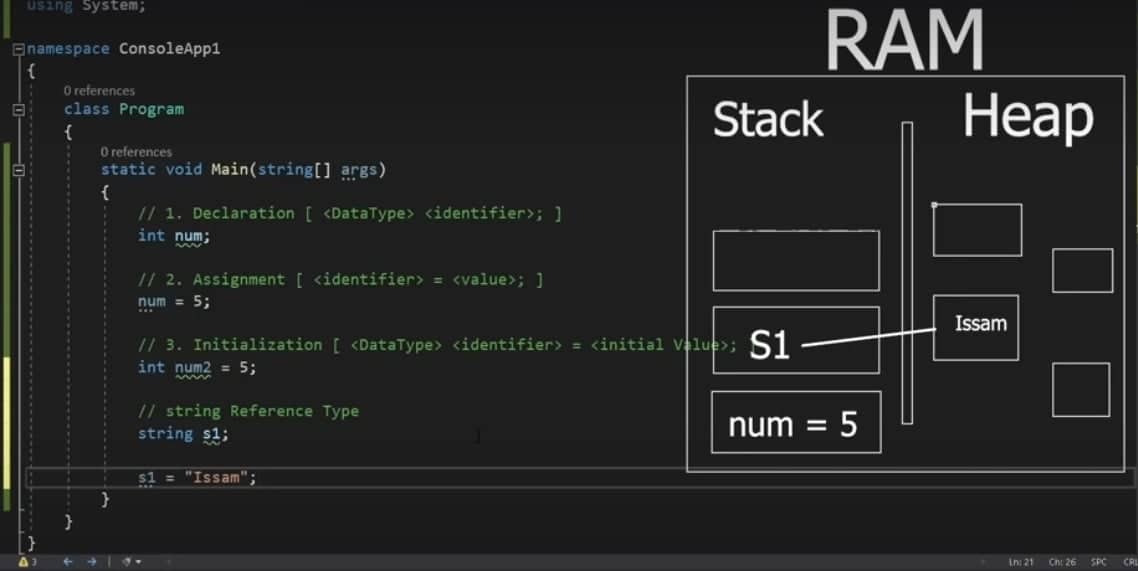
يعنى لما باجى اعرفها ذى ما فى الابلكيشن ده كده

Int16 Num = 12345; // Decleration and assignation Value is in Stack

بينما ال Refrence Type بيكون التعريف او الدكليريشن بتعها فى ال Stack و بيشاور على القيمه الى موجوده فى ال Heap و من اهم الانواع الى من نوع Refresnce Type هى string , Array , List , Objects

string Num = "Mebo"; // Decleration in stack and assignation Value is in Heap

و الصوره دى هتوضح اكتر للناس علشان تفعم طريقه التخزين



ذى ما احنا شايفين الانتيجر اتسجل الدكليريشن بتاعه الى هو ال num فى Stack و كذلك القيمه الخاصه بيه الى هى 5 فى ال Stack  
اما الاسترينج بقه راح اتسجل الدكليريشن بتاعه الى هو S1 فى Stack و راح رايح مخزن فيمته فى مكان تانى اوسع فى المساحه و اسرع اسمه ال Heap

الفرق التانى بقه يا معلم و المهم   
اولا ال Value Type انت لما بتديله قيمه بياخدها بشكل مباشر القيمه و ليس عنوان القيمه وبالتالى لو عمل للقيمه دى اسين على فاريبول جديد فهو بياخد القيمه من الفاريبول الاول برضوا بشكل مباشر و ليس عنوان القيمه و بالتالى لو فرضنا ان الفاريبول الاصلى غير القيمه بعد كده فالفاريبول الى اخد القيمه القديمه منه مش هيتاثر خالص و العكس صحيح

بالتالى طريقه التخزين اشبه ب ان   
واحد راح يتصور فى الاستوديو 2 صور و لما اخد ال 2 صور الورق دول راح مدى صوره لصحبه يعينهاله معاه و صوره لنفسه فبقى كل واحد منهم معاه الصوره دى الى هى ستايل كاجول هنقول قيمتها 5 مثلا فكده القيمه 5 دى مع الراجل و مع صاحبه برضوا

جه بعد ساعه لما صاحبه مشى و اخد الصوره بتعته معاه قعد يفكر و راح الراجل ده قالك الصوره الى معايا دى مش عجبانى و قطعها و راح اتصور صوره تانيه بلبس تانى خالص قيمته و ليكن 10 طيب اصبح دلوقتى الراجل معاه صوره ستايل جديد ب 10 و صاجبه معاه صوره ستايل قديم ب 5 فكده تغيير الصوره الى مع الراجل الاصلى و قيمتها ماثر على الصوره او القيمه الى مع صاحبه و ده الى بيتسمى ال Value type و سميت بالاسم ده نظرا لانها تعتمد على القيمه الفعليه لل Variable   
مثال

void ValueType()

{

int num1 = 5;

int num2 = num1;

num1 = 10;

Console.WriteLine(num1); // 10

Console.WriteLine(num2); // 5

}

ثانيا ال Reference Type ده يا سيدى انت لما بتديله قيمه بياخدها بشكل عنوان و ليس القيمه وبالتالى الفاريبول الاصلى بيعمل للعنوان ده اسين على فاريبول جديد فهو كمان بياخد العنوان ده من الفاريبول الاول و لما بيحب اى حد منهم يعرف القيمه بيروح يجيبها عن طريق العنوان الى اتسجل فى ال Heap ده و بالتالى اى تغيير فى القيمه سواء عن طريق الفاريبول الاصلى او النسخه الى منه الاتنين بيشوفوا القيمه الجديده و ده لان الاتنين مخزنين عناوين و ليس قيم و العنوان متمش تغييره لا ده القيمه المقابله للعنوان هى الى اتغيرت و بما ان الاتنين بصيين على نفس العنوان فالاتنين هيشوفوا نفس القيمه الجديده و لذلك طريقه التخزين اشبه ب ان

واحد راح عمل جروب واتس و عمل جروب مع صاحبه و الاتنين ادمنز جه حط صوره للجروب الى هنعتبرها بقيمه 5 فهو و صاحبه دلوقتى شايفين الصوره الى قيمتها 5 على الجروب ده الى هو هنعتبره العنوان الى هخزن فيه الصوره بتاعت الجروب و الى هى هنعتبرها

جه بعد يومين صاحبه مش عاجبه الصوره الى الاولانى حطها فراح مغيرها لصوره جديده قيمتها 10 فدلوقتى الاتنين صحاب شايفين الصوره الى قيمتها 10 و ادى مثال يا جماعه

void RefrenceType()

{

int[] num1 = { 5 };

int[] num2 = num1;

num1[0] = 10;

Console.WriteLine(num1[0]); // 10

Console.WriteLine(num2[0]); // 10

}

<https://www.tutorialsteacher.com/csharp/csharp-value-type-and-reference-type>

ال String بنعتبرها RefrenceValue علشان هى عباره عن Array of Char بس مع ذلك هى عمتا immutable Value يعنى غير قابله للتغيير و ده لان اى تغيير فى الفاريبول بتعتها هى مش بتغير فى الفاريبول لأ هى بتشيل الفاريبول خالص و تعمل فاريبول جديد بنفس الاسم و بترمى القديم فى الميمورى و بالتالى ده فى البرامج الكبيره لو ممسحناش الميمور اول باول بيحصل memory Leck و الميمور بتتملى و بيبقى حالها ما يعلم بيه الا ربنا

MB

########################################################################

Dash Between Number Digits   
---------------------------------

تعرف ان انت ينفع فى الارقام لما تكون كبيره انك تحط كل 3 ارقام علامه \_ و السى شارب هتشوفها عادى تماما و دى معموله علشان لو هتكتب مليار مثلا متتلغبطش فى عدد الاصفار و ده مثال للمليار بطريقه الكتابه بالعاديه و امو \_ ههههههه

using System;

class Program

{

static void Main(string[] args)

{

int Num = 1000000000;

int NumWith\_ = 1\_000\_000\_000;

Console.WriteLine(Num); // 1000000000

Console.WriteLine(NumWith\_); // 1000000000

}

}

MB

########################################################################

Var Vs dynamic   
---------------------------------

Var احنا بنحطها و نعرف المتغير و بعد كده نحط اى قيمه للمتغير و هى بتفهم هو من نوع ايه و على اساسه بتتعامل على انها من النوع ده فلو ادناها انتيجر هتبقى انتيجر اديناها دابل هتبقى دابل ادينها استرينج هتبقى استرينج

و مينفعش انى اجى بعد ما ادتها قيمه من نوع معين و اروح مغيره يعنى مينفعش اديلها مثلا قيمه استرينج و اجى بعديها بسطر اديها انتيجر لا ساعتها هتعمل ايرور

اما ال Dynamic لا مختلف عن ال Var فى انه ينفع تديله فى السطر الاول انتيجر و فى السطر التانى استرينج و فى التالت بتنجان مقلى عادى خالص

و دى مثال للكلام ده

using System;

class Program

{

static void Main(string[] args)

{

var IntNum = 5;

Console.WriteLine(IntNum); // 5

IntNum = "5"; // syntax Error

}

}

هيعمل مشكله اول ما اكتب جوه الفاريبول استرينج بعد ما اديته اول قيمه انتيجر

بينما فى ال dynamic هيشتغل عادى خالص

using System;

class Program

{

static void Main(string[] args)

{

dynamic IntNum = 5;

Console.WriteLine(IntNum); // 5

IntNum = "5";

Console.WriteLine(IntNum); // 5

}

}

MB

########################################################################

Escape Sequences in string   
---------------------------------

<https://msdn.microsoft.com/en-us/library/h21280bw.aspx>

دى تماما ان اذاى انت بتكتب دابل كوت فى الاسترينج و اذاى بتكتب باكسلاش بص يا سيدى فى طريقيتن الطريقه الاولى انك بتعمل باك سلاش قبل الحاجه الى انت عاوز الكمبيلر يعتبرها من ضمن الاسترينج  
ذى   
string Name = "E:\\Courses\\Summries\\";  
دى الاوت بوت بتعها هيبقى   
E:\Courses\Summries\

طيب لو عاوز تعمله بالطريقه التانيه و دى الطريقه الاسهل فى السى شارب ؟؟ انك تحط قبل الدابل كوتس الى قبل كتابه الاسترينج   
 @  
و أدى كود يبينلك يا حج   
  
string Name = @"E:\Courses\Summries\";

MB

########################################################################

Common c# operators in arabic  
--------------------------------------

عند الاحتياج لاعطاء قيمه لمتغير ما

بيساوى و تستخدم فى انى اساوى حاجه بحاجه =

عند عمل عمليات حسابيه

موديلاس يعنى باقى قسمه كذا على كذا %  
 طرح -  
 جمع+  
 ضرب\*  
 قسمه /

عند المقارنات   
 بيقارن بين حاجتين و بيشوفهم لا يساوى بعض ولا لأ و استخدامها دائما مع ايف ستيت مينت !=

بيقارن بين حاجتين و بيشوفهم بساوى بعض ولا لأ و استخدامها دائما مع ايف ستيت مينت ==  
 بيقارن بين حاجتين و بيشوفهم بيساوى او الاول اكبر ولا لأ و استخدامها دائما مع ايف ستيت مينت >=

بيقارن بين حاجتين و بيشوفهم بيساوى او الاول اصغر ولا لأ و استخدامها دائما مع ايف ستيت مينت <=

بيقارن بين حاجتين و بيشوفهم الاول اكبر من التانى ولا لأ و استخدامها دائما مع ايف ستيت مينت >

بيقارن بين حاجتين و بيشوفهم الاول اصغر من التانى ولا لأ و استخدامها دائما مع ايف ستيت مينت <

وضع شروط فى الايف ستيت مينت

دى يعنى اند (و) و تستخدم فى الايف ستيت منت و هنا الشرطين يجب تحقيقهم لارجاع ترو &&

دى يعنى أور (أو) و تستخدم فى الايف ستيت منت و هنا تحقيق شرط واحد كافى لارجاع ترو ||

دى بختصر بيها الايف كونديشن فى سطر واحد ابقى ابحث عن الشورت ايف فى السى شارب?:

?? دى بنستخدمها فى الايف كونديشن ولكن بشرط نكون بنستخدم الى بيحول الى مش نولابول الى نولابول هتفهم بعدين

MB

########################################################################

nullable types  
--------------------------------------

Value Types : int , Float ,double,Struct , Enum , Etc

لو مديتش ليهم اى قيمه فاصلا الانشسال فاليو متسجله فيهم هيرجعوها بشكل تلاقى و ليكن مثلا هتلاقى ان الانتيجر الانشيال فاليو بتاعه صفر و كذلك الفلوت و الدابل و البتالى هى مش هتديك نووووووول ابدا   
  
طيب امال ايه الى بيشيل الانشيال فاليو ب   
Null   
هتلاقى يا سيدى ان الى بيشيل الانشيال فاليو ب نوووووووول هما ال   
Refrenced Types   
على سبيل المثال

int I = 0 ;

Refrenced Types : String , Class , Interface , Delegates , Arrays

على سبيل المثال

String I = null;

طيب من الكلام ده انا فهمت ان ال   
Value Types  
مش بتقبل Null طيب بس فى طريقه تخليها تقبل Null عارف هى ايه انك تحط بعد النوع علامه استفهام و بالشكل ده اصبحت بتقبل Null عادى جدا   
و ادى مثال

int? I = null;

بس انا ليه هحتاج اخليه يقبل النول بقه ؟؟؟  
عشان فى الداتا بيز الانتيجر و الكاراكتر و حاجات كتير بتقبل ال null و بالتالى هيكون افضل جدا لو احنا خلينا النوع يقبل ال null لان ده هيريحنا جامد جدا فى التعامل مع الداتا بيز

bool? Ready = null;

if (Ready == true)

{

Console.WriteLine("Yes Ready ");

}

else if (Ready == false)

{

Console.WriteLine("Not Ready ");

}

else

{

Console.WriteLine("Do Nothing there is no Answer From User Only Null");

}

-----------------------------------------------------------------------------------------------------------

طيب هل لما انا بحول الانتيجر بحيث يقبل ال null هل ساعتها بيبقى انتيجر ؟؟  
لا طبعا ساعتاها بيبقى انتيجر متحول و بالتالى عشان ارجعه لانتيجر تانى بعمله تحويل لانتيجر بالكاسيتنج   
الى هو اذاى ؟؟  
بص مثال اهه عشان تفهم عشان يرجع لاصله بعمل الكاستينج ليه اذاى و كمان هوريك هو هيرجع انتيجر ولا لأ ؟

int? num = 1;

Console.WriteLine("Yes it is {0} ", num); //No casting

num = 2;

Console.WriteLine("Yes it is {0} ", (int)num); //casting

num = 3;

Console.WriteLine("Yes it is {0} ", Convert.ToInt32(num)); //casting

لازم تبقى فاهم بقه ان Null دى من الاخر بتخلى النتيجه فاضيه يعنى Empty or Nothing

عمليه التحويل بال كونفيرت هى الاصح

MB

##############################################################

Using ?? and :?

--------------------------------------------------------------

using System;

class Program

{

static void Main(string[] args)

{

int? num = 100;

int Num1;

if (num == 100)

{

Num1 = (int)num;

}

else

{

Num1 = 0;

}

Console.WriteLine(" Result is {0}", Num1);

// :?

int? nums = 100;

int Nums1 = (int)nums == 100 ? (int)nums : 0;

Console.WriteLine(" Result is {0}", Nums1);

// ??

int? numsa = null;

int Numsa1 = numsa ?? 0;

Console.WriteLine(" Result is {0}", Numsa1);

int Numsa1\_ = num ?? 0;

Console.WriteLine(" Result is {0}", Numsa1\_);

}

}

التلاته طرق دول فى الكونديشن معناهم واحد و هو : لو الرقم ب 100 فخلى رقم1 هو كمان ب 100 و لو بنول خلي رقم1 يساوى صفر

MB

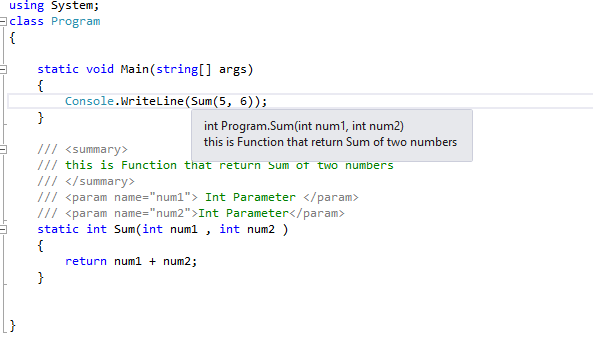
##############################################################

Comments

-------------------------------

لو الكومنت على سطر واحد بعمله ب //

لو كذا سطر بكتب الكذا سطر وسط دول بقه /\* سطور الكومنت \*/  
طيب لو عاوز تعمل دوكمنتيشن يظهرلك على فانكشن مثلا لما تيجى تمشى الماوس عليها دى بتتعمل اذاى   
بتدوس /// و بتكتب السامرى بقه بعد كده برحتك



طيب فى حاجه اسمها todo Comment ده يا معلم انت بتعمل كومنت عادى خالص بس بتكتب فى اوله todo فهو بيعتبرها تاسك ليك تقدر توصلها من ال task list فى ال Visual studio تعالى اوريهالك بتتعمل اذاى و بتتجاب منين فى صوره   
A screenshot of a computer

Description automatically generated

و علشان تفتح التاسك ليست بتجيبها من View الى فوق فى ال Visual Studio يا صديقى

A screenshot of a computer

Description automatically generated

MB

##############################################################

Coverting Type Methods

-------------------------------------------

اول حاجه لازم تعرفها ان اى حاجه فى الدوت عباره عن type و اهمهم ال Object لان اى شئ فى الدوت نت الروت بتاعه هو ال Object   
ذى ما البشر ابوهم أدم فاى كلاس او اى نوع او اى حاجه ابوها ال Object

طيب دلوقتى انا عندى كذا DataType وانا عاوز احول من نوع الى نوع هل ينفع ده ؟؟

ايوه ينفع لكن بشروط معينه مينفعش اجى اخزن string جوه int كده من غير ما اقوله لو ينفع تحولى string ده ل int حوله و خزنهولى و من هنا طلع طرق تحول الفاريبول من نوع الى نوع

هنا بوريك المشكله و بعد كده هوريك حلها و مشاكل حلولها لحد ما اوصلك لاحسن حل فى الاخر

المشكله هنا انى بحاول اخزن string جوه int فالكومبايلر صوت

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "0";

int Num = stringNum; // Syntax Error (Can not implicitly Convert From string to int)

Console.WriteLine(Num);

}

}

}

هنا هو بيقولك لا يمكن تخزين string فى int بشكل ضمنى او بطاعه عمياء يعنى لازم علشان يحولهالك تنفذله شرط

طب و ايه الشرط ده يا سيدى ؟؟

شرط انك تعمل تحويل بالكود الاول بنفسك من string الى int و بعد كده هو هياخد الى انت حولته ده يخزنه فى int بقه لانك خلاص حولته int

الطريقه الاولى : explicit Cast

الطريقه دى فى حالتنا بتاعت تحويل ال الاسترنيج ل int مش هينفع لانه بيتعامل مع تحويل الارقام من نوع long ل int بس بشرط يكون الرقم مش اكبر من الماكس بتاع الانتيجر

و مثال على ذلك اهه

namespace Test

{

class Program

{

static void Main(string[] args)

{

//string stringNum = "0";

//int Num = (int)stringNum;

double Num = 500d;

if(Num >= int.MinValue && Num <= int.MaxValue)

{

int intNum = (int)Num; // Explicit Convert

Console.WriteLine(intNum);

}

Console.ReadKey();

}

}

}

طب لو عوزت بقه احول من استرينج ل int المفروض استخدم ايه ؟

استخدام فانكشن

Convert.

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500";

int Num = Convert.ToInt32( stringNum);

Console.WriteLine(Num);

Console.ReadKey();

}

}

}

الطريقه دى هتشتغل و ذى الفل بس لو اتبعت اى char فى الاسترينج و مبقاش ارقام بس البرنامج هيضرب

و ده مثال للبرنامج الى هيضرب لما ابعتله حروف مش ارقام و بس

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500ff";

int Num = Convert.ToInt32( stringNum); // Run Time Error

Console.WriteLine(Num);

Console.ReadKey();

}

}

}

فى طريقه اخرى للتحويل عن طريق اال parsing و دى ميزيتها انها اسرع من ال convert

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500";

int Num = int.Parse( stringNum);

Console.WriteLine(Num);

Console.ReadKey();

}

}

}

هتقابلنى برضوا نفس مشكله ان لو جيت احول حاجه غلط هيضرب فى وشى

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500ffff";

int Num = int.Parse( stringNum); // Run Time Error

Console.WriteLine(Num);

Console.ReadKey();

}

}

}

فلما لقوا الدنيا خربانه كده واوقات كتير بيحصل مشاكل اثناء التحويل قالك احنا نعمل طريقه بقه تهندل الكلام ده

لو التحويل تم يبقى فل الفل لو متمش يرجعلى ديفولت فاليو

و دى عن طريق tryparse هنا بيقوله روح حاول تحول الاسترينج ده لانتيجر لو نفع حطلى القيمه فى ال out الى اسمه Num و فعلا فى المثال ده هينفع فهيجطه 500 دوه ال Num و طبعا الريتيرن بتاع الفانكشن ده Boolean يعنى true او false ال true يعنى التحويل تم بنجاح و ال false يعنى فشل و لما بيفشل بيدى ال Num ال default value of int الى هى صفر

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500";

int Num;

bool IsConverted = int.TryParse( stringNum , out Num);

if(IsConverted == true)

{

Console.WriteLine(Num); // 500

}

else

{

Console.WriteLine(Num); // 0

Console.WriteLine("Can not Convert this String to int");

}

Console.ReadKey();

}

}

}

دى بقه لما الفانكشن تفشل فى التحويل فتطبعله رساله انه منفعش بس مش هتضرب الكود

namespace Test

{

class Program

{

static void Main(string[] args)

{

string stringNum = "500dsdfg";

int Num;

bool IsConverted = int.TryParse( stringNum , out Num);

if(IsConverted == true)

{

Console.WriteLine(Num); // 500

}

else

{

Console.WriteLine(Num); // 0

Console.WriteLine("Can not Convert this String to int");

}

Console.ReadKey();

}

}

}

دى شويه امثله بقه لكل حاجه

using System;

class Program

{

static void Main(string[] args)

{

float Num1 = 12345.65F;

int Res1 = (int)Num1;

Console.WriteLine(Res1); //output will be = 12345

//#########################################################

int Res2 = Convert.ToInt32(Num1);

Console.WriteLine(Res2); //output will be = 12346

//#########################################################

string Num2 = "1235";

int Res3 = int.Parse(Num2);

Console.WriteLine(Res3); //output will be = 1235

//#########################################################

string Num4 = "1235Fsd";

int Res = 0;

bool ISConverted\_Success = int.TryParse(Num4, out Res);

if (ISConverted\_Success == true)

{

Console.WriteLine("Converting done : " + Res);

}

else

{

Console.WriteLine("No thing Converting ");// this condition will printed

}

//#########################################################

string Num3 = "1235Fsd";

int Res4 = int.Parse(Num3);

Console.WriteLine(Res3); //Error because it can not convert Char to int

}

}

الخلاصه من الكود ده الكونفيرت افضل من حيث الامان من انى اعمل كاست بالاقواس   
البارس بقه بيحول الاسترينج الى مكتوب على شكل ارقام لانواع رقميه عادى   
التراى برس ده بقه ذى ما يكون بيعمل تيست لو ينفع الاسترينج ده يتحول لنوع رقمى يبقى ترو لو مينفعش يرجع فولس لانه بيرجع نتيجه من نوع bool و فى حاله انه نفع بيرجع القيمه فى ال Res الى هى من نوع out

فى بقه شويه حاجات بينفع يتحولوا من غير دخل من اى حد ذى انك تخزن حاجه int جوه حاجه double و ده لان حجم الانتيجر يسمحله يتخزن جوه الدوبل و كمان الاتنينج ارقام و الانتيجر يعتبر جزء صغنن من ال Double انما العكس مش بينفع بمعنى مينفعش اخزن ال double جوه ال int

MB

##############################################################

Array

-------------------------------------------

ذى ما انت شايف الاراى طريقه كتباتها اهه بقوله عندى اراى من نوع انتيجر اسمه arr اعملى منه بقه نسخه جديده و حط فيها 3 قيم بس الى هى حجم الاراى بمعنى الاراى ده ذى ما تقول بيقولك   
عاوز اعمل بيت البيت ده هنقول نوعه عماره طيب العماره الجديده دى بقه انا عاوزها 3 شقق بس كده   
------------------------------------------  
بتحط فى كل شقه قيمه القيمه دى تناسب نوع البيت يعنى ما جبش ناس عايشين طول عمرهم فى قصر و اقوله تعالى اعد فى العماره بتكلم عن الانتيجر الى هو النوع هههههه و بالتالى كل شقه لازم قيمه مناسبه ليها و لما اجى عاوز احط او عاوز اعرف القيمه دى بقه ايه مش المفروض انى بحددله العنوان بتاع الشقه الى هيحط فيه القيمه الى هيسكنه بمعنى اصح ولا هتبقى سداح مداح لا اكيد   
طيب العنوان ده بقه الى هو رقم الشقه الى هو الانديكس طيب هل ينفع يبقى عندى 3 شقق و اروح جايب ساكن رابع ؟؟  
أكيد لا الساكن ساعتها مش هيلاقى شقه و هيعمل مشكله معايا   
طيب يبنفع اسكن اتنين بس و فى 3 شقق ؟؟  
ايوه و هتبقى قيمه الى الشقه الفاضيه دى على حسب نوع الناس البيت يعنى هنا انتيجر فهتبقى قيمه الى جوه الشقه فاضى (صفر)  
طب لو الاراى بقه من نوع سترينج هيرجعلى Null او بمعنى تانى مش هيرجع الا هواااااااء طليق ههههههههه فاضى   
اما لو اديته انديكس فيه حاجه اكيد هيرجعها

static void Main(string[] args)

{

//type[] name = new type[size]

int[] arr = new int[3];

//arrayName[Index] = Integeral Value

arr[0] = 5;

arr[1] = 7;

arr[2] = 6;

Console.WriteLine(arr[1]);

}

static void Main(string[] args)

{

//type[] name = new type[size]

int[] arr = new int[3];

//arrayName[Index] = Integeral Value

arr[0] = 5;

arr[1] = 7;

Console.WriteLine(arr[2]);

}

static void Main(string[] args)

{

//type[] name = new type[size]

String[] arr = new String[3];

//arrayName[Index] = String Value

arr[0] = "Mahmoud";

arr[1] = "Mostafa";

Console.WriteLine(arr[2]);

}

**طريقه كتابه اسهل فى الاراى**

static void Main(string[] args)

{

//type[] name = {Values here}

int[] arr = {3,6,5};

Console.WriteLine(arr[2]);

}

مميزات الاراى :-   
1- بتوفر ليك انك تحط مجموعه فاريبول من نوع واحد فى مكان واحد و تستدعيه بالانديكس بتاعه

عيوب الاراى :-   
1- لها حجم معين و ده كده اوحش عيب فيها مش يعنى غير محدوده الحجم فاحيانا بتعمل مشكله لو اديتها حاجه زياده عن حجمها   
2- الاستدعاء الانديكس برضوا عشان ممكن تتلغبط فى انديكس فالدنيا تخرب

MB

##############################################################

If Statements

-------------------------------------------

بص يا سيدى الايف ستيت مينت دى ما هى الا انى بدخل قيمه ما و اقارنها بالى متخزنه فى الميمور سواء ببيساوى == او >= او <= أو > أو < أو =!   
و لو شرط المقارنه ده اتحقق بينفذ السطور الى فى القوس الخاص بالايف ستيتمينت دى فمثلا فى المثال الى جنبنا ده لو انا دايته رقم 20 مثلا هيروح يقارن اول واحد هيقوله هل 10 بتساوى 20 طبعا لا هيسيبها و يخش على ايف الى بعده و يقوله هل 20 بيساوى 20 هيقوله اه فهينفذ و يطبع ان الرقم بيساوى 20 و عشان البرنامج ده غبى حبه فهيروح يقارن و يقوله هل ال 30 بتساوى ال 20 فهيقوله لأ طبعا ده كده برفورمنس معفن فى البرنامج و المفروض انه اول ما بيلاقى الشرط الى هو عاوزه بيبصش لحاجه تانيه طب نحل المشكله دى اذاى ؟؟  
 خنستخدم بقه يا سيدى Else if و Else

static void Main(string[] args)

{

Console.WriteLine("Please Enter Number : ");

int Num = int.Parse(Console.ReadLine());

if (Num == 10)

{

Console.WriteLine("The Number is : {0} ", Num);

}

if (Num == 20)

{

Console.WriteLine("The Number is : {0} ", Num);

}

if (Num == 30)

{

Console.WriteLine("The Number is : {0} ", Num);

}

هنا بقه لما استخدمنا Else if و Else ده بقه خلى البرنامج لو انت مثلا اديته 10 هيروح رايح فى اول ستيتمينت و يقوله هل 10 بتساوى 10 هيقوله اه هيروح منفزها وشكرا مش هيفحص الباقيين بقه و مش هيعطل نفسه   
  
موضوع البرفورمنس ده بيبان مع الداتا الكبيره مش التجارب العبيطه بتعتنا دى   
static void Main(string[] args)

{

Console.WriteLine("Please Enter Number : ");

int Num = int.Parse(Console.ReadLine());

if (Num == 10)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else if (Num == 20)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else if (Num == 30)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else

{

Console.WriteLine("The Number must be (10,20,30)";

}

}

MB

##############################################################

If Statements (&& , & , || , | )

-------------------------------------------

لما تكون عاوز تعمل شارطين او تلاته فى الايف ستيت منت الواحده بتستخدم   
OR || و دى معناها ان اى شرط هيتحقق هينفذ الى جوه الايف ستيتمينت على طول   
طيب   
And && دى معناها ان لازم كل الشروط تتنفذ عشان الايف ستيت مينت تتنفذ و اى شرط منفعش بيبطلعا من الايف ستيت مينت دى   
=====================  
طيب عاوز اعرف الفروقات بين لما اعمل && و & و كذلك || و |  
  
بص يا سيدى   
|| دى يعنى هيتشك و أول ما يشوف اى شرط اتحقق ميبصش على الباقى خالص و ينفذ الامر   
  
| دى بقه يعنى لازم يعمل فحص لكله حتى لو لقى شرط هيحقق الايف ستيت مينت و يخليها تتنفذ فبيأجل التفيذ لحسن الفحص لى كل الشروط و طبعا ده بيعطل البرنامج   
  
&& دى بقه بيفحص اول ما يشوف شرط متحققش بيطلع من الايف ستيت مينت دون النظر على الباقى بيخدها من قاصرها يعنى عارف انه كده كده مش هينفذها   
  
& دى بقه لا بيفحص كله مع انه ممكن يبقى عرف انه هيطلع عشان شرط فى الاول متحققش بس لا لازم عشان يطلع يفحص كله الاول غبى يعنى

static void Main(string[] args)

{

Console.WriteLine("Please Enter Number : ");

int Num = int.Parse(Console.ReadLine());

if (Num == 10 || Num == 20 && Num != 30)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else if (Num >= 22 || Num == 60)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else if (Num == 21 && Num > 20)

{

Console.WriteLine("The Number is : {0} ", Num);

}

else

{

Console.WriteLine("The Number is Invalid");

}

}

MB

##############################################################

Switch & break & Goto

--------------------------------------

نفس شغل الايف ستيت مينت بس هنا بقه بشكل تانى مش اكتر هنا بقوله شوفلى القيمه دى لو فى حاله انها بكذا نفذلى كذا و بعد كده اعمل بريك و اطلع   
 خلى بالك لو نسيت البريك هيعتبرالكيس الى بعده معاه هههههههههههه فمهم متنساش البريك خالص   
و فى حاجاه تانيه   
فى حاجه اسمها Goto Label   
دى حاجه عفا عليها الزمن بس اديك تعرفها و خلاص مهمتا انى ممكن احط ليبل فى مكان ما فى الكود و اول ما الكود يشوف Goto   
لليبول ده يروح جرى رايح واقف على الليبول ده و مكمل الكود تانى   
ايام زمان كانوا بيستخدموها فى عمل فور لوب بس من التمنينات محدش بيستخدمها يا جدعان   
تعالوا بقه اوريكوا كود جميل يوضح عمل السويتش

كود يوضح السويتش و الجوتو و البريك

static void Main(string[] args)

{

LabelNum1 :

Console.WriteLine("Please Enter Number : ");

int Num = int.Parse(Console.ReadLine());

switch (Num)

{

case 5:

Console.WriteLine("Please Enter Number : {0}", Num);

break;

case 10:

Console.WriteLine("Please Enter Number : {0}", Num);

break;

case 15:

Console.WriteLine("Please Enter Number : {0}", Num);

break;

default :

Console.WriteLine("InValid Num");

goto LabelNum1;

}

}

شرح الكود بقه   
الكود هنا بقوله اكتب رقم لو الرقم ده 5 خش فى الكيس 5 و اكتبلى الرقم 5 و اوقف الفحص بالبريك و لو 10 خش فى الكيس بتاعت 10 و لو 15 خش فى الكيس بتاعه ال 15 غير كده يبقى خش فى الديفولت و اكتبله ان الرقم مش مش صح و بعد كده خليه بالجو تو يروح لمكان الليبول الى اسمه LabelNum1 فهيروح رايح مشغل الكود من فوق خالص تانى جرب بقه الكود و هتفهم اكتر

كود كمان يوريك انك جوه كل سويتش تقدر تكتب كود كتير مش مجرد كونسول يعنى

static void Main(string[] args)

{

LabelNum1:

Console.WriteLine("Please Enter Number : ");

string Message;

int Num = int.Parse(Console.ReadLine());

switch (Num)

{

case 5:

Message = "From Case 5 the Entered Number : {0}";

Num = Num \* 5;

Console.WriteLine(Message, Num);

break;

case 10:

Message = "From Case 10 the Entered Number : {0}";

Num = Num \* 10;

Console.WriteLine(Message, Num);

break;

case 15:

Message = "From Case 15 the Entered Number : {0}";

Num = Num \* 15;

Console.WriteLine(Message, Num);

break;

default:

Console.WriteLine("InValid Num");

goto LabelNum1;

}

}

MB

##############################################################

Switch Expresssion

-------------------------------------------

دى بقه يا سيدى نوع جديد من السويتش جميل جدا فى طريقه كتابته و مهم جدا و ده لينك بيشرحه تفصيليا

<https://zetcode.com/csharp/switch-expression/?fbclid=IwAR2qI2wUgSQiYfuRcMcicv8Zv4fDow69asoYZblQrgWhKhg6o-5bVmklz0I>

و طريقه كتابته كالتالى

using System;

class Program

{

static void Main(string[] args)

{

Console.Write("Enter a domain name: ");

string? domain = Console.ReadLine();

domain = domain?.Trim().ToLower();

string result = domain switch

{

"us" => "United States",

"de" => "Germany",

"sk" => "Slovakia",

"hu" => "Hungary",

\_ => "Unknown"

};

Console.WriteLine(result);

}

}

هنا ذى ما انت شايف بقوله انا هعملك سويتش بس السنتكس بتعها شكلبه جميل

بقوله عندك حاجه اسمها domain لو الدومين دى انت حاطط فيها اى اختصار لاى دوله ذى us فهاتلى الى السهم بتاع ال arrow function مشاور عليه و شغله و لو دخلتله حاجه غريبه مش موجوده فيه طلع رساله unknown

يعنى هو بياخد الارقام الى هيعمل بيها سويتش او الانبوت بتاع الوسيتش يعنى من ال domain علشان كده بيقول domain switch و لما بيطلع النتيجه بيحطها فى الاسترينج الى اسمها result

المثال التانى ده بديله اراى كله ارقام و اقوله امشى جواها و اعملى سويتش على الى الارقام و الى اكبر من صفر يبقى موجب و الاصغر يبقى سالب و الى يساوى الصفر يبقى كده صفر

واخد بالك بقه من انه و هو بيعمل اكبر من صغر و اصغر من صفر هو مش مضطر يكتبها لان هو فهم اصلا ان ده سويتش على ال num و اشتغل عليه على اساسه من دون ما حد يفهمه و يكتب كود

using System;

class Program

{

static void Main(string[] args)

{

var nums = new List<int> { -3, 2, 0, 1, 9, -2, 7 };

foreach (var num in nums)

{

var res = num switch

{

< 0 => "negative",

0 => "zero",

> 0 => "positive"

};

Console.WriteLine($"{num} is {res}");

}

}

}

هنا بقه السويتش على الكلاسيس يعنى ابعتله كلاس فى السويتش فهو يخش يشوف لو فى حاجه من جوه الكلاس بتوافق شرط ما من شروطه بتكمل شغلها و لو مفيش حاجه متوافقه معاها فنديها رساله عدم التوافق ذى المثال ده

using System;

class Program

{

static string checkPerson(User u)

{

return u switch

{

{ LName: "Doe" } => "Doe family",

{ Age: var age } when age < 18 => "minor person",

\_ => $"{u} is an Object that does not match conditions"

};

}

static void Main(string[] args)

{

var users = new List<User>

{

new ("John", "Doe", 34),

new ("Roger", "Roe", 44),

new ("Jane", "Doe", 44),

new ("Robert", "Molnar", 41),

new ("Lucia", "Petrovicova", 16),

};

foreach (var user in users)

{

Console.WriteLine(checkPerson(user));

}

}

}

internal class User

{

public User(string fName, string lName, double age)

{

FName = fName;

LName = lName;

Age = age;

}

public string FName { get; set; }

public string LName { get; set; }

public double Age { get; set; }

}

هنا عندنا ليسته فيها مستخدمين كل واحد ليه اسم و اسم اب و سن

عندنا بقه عاوزين نعمل سويتش على الكلاس بمعنى سويتش على المستخدم ده لو المستخدم ده مثلا اسمه محمود طلعلى رساله ازيك يا محمود لو المستخدم ده اسمه محمد و سنه اكبر من 20 سنه رجعلنا فى النتايج و هكذا يعنى هنا بعتله لسته مستخدمين و بقوله جوه السويتش لو فى اسم اخبر لاى مستخدم اسمه doe اطبعهولى انه doe Family الاخنيار التانى ان لو السن اصغؤ من 18 ابعتله رساله miner person

طبعا انت واخد بالك ان لما جيت اشغل الانتيجر فى السويتش الى عامل المفروض بياخد user object رحت مبعتش الانتيجر بشكل مباشر لأ انا قولتله انا عندى اوبجيكت فيه حاجه اسمها Age انا عاوزها لو اقل من 18 تطلع رساله Miner person

و طبعا هى مكتوبه كده

{ Age: var age } when age < 18 => "minor person",

الى هو بقوله انا الاوبجيكت بتاعى هيجيلى فيه عمر العمر ده خزنه فى فاريبول و لما يكون الفاريبول ده ب اقل من 18 طلعلى الرساله بس ليه كل ده منا اقدر اكتبه كده

{ Age: < 18 } => "minor person",

و يبقى الكود بتاع السويتش كله على بعضه شكله كده جميل اهه

static string checkPerson(User u)

{

return u switch

{

{ LName: "Doe" } => "Doe family",

{ Age: < 18 } => "minor person",

\_ => $"{u} is an Object that does not match conditions"

};

}

بقى حتى اسهل فى الفهم

هنا بقه فى المثال ده ذى ما انت شايف السويتش بياخد حاجتين و بيرجع رساله و البرنامج ده عباره عن لعبه بدخله انا اختيارى و الكمبيوتر بيختار اختيار راندوم و لو الاختيارين متشابهين لسويتش بنفس الترتيب بيروح مطلع الرساله الى قدام السويتش

ذى مثلا انا اختر paper و الكمبيوتر اختار rock فلما يخشوا فى السويتش مع بعض هيبقى الناتج الرساله الى هى Rock is covered by paper. You loose

using System;

class Program

{

static void Main(string[] args)

{

while (true)

{

var menu = "Select: \n1 -> rock\n2 -> paper\n3 -> scissors\n4 -> finish";

Console.WriteLine(menu);

string[] options = { "rock", "paper", "scissors" };

int val;

try

{

var line = Console.ReadLine();

if (string.IsNullOrEmpty(line))

{

Console.WriteLine("Invalid choice");

continue;

}

val = int.Parse(line);

}

catch (FormatException)

{

Console.WriteLine("Invalid choice");

continue;

}

if (val == 4)

{

break;

}

if (val < 1 || val > 4)

{

Console.WriteLine("Invalid choice");

continue;

}

string human = options[val - 1];

var rnd = new Random();

int n = rnd.Next(0, 3);

string computer = options[n];

Console.WriteLine($"I have {computer}, you have {human}");

var res = RockPaperScissors(human, computer);

Console.WriteLine(res);

}

Console.WriteLine("game finished");

string RockPaperScissors(string human, string computer) => (human, computer) switch

{

("rock", "paper") => "Rock is covered by paper. You loose",

("rock", "scissors") => "Rock breaks scissors. You win.",

("paper", "rock") => "Paper covers rock. You win.",

("paper", "scissors") => "Paper is cut by scissors. You loose.",

("scissors", "rock") => "Scissors are broken by rock. You loose.",

("scissors", "paper") => "Scissors cut paper. You win.",

(\_, \_) => "tie"

};

}

}

و فى نوع تانى بقه بياخد الاوبجيكت و يسويتش على حسب نوع الاوبجيكت او على حسب الداتا تايب

ذى ده مثلا

using System;

class Program

{

static void Main(string[] args)

{

int age = 23;

string name = "Peter";

List<string> colors = new List<string> { "blue", "khaki", "orange" };

var nums = new int[] { 1, 2, 3, 4, 5 };

Console.WriteLine(check(age));

Console.WriteLine(check(name));

Console.WriteLine(check(colors));

Console.WriteLine(check(nums));

object check(object val) => val switch

{

int => "integer",

string => "string",

List<string> => "list of strings",

Array => "array",

\_ => "unknown"

};

}

}

MB

##############################################################

While Loop

-------------------------------------------

دى بتخليك تلف كذا مره و تنفذ الاوامر دى اد العدد المطلوب فى الشروط بتعتها   
بص هنفذ برنامج و اشرحهولك احسن

static void Main(string[] args)

{

int Num = 1 ;

while (Num<=15)

{

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", Num));

Num++;

}

}

الكود ده معناه بص ياعم عرفى فاريبول انتيجر و قول انه بيساوى واحد طيب تعالى بقه فى ال While بقوله طول ما الرقم ده اقل من او يساوى 15 طلعلى الرساله الى انا حاططها دى زود الرقم 1   
فهينفذ الرساله دى من 1 لحد 15 و اول ما يوصل 15 هيقف بس كده

MB

##############################################################

Do While Loop

-------------------------------------------

نفس الى فوق بس الفرق ان فى ال do while انا انا بحط كود do الاول فبينفذه قبل ما يخش فى ال While و بعد كده بيبتدى يشتغل على do فالفرق ان فى ال do اول مره بيحصل ده بيبقى اجبار و بعد كده بيخش فى الوايل لوب   
انما فى الوايل بس بيخش فى الوايل من اول لفه   
من الاخر الفرق ان فى ال do while بحط الى هيحصل الاول فبيحص و بعد كده بيعمل check على حالته بينما فى ال while لا بيعمل ال check فى الاول و لو نفع بينفذ منفعش بينفذش   
 مثال توضيحى هتفهم منه بانى اعملك الرقم 16 و الاتشيك فى الوايل على اصغر من او يساوى 15

static void Main(string[] args)

{

int Num = 16 ;

do

{

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", Num));

Num++;

} while (Num <= 15);

}

ده عشان ما هو do while فالرقم 16 تمام هيخش ينفذ و يطلع الرساله ازيك يا محمود 16 بعد كده هيزود الرقم ل 17 بعد كده هيشوف الرقم ده اصغر من او يساوى ال 15 هيطلع لا فمش هيلف و يرجع ينفذ تانى بس كده   
لو ده بقه فى الوايل بس مكنش طلع اى نتيجه من اساسه

ايوه عشان تبقى فاهم   
Num++ دى يعنى زود على الرقم ده 1 طب لو عاوز ازود اتنين Num = Num+2 او Num +=2   
طيب لو عاوز اضربه فى 3 مثلا Num = Num\*3 او Num \*=3  
  
خلى بالك بقه فى حته معينه انك ممكن تعمل برنامج ومتخدش بالك انك عامل الشرط هيبقى على طول true ده بقه لو مفيش اى نوع من انواع البريك فطبعا هيفضل شغال للابد و ده انفينتى لوب طبعا مينفعش   
و ده مثال عليه و مثال على حله بطريقه ال Break الى بتخرجك من اللوب فى حاله ما

static void Main(string[] args)

{

//infinty loop

while (true)

{

Console.WriteLine("Hello Mahmoud badawy");

}

}

خد بالك ال break اول ما بتتنفذ بتخرج الكومبلير من اللوب و بيرجعش تانى يعنى خروج دون رجعه   
بينما ال Continue كانك بيعمل skip للفه واحد و يرجع يلف تانى فى اللوب عادى لو شرط لفته لسه متوفر و ليكن مثلا كل اما يشوف رقم زوجى يعمل كونتينو و يسيبه ميظهروش بينما لو جاب رقم 21 مثلا يعمل بريك

static void Main(string[] args)

{

//infinty loop with break

int num = 1;

while (true)

{

if (num == 21) { break; }

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", num));

num++;

}

}

MB

##############################################################

For Loop one Counter

-------------------------------------------

دى يا باشا ذى الوايل لو بس تغيير شكل بس ههههههه بيجمع الشرطه و الكاونتار و الزياده فى الكونتر فى مكان واحد و خلاص

static void Main(string[] args)

{

for (int Num = 1 ;Num <= 15; Num++ )

{

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", Num));

}

}

يعنى هنا انا عندى اللوب ده بيعمل ايه ؟ هنا الرقى الاولى للفاريبول Num = 1 فالكود كل الى هو عاوزه ان طول ما الرقم اقل من او يساوى 15 يطبع جمله hello Mahmoud و جمبها الرقم لما زاد بقيمه +1

فالى هيحصل هنا ان الرقم 1 اقل من 15 يبقى يطبع hello Mahmoud 1 و بعد كده يزود على الرقم 1 بقى 2 فيشوف الرقم ده اقل من 15 يروح طابع hello Mahmoud 2 و هكذا لحد ما يوصل انه يبقى بيساوى 15 ساعاتها هيطلع من اللوب و البرنامج يخلص على كده

MB

##############################################################

For Loop Counter With Prev and current Counters

-----------------------------------------------------------------------------

دى ذيها ذى الى فاتت بالظبط بس الفرق ان هنا بدل ما فى كاونتر واحد بس لأ فى اتنين كاونتر جوه اللوب و خير مثال عليه ال fobonatcci numbers الى هى اللوب بيشوف الرقم الحالى يجمع عليه الرقم السابق و يبقى دى نتيجه الرقم الى جاى بعديهم يعنى يبقى الرقم السابق 1 و التانى 2 يبقى الرقم الى بعديهم 3

مثال حقيقى بقه يعنى لو قولتله هاتلى الفوبوناتشى نامبرز لما يكون الانشيال نامبر للكونتر الرئيسى بصفر

Counter = 0 و طول ما هو اقل من 10 يفضل شغال

و الانشيال نامبرز للبريفياس بيساوى صفر previus =0 و الانشايل نامبر للحالى بيساوى 1 يعنى Current=1 مثالا هتبقى كده [0,1,1,2,3,5,8,13,21,34] و كود الحل هيبقى كده

الكود العادى خالص اننا نعمله كده

namespace Test

{

class Program

{

static void Main(string[] args)

{

int prev = 0, current = 1;

for (int Counter = 0; Counter < 10; Counter++)

{

Console.WriteLine(prev);

int NewFob = prev + current;

prev = current;

current = NewFob;

}

Console.ReadKey();

}

}

}

اما الطريقه الجديده انى احط الفاريبول prev , Current بجانب الكاونتر ذى ما انتوا شايفين

namespace Test

{

class Program

{

static void Main(string[] args)

{

for (int Counter = 0, prev = 0, current = 1; Counter < 10; Counter++)

{

Console.WriteLine(prev);

int NewFob = prev + current;

prev = current;

current = NewFob;

}

Console.ReadKey();

}

}

}

MB

##############################################################

For Each Loop

-------------------------------------------

يتم استخدامها فقط مع الاراى او الليست او الدكشنارى او الاراى ليست لان هى يتشتغل على الحاجات الى ذى الااراى لانها بالعربى كده بتلف على العناصر الى جواها بس فهى معموله مخصوص للحاجات الى ذى الاراى بدل ما اتعب دماغى و استخدم معاها الفورلوب العادى اعد احسب العناصر الى جوه الاراى و الحوارات البيض دى   
الفور ايتش من الاخر بتتعامل مع الكوليكشن او مجموعه من العناصر من نفس النوع بس   
مثال بالفور و بالفور ايتش بقه

static void Main(string[] args)

{

int[] MyArr = new int[3];

MyArr[0] = 10;

MyArr[1] = 20;

MyArr[2] = 30;

for (int Num = 0; Num < MyArr.Length; Num++)

{

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", MyArr[Num]));

}

}

static void Main(string[] args)

{

int[] MyArr = new int[3];

MyArr[0] = 10;

MyArr[1] = 20;

MyArr[2] = 30;

foreach (int item in MyArr)

{

Console.WriteLine(string.Format("Hello Mahmoud badawy {0}", item));

}

}

MB

##############################################################

For Each Loop and for loop

-------------------------------------------

هنا بوريك اذاى بتعمل انشياليز للارارى و اذاى تقراها بالفور و بالفور ايتش

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

int[] num = { 6,7,4,6,7,24,1};

int[] secNum = new int[5];

secNum[0] = 6;

secNum[1] = 8;

secNum[2] = 98;

secNum[3] = 10;

secNum[4] = 21;

for(int i=0; i<num.Length;i++)

{

Console.WriteLine(num[i]);

}

Console.WriteLine("####################################");

foreach (int item in secNum)

{

Console.WriteLine(item);

}

}

}

}

MB

##############################################################

Methods in C#

-------------------------------

بص ياعم دى الفانكشنز و دى مهمتها انك بتكتب الفانكشن مره واحده و تناديها في الكود براحتك 200000 مره المهم انك بتكتبش الكود تانى و بالتالى هي بتوفر عليك الوقت و طرقه الكتابه اهه و اذاى تناديها كمان

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

Program Obj = new Program();

Obj.MyInatancefunc(); // First Call

Obj.MyInatancefunc(); // second Call

Obj.MyInatancefunc(); // third Call

}

public void MyInatancefunc()

{

Console.WriteLine("hello Mr Mahmoud");

}

}

}

ذي ما انت شايف عملت الفانشكن هنا اهه الأول بكتب الاكسيس موديفاير و بعد كده نوع الريتيرن و بعد كده اسم الفانكشن و قوس البراميترز و بعد كده بكتب الفانكشن عادى خالص بقه و بالنسبه للكول انا بروح لل main فانكشن و بعمل اوبجيكت من الكلاس و انادى للفانكشن بقه بالاوبجيكت لان الفانكشن دى من نوع instance function

اما لو استاتيك فانكشن تتعامل اذاى بقه ؟؟

كل الاختلاف ان في الاستاتيك فانكشن بحط قبل نوع الريتيرن كلمه static و لما باجى اعمل كول للفانكشن بدل ما بعمل اوبجيكت من الفانكشن بروح مريح دماغى و منادى على الفانكشن باسم الكلاس بشكل مباشر و ده لانها ستاتيك يعنى مش هتتغير كل اما الاوبجيكت يتجدد او بمعنى اصح هى فانكشن واحده لكل الاوبجيكتس مش بتتغير فى اى اوبجيكت جديد فمنوش لازمه بقه انى اناديها بالاوبجيكت لا انا هناديها بالكلاس نفسه لانه حاجه فريده جلوبال للكل لا تتغير بتغير الانستانس فاهم قصدى و عشان كده هما عملوها تتنادى بااسم الكلاس على طول

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

Program.Myfunc(); // First Call

Program.Myfunc(); // second Call

Program.Myfunc(); // third Call

}

public static void Myfunc()

{

Console.WriteLine("hello Mr Mahmoud");

}

}

}

لو عاوز احط براميترز بقه في الفانكشن ؟؟ اهه ياعم الفانكشن فيها براميترز و كل براميتر معموله النوع بتاعه و في الكونسول مستخدم الاسترينج فورمات كمان

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

Program.Myfunc("Mahmoud",5); // First Call

}

public static void Myfunc(string name , int num)

{

Console.WriteLine("hello Mr: {0} ===> you have {1} kids" ,name ,num);

}

}

}

MB

################################################################

الفرق بين الفاليو باراميتر و الريفرنس

Call by ref vs call by value

--------------------------------------------

الكول باى فاليو هتلاحظ ان اى تغيير في ال num لا يأثر على ال MyNum و ده لانهم أصلا بيشاوروا على مكان مختلف تماما في الميمورى

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

int MyNum = 10;

Myfunc(MyNum);

Console.WriteLine(MyNum); // 10

}

public static void Myfunc(int num)

{

num = 15;

Console.WriteLine(num); // 15

}

}

}

اما الكول باى ريفرنس بقه فهتلاقى ان اى تغيير في ال num بيغير ال MyNum و العكس صحيح برضوا يعنى الى بيتغير بيغير التانى و الكود عشان يفهم ان الكول باى ريفرنس بنكتبه كده

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

int MyNum = 10;

Myfunc(ref MyNum);

Console.WriteLine(MyNum); // 15

}

public static void Myfunc(ref int num)

{

num = 15;

Console.WriteLine(num); // 15

}

}

}

هيعمل هنا اوت بوت 15 ثم 15 و ده لان هيعمل كول و يبعت براميتر ب 10 و اول ما يخش الفانكشن النمبر الى كان بعشره ده بقى بيشاور على 15 فهيطلع في كونسول الفانكشن 15 و بما ان ال MyNum بيشاور على نفس المكان فكل الى هيحصل بقه انه هيتغير هو كمان ل 15 ليصبح الاتنين بيشاوروا على نفس المكان و الاتنين قيمتهم ب 15

و ذي ما انت شايف الفرق بين العادى و الى بيشتغل بالريفرنس انى بحط كلمه ref في تعريف البراميتر جوه الفانكشن و في الكول برضوا

MB

################################################################

Out parameters

----------------------------------------

انت طبعا عارف ان الريتيرن بتخلى الفانكشن ترجع قيمه واحده تمام كده ؟ ايوه تمام

طيب لما أكون عاوز ارجع من الفانكشن كذا قيمه تتعمل اذاى ؟؟

بص يا سيدى بستخدم حاجه اسمها out paramerters و دى الى برجع فيها براحتى و بتتعمل اذاى بقه ؟؟

كده يا صديقى

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

int Sum = 0;

int Multi = 0;

Myfunc(5,6,out Sum , out Multi );

Console.WriteLine("Sum = {0} and Multi = {1}", Sum, Multi);

}

public static void Myfunc(int num1 , int num2 , out int Sum , out int Multi)

{

Sum = num1 + num2;

Multi = num1 \* num2;

}

}

}

ذي ما انت شايف بعرف الفانكشن ان ده براميتر نوعه انتيجر او اى حاجه بس بقوله قبل النوع ان ده مش انبوت لا ده اوتبوت هحط فيه النتايج و في الكول بعرفه برضوا ان معايا حاجه هيتحط فيها اوتبوت و طبعا بعملهم ديكليريشن و اعرف الكول انهم انتيجر و اوتبوت بس الدكليريشن ده الى بيحدد نوعهم و جوه الكول بقه بقوله انهم اوتبوت

MB

################################################################

Params parameters

----------------------------------------

دى كانك عملت البراميتر الى داخل ده اوبشنال بالظبط بس هي شرطها تبقى اراى و تبقى اخر براميتر في الفانكشن و تبقى هي الوحيده من نوع params لان دى بتشيل عدد مهول من البراميتر من نوع معين و بالتالى هي يعتبر عباره عن اراى من المدخلات و على سبيل المثال هوريك كول كله براميترز بالعبيط و هيتقرى عادى

class Program

{

static void Main(string[] args)

{

int[] num = { 6, 7, 4, 6, 7, 24, 1 };

Program.Myfunc();

Program.Myfunc(num);

Program.Myfunc(9,6,7,6);

}

public static void Myfunc(params int[] MyNum)

{

Console.WriteLine("the length of Array is : " + MyNum.Length);

foreach(int item in MyNum)

{

Console.WriteLine(item);

}

}

}

MB

################################################################

delegates video : 36

----------------

بص يا سيدى ده معمول كده عشان يخلى الكود اكثر استخداما   
و عشان تفهم اكتر هعملك مثال فيه فانكشن بناديه بطريقه عاديه و كذلك مثال بنادى فيه الفانشكن بالديليجيت و بعد كده اوريك اذاى هنستخدم الديليجيت بعد كده

أولا المثال العادى بنادى فانكشن بكل سهوله و يسر

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

MyFunc("Mahmoud Badawy");

}

public static void MyFunc(string Name)

{

Console.WriteLine("My name is : " + Name);

}

}

}

تعالى بقه نعمل الكود الى فوق ده بالديليجيت : هتلاحظ ان أولا الديليجيت و انت بتعمله لازم يبقى   
1- الاكسيس موديفاير بتاعه بنفس نوع الاكسيس موديفاير بتاع الفانكشن الى هيشاور عليها   
2- كذلك نفس الريتيرن تايب و كذلك نفس البراميترز و نفس البراميترز اوردر و نفس البراميتر تايب   
3- و هتلاحظ ان الديليجيتس و انت بتناديه بيبقى عامل ذي الاوبجيكت بالظبط تعالى بقه اوريك الكود بتاعه

namespace LearningCSharp

{

class Program

{

public delegate void MyDelegate(string Name);

static void Main(string[] args)

{

// Del\_Name Del\_Obj = new Del\_Name(FunctionName);

// Del\_Obj(Function\_param);

MyDelegate Del = new MyDelegate(MyFunc);

Del("Mahmoud Badawy");

}

public static void MyFunc(string Name)

{

Console.WriteLine("My name is : " + Name);

}

}

}

MB

################################################################

delegates video : 37-38

-----------------------------------

بص ياحج ده كود المفروض ان في اتنين كلاس الكلاس الاساسى و الكلاس بتاع الموظفين و هنا بوريك اذاى ان انا هعمل اتشيك عشان اوظف الموظفين مره بالسلارى و مره بالخبره

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

List<Employee> EmployeeList = new List<Employee>();

EmployeeList.Add(new Employee { ID = 1, Name = "Mahmoud Badawy", Salary = 20000, Experience = 10 });

EmployeeList.Add(new Employee { ID = 1, Name = "Said Foad", Salary = 12000, Experience = 5 });

EmployeeList.Add(new Employee { ID = 1, Name = "Ahmed Sadek Badawy", Salary = 10000, Experience = 4 });

EmployeeList.Add(new Employee { ID = 1, Name = "Mostafa Badawy", Salary = 15000, Experience = 6 });

Employee obj = new Employee();

Console.WriteLine("################# Experience #################");

obj.HireEmpByExp(EmployeeList);

Console.WriteLine("################# Salary #################");

obj.HireEmpBySalary(EmployeeList);

}

}

public class Employee

{

public int ID { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public int Experience { get; set; }

public void HireEmpByExp (List <Employee> EmpList) {

foreach(Employee Emp in EmpList)

{

if (Emp.Experience > 5)

{

Console.WriteLine("Hire : " + Emp.Name);

}

}

}

public void HireEmpBySalary(List<Employee> EmpList)

{

foreach (Employee Emp in EmpList)

{

if (Emp.Salary > 10000)

{

Console.WriteLine("Hire : " + Emp.Name);

}

}

}

}

}

طبعا ذي ما انت شايف مفيش اى ريوز في الموضوع بالعكس ده انت عشان تعمل اتشكايه صغيره تانيه رحت عامل فانكشن تانيه فالديلجيتس اتعملت عشان معدش العب كتير في اللوجيك جوه الكلاس ذي ما انت شايف فوق كده   
لأ هو فانكشن واحده الى بتتكتب و اللوجيك بتعها بغيره في الكلاس الاساسيه بالدليجيتس بقه

و أدى الكود بالدليحيتس هتلاقى فرق شاسع في الكلاس ال Employee في طريقه استخدام الكود مع اكثر من لوجيك كذا مره عادى

أول حاجه انته عاوزه يعمل اللوجيك انهى ؟؟

انه يعمل الاتشيك ان السالارى اكبر من 10000 لو كده يرجع الموظفين الى اكتر من 10000 و يعينهم عنده

يعنى معنى كده ان اللوجيك عباره عن تشكايه بترجع Boolean ؟؟ ايوه

اذا المفروض الفانكشن الى انا هعملها تتشيك بتاخد براميتر من نوع موظف تشوف جواه اذاى كان السالرى اكبر من 10000 يرجع فيها بولين ب true

اذا الدلجيتس هيبقى برضوا بياخد براميتر من نوع Employee و بيرجع true

public delegate bool HiredDelegate(Employee obj);

و دى هتبقى الفانكشن اى الدليجتس بيشاور عليها

public static bool IsHiredBySalary(Employee obj)

{

if(obj.Salary > 10000) {return true;}

else {return false;}

}

و ده اوبجيكت من الدليجيتس و بشاور فيه على الفانكشن ذي ما انت شايف اهه   
HiredDelegate SalaryDel = new HiredDelegate(IsHiredBySalary);

بعد كده بعمل الفانكشن الى المفروض هناديها في الكلاس ال Employee البراميترز بتوعها لسته الموظفين و الدليجيت الى هيتعمل و ده لان الدلجيت ده هو الى هعمل قصاده 1000 فانكشن و اجع بيه الداتا الى محتاجها الى هي بولين هنا عشان اعرف استخدمها

public void Hired\_Emp (List <Employee> EmpList , HiredDelegate IsHired) {

foreach(Employee Emp in EmpList)

{

if (IsHired(Emp))

{

Console.WriteLine("Hire : " + Emp.Name);

}

}

}

ذي ما انت شايف الفانكشن بتاخد الليست من الموظفين و بتلف عليهم و لو واحد فيهم اتبعت للفانكشن بتعته عن طريق الدليجيت و رجع ترو يطبعه اما لو رجع فولس مش هيطبعه

و ده بقه الكود كله متجمع و فيه الكلاسين و الدليجيت

namespace LearningCSharp

{

public delegate bool HiredDelegate(Employee obj);

class Program

{

static void Main(string[] args)

{

List<Employee> EmployeeList = new List<Employee>();

EmployeeList.Add(new Employee { ID = 1, Name = "Mahmoud Badawy", Salary = 20000, Experience = 10 });

EmployeeList.Add(new Employee { ID = 1, Name = "Said Foad", Salary = 12000, Experience = 5 });

EmployeeList.Add(new Employee { ID = 1, Name = "Ahmed Sadek Badawy", Salary = 10000, Experience = 4 });

EmployeeList.Add(new Employee { ID = 1, Name = "Mostafa Badawy", Salary = 15000, Experience = 6 });

HiredDelegate SalaryDel = new HiredDelegate(IsHiredBySalary);

HiredDelegate ExperienceDel = new HiredDelegate(IsHiredByExperience);

Console.WriteLine("################# Experience #################");

Employee.Hired\_Emp(EmployeeList, SalaryDel);

Console.WriteLine("################# Salary #################");

Employee.Hired\_Emp(EmployeeList, ExperienceDel);

}

public static bool IsHiredBySalary(Employee obj)

{

return (obj.Salary > 10000) ? true : false;

}

public static bool IsHiredByExperience(Employee obj)

{

return (obj.Experience > 5) ? true : false;

}

}

public class Employee

{

public int ID { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public int Experience { get; set; }

public static void Hired\_Emp(List<Employee> EmpList, HiredDelegate IsHired)

{

foreach (Employee Emp in EmpList)

{

if (IsHired(Emp))

{

Console.WriteLine("Hire : " + Emp.Name);

}

}

}

}

}

**اهه بالدلجيتس اللزيز و فيه طريقه تانيه بدل الاوبجيكت بتاع الدليجيت و هي انى استخدم اللندا اكسيريشن طبعا لا غنى عن استخدام الدليجيتس ذات نفسه و هو في الكول بيعتبر اللندا اكسبريشن عمتا كائنها اوبجيكت من نوع دليجيتس**

namespace LearningCSharp

{

public delegate bool HiredDelegate(Employee obj);

class Program

{

static void Main(string[] args)

{

List<Employee> EmployeeList = new List<Employee>();

EmployeeList.Add(new Employee { ID = 1, Name = "Mahmoud Badawy", Salary = 20000, Experience = 10 });

EmployeeList.Add(new Employee { ID = 1, Name = "Said Foad", Salary = 12000, Experience = 5 });

EmployeeList.Add(new Employee { ID = 1, Name = "Ahmed Sadek Badawy", Salary = 10000, Experience = 4 });

EmployeeList.Add(new Employee { ID = 1, Name = "Mostafa Badawy", Salary = 15000, Experience = 6 });

Console.WriteLine("################# Experience #################");

Employee.Hired\_Emp(EmployeeList, obj => obj.Experience > 5);

Console.WriteLine("################# Salary #################");

Employee.Hired\_Emp(EmployeeList, obj => obj.Salary > 10000);

}

}

public class Employee

{

public int ID { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public int Experience { get; set; }

public static void Hired\_Emp(List<Employee> EmpList, HiredDelegate IsHired)

{

foreach (Employee Emp in EmpList)

{

if (IsHired(Emp))

{

Console.WriteLine("Hire : " + Emp.Name);

}

}

}

}

}

MB

################################################################

multicast delegates

-----------------------------------

دى يا سيدى بقه ان ديلجيت واحده استخدمها فى انى اعمل انفوك او كول لعدد كبير من الفانكشنز فى وقت واحد بشرط ان يكون الريتيرن ليهم هو نفس الريتيرن تايب و البراميتر الى دخلاهم تبقى نفس العدد و النوع و و الاكسيس موديغاير برضوا متشابهه يعنى من الاخر لازم يبقى كلهم متشابهين فى نفس البناء بتعهم و طريقه كتبتها بقه ليها طريقتين

الطريقه الاولى طريقه الجمع و الطرح : الجمع بين الدليجيتس يعنى كل الى بجمعه مع بعضه هيرن مع بعضه لو طرحت واحده منهم معناها شيلها يا كبير من الانفوك بتاعك

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

DelGate Del1 = new DelGate(MyfuncOne);

DelGate Del2 = new DelGate(MyfuncTwo);

DelGate Del3 = new DelGate(MyfuncThree);

DelGate All\_Del = Del1 + Del2 + Del3;

All\_Del("Mahmoud");

DelGate All\_DelWithOut\_2 = Del1 + Del2 + Del3 -Del2;

All\_DelWithOut\_2("Mostafa");

}

public static void MyfuncOne(string Name1)

{

Console.WriteLine("Hello {0} From MyfuncOne", Name1);

}

public static void MyfuncTwo(string Name2)

{

Console.WriteLine("Hello {0} From MyfuncOne", Name2);

}

public static void MyfuncThree(string Name3)

{

Console.WriteLine("Hello {0} From MyfuncThree", Name3);

}

}

}

الطريقه الاخرى هى كذلك

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

DelGate All\_Del = new DelGate( MyfuncOne) + MyfuncTwo + MyfuncThree;

All\_Del("Mahmoud");

}

public static void MyfuncOne(string Name1)

{

Console.WriteLine("Hello {0} From MyfuncOne", Name1);

}

public static void MyfuncTwo(string Name2)

{

Console.WriteLine("Hello {0} From MyfuncOne", Name2);

}

public static void MyfuncThree(string Name3)

{

Console.WriteLine("Hello {0} From MyfuncThree", Name3);

}

}

}

من المحوظات المهمه بقه ان لو الريتيرن حاجه غير الفويد يعنى و ليكن الريتيرن سترينج او انتيجر و عملت المالتى كاستينج ديلجيت ليهم فهو فى الاخر الريتيرن بيبقى اخر ريتيرن من اخر فانكشن حطيتها فى الدليجيت و شكرا

امتى بنستخدم فعلا المالتى كاستينج دليجيت و بنستفاد منه فعليا ؟؟

فى ال Observer Pattern الى هو فيه Subscribe and unsubscribe and Publish

MB

################################################################

Exception Handeling

-----------------------------------

بص يا سيدى دى من الاخر كده لازم فى كل مشروع ايرورز تحصل فلما يحصل ايرور ما فى الكود فطبعا انت عشان الدنيا متخربش فى الموقع فالمفروض فى كل فانكشن تعمل حاجه اسمها   
try catch عشان تهندل فيها رد معين او تخزين فى اللوج للايرورز الى بتطلع بحيث يبقى عندك حاجه تخزن فيه و تعرف الايرورز الى فى المشروع و كذاك عشان تتفادى ان المشروع يقف تماما لما يحصل ايرور ما   
فانت بقه يا سيدى بتهندل اى ايرور ممكن تتوقعه فى الكود بتاعك مثلا Null Exception او FineNotfoundException لو ملقاش فايل انت عاوز تقرأ او تكتب فيه مثلا و كذلك SqlException لو فى مشكله فى اى حاجه تخص الداتا بيز

و هكذا فدلوقتى انا هوريك فى الكود اذاى بعمل تراى و كاتش و اذاى بهندل اى ايرور فى الكاتش و كدا

using System.IO; // For Files

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

try

{

StreamWriter SW = new StreamWriter(@"C:\TestFiles\File.txt"); // Open and Determine path for File to Write

SW.WriteLine("Hello Mahmoud"); //Write inside File (Overwrite) text in side file

SW.AutoFlush = true; //Auto Save

SW.Close(); // Close File After Wrtit

StreamReader SR = new StreamReader(@"C:\TestFiles\File.txt"); // Open and Determine path for File to Read

String Res = SR.ReadToEnd(); //Read All File and Put Return string in Result Var

Console.WriteLine(Res);

SR.Close(); // Close File

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); // To know Error Message

Console.WriteLine(ex.StackTrace); // To know The Error Line

}

}

}

}

ألكود الى فوق ده فيه الهاندل العادى و انا اصلا عامله صح تعالى بقه جربه و حطله path غلط للفايل و شوف هيطلعلك ايه فى الكاتش يا كبير

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

try

{

StreamReader SR = new StreamReader(@"C:\TestFiles\File11111.txt"); // Open and Determine path for File to Read

String Res = SR.ReadToEnd(); //Read All File and Put Return string in Res Var

Console.WriteLine(Res);

SR.Close(); // Close File

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); // To know Error Message

}

}

}

}

مش هيضرب البرنامج و هيمشى عادى خالص بس هيطلعلك رساله بالايرور الى حصل

بس انا عامل الاكسيسبنش عام اوى انا بقه راجل متمكن و عاوز اعمل لكل ايرور حاجه خاصه بيه يبقى اعمل كده يا باشا و مخليش الموضوع عايم اوى كده بدل ما اقوله Exception بس هقوله FileNotFoundException يبقى التغيره فى الحته دى

catch (FileNotFoundException ex)

{

Console.WriteLine(ex.Message); // To know Error Message

}

هنا بقه هنعمل كود مهندل فيه حاجتين مره لو الفايل مش موجود و مره لو الديريكتورى مش موجود يطلع رساله تانيه و انا عامل الرسايل مخصصه يعنى انا الى كاتبها بايدى مش اللغه

namespace CsharpLearning

{

class Program

{public delegate void DelGate(string Name);

static void Main(string[] args)

{

try

{

StreamReader SR = new StreamReader(@"C:\TestFiles\_\_\_\_\_1\File11111.txt"); // Open and Determine path for File to Read

Console.WriteLine(SR.ReadToEnd());

SR.Close(); // Close File

}

catch (FileNotFoundException ex)

{// Custom Error Message

Console.WriteLine("Check if File {0} in Path is Exist Or Not ", ex.FileName);

}

catch (DirectoryNotFoundException ex)

{// Custom Error Message

Console.WriteLine("Check if Directory in Path is Exist Or Not");

}

}

}

}

نيجى بقه لجزئيه ال Finaly دى بقه يا سيدى الى بحط فيها الحاجه الى عوزها تترن سواء فى ايرور او مفيش و هنا انا هخليه يقفل الفايل فى كل الاحوال عشان ميستهلكش من الرامه فكتب فى الفاينالى كود القفل بتاع الفايلات

أول كود و الى فى حالته انا بقوله اعملى تراى لو لقيت ايرور طلعلى رساله الاكسيبشن و بعد كده نفذ الفاينالى انا بقه داخل الفاينال قولتله لو الاستريم ريدر جايلك Null قولى ان الاستريم ريدر بنال و خلاص غير كده اقفل الفايل

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

// To assign StreamReader Var Give it Null (it is mean it still Close)

StreamReader SR = null;

try

{

SR = new StreamReader(@"C:\TestFiles\File11111.txt");

// Open and Determine path for File to Read

Console.WriteLine(SR.ReadToEnd());

}

catch (FileNotFoundException ex)

{// Custom Error Message

Console.WriteLine("Check if File {0} in Path is Exist Or Not ", ex.FileName);

}

finally

{

if(SR != null)

{

SR.Close(); // Close File

Console.WriteLine("File Is Closed Successfully");

}

else

{

Console.WriteLine("Stream Reader is Null");

}

}

}

}

}

اخر ملحوظه بقه انك لو هتكتب اكسيبشن مخصوصه ذى FileNotFoundException و معاها الاكسيبشن العام بتاع كله الى هو Exception بيقولك بقه انك لازم ساعتها تكتب الاكسيبشن العام اوى ده أخر حاجه قبل ال Finally

MB

################################################################

Inner Exception video 41 :

-----------------------------------

بص يا سيدى ده بيبقى فى ايرور اساسى حصل و هو فى الاول بيبقى كارنت ايرور اكسيبشن راح مدخلنى فى الكاتش و لما دخل فى الكاتش حصل جوه الكاتش ايرور تانى بقى هو الكارنت ايرور و الاولانى خالص بقى دلوقتى inner و ده لان من الاخر الاينر اكسييبشن هو الاكسيبشن الاساسى الى اتسبب فى الكارنت اكسيبشن الحالى و عشان تشوف رساله الاينر اكسيبشن بتبعت رساله الاينر جوه رساله الكارنت تعالى اوريك مثال   
1- اول ايور انى هقسم على 0 فهيدخلنى على كاتش A  
2- جوه الكاتش A هعمل فايل لوج عشان اكتب جواه الايرورز الى بتطلع ماشى بس للاسف حصل ايرور لانى اديته فايل مش موجود ساعتها المفروض يعمل كاتش تانيه عشان يطلع فيه ايرور جديد فى New exception   
حاليا ايرور الفايل بقى الكارنت ايرور و الايرور بتاع القسمه على الصفر بقى هو الاينر   
3- عشان تشوف رساله الاينر بقه لازم تبعت الاكسيبشن بتعها جوه قوص الكارنت الحالى   
تعالى اوريك الكود بقه

namespace CsharpLearning

{

class Program

{

public delegate void DelGate(string Name);

static void Main(string[] args)

{

int num1 = 5;

int num2 = 0;

try

{

try

{

Console.WriteLine(num1 / num2);

}

catch (Exception innerExcep)

{

string FilePath = @"C:\TestFiles\File\_\_\_\_\_.txt";

if (File.Exists(FilePath))

{

StreamWriter SW = new StreamWriter(FilePath);

SW.WriteLine(innerExcep.GetType().FullName);

SW.WriteLine(innerExcep.Message);

SW.Close();

}

else

{

throw new FileNotFoundException(FilePath + "Not Found", innerExcep);

}

}

}

catch ( Exception currentExep)

{

Console.WriteLine("Current Exception : " + currentExep.GetType().Name);

Console.WriteLine("Inner Exception : " + currentExep.InnerException.GetType().Name);

}

}

}

}

MB

################################################################

Custom Exception

-------------------------------

امتى نستخدم الكاستوم اكسيبشن ؟؟ اذا كان الاكسيبشنز الى موجودين في الوقت الحالي مش بيوصفوا الاكسيبشن بشكل كافى و انت عاوز تفهم التستر او الكودر الى بعدك عن الايرور فهم عميق لو الايرور حصل

مثال : عمل اكسيبشن لو انا عملت لوج ان مره و جيت فتحت تابه تانيه و دخلت لوج ان تانى يروح عامل ايرور و يقولى مينفعش تعمل لوج ان مرتين ياحج انت جاى تهزر هنا

بص من الاخر في الكود بقه انت بتعمل للاكسيبشن الجديد كلاس و بيورث من ال exception Class و انت بتعمل الاكسيبنش الجديد بقه بتقعد تقلد كيفيه عمل الاكسيشن العام و تعمل زيه بس بتغيير في ال throwing للرسائل بتاعت الايرور بس

و أدى الكود الى بيوريك تعمل اذاى كاستوم اكسيبشن ياعم

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

try

{

throw new MahmoudBadawyLoginException("Can Not Log in 2 Times");

}

catch (MahmoudBadawyLoginException MBL\_Ex)

{

Console.WriteLine(MBL\_Ex.Message);

}

}

}

class MahmoudBadawyLoginException : Exception

{

public MahmoudBadawyLoginException()

{

}

// Those Constractors ==> Inherit Constractor From Base Class

public MahmoudBadawyLoginException(string message):base(message)

{

}

public MahmoudBadawyLoginException(string message, Exception innerException) : base(message, innerException)

{

}

}

MB

################################################################

Exception Handling Abuse

------------------------------------

ده يعنى من الاخر كده ان التيستر يعمل تيست على البرنامج و يطلعلى 3 أنواع ايرور فروح انا ذي الاهبل اعملهم كاتش و خلاص و ده بقه الاستخدام السئ جدا جدا للكاتش لان في الأصل المفروض تعمل Validations تخليك متوصلش للكاتش أساسا و يبقى الايرورز الى بتحصل معدومه عشان كتر التراى و الكاتش خصوصا لو انت عامل لوج فايل بيتقل السيرفر جدا

تعالى اوريك بقه مثال لل Exception Handling abuse انى مره اقسم رقم على صفر و مره ادخل ارقام حجمها اكبر من حجم الانتيجار فاريبولز و مره ادخل استرينج بدلا من انتيجر الكود اهه

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Enter Num1 : ");

Int32 Num1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Num2 : ");

Int32 Num2 = Convert.ToInt32(Console.ReadLine());

Int32 Res = Num1 / Num2;

Console.WriteLine("the Result is : {0}", Res);

}

catch (DivideByZeroException DBZex )

{

Console.WriteLine("You Can Not Divide by Zero Change Num2");

}

catch (OverflowException SOFex)

{

Console.WriteLine("Num1 and Num2 must be Between {0} and {1}" ,Int32.MinValue, Int32.MaxValue);

}

catch (FormatException Formatex)

{

Console.WriteLine("Num1 and Num2 must be Int32");

}

catch (Exception ex)

{

// For Errors we do not Expect or Know

Console.WriteLine(ex.Message);

}

}

}

}

كده طبعا 1000 كاتش هههههههه حرام بجد لازم يبقى في فاليديشن بدل البهدل دى و امنع الايرور من الأول انه يحصل أصلا و هنا هستخدم التراى بارس الى هو يحاوص يحول القيمه الى انا ادتهاله لو عرف يحولها لانتيجر يبقى فل الفل و يعدى معرفش يبقى انا مدخلهاله غلط فطلعله رساله و من الاخر هتعمل فاليديشن انه يتشك لو مش رقم او رقم كبير او الى هقسم عليه صفر يطلعلى رساله انه يظبط القيم الى مدخلها الأول عشان البرنامج يشتغل صح

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

try

{

Int32 Num1, Num2;

Console.WriteLine("Enter Num1 : ");

bool Num1\_isNumber = Int32.TryParse(Console.ReadLine(), out Num1);

Console.WriteLine("Enter Num2 : ");

bool Num2\_isNumber = Int32.TryParse(Console.ReadLine(), out Num2);

if(Num1\_isNumber && Num2\_isNumber)

{

if(Num2 !=0)

{

Int32 Res = Num1 / Num2;

Console.WriteLine("the Result is : {0}", Res);

}

else

{

Console.WriteLine("Num2 must not be Zero");

}

}

else

{

Console.WriteLine("Num1 Or Num2 must be Int32 between {0} : {1}", Int32.MinValue,Int32.MaxValue);

}

}

catch (Exception ex)

{

// For Errors we do not Expect or Know

Console.WriteLine(ex.Message);

}

}

}

}

MB

################################################################

Why Enums ?

------------------------------------

بص يا سيدى انت عندك حاجات ذي مثلا ال Gender و ليكن انا عملها لوك اب بحيث ان   
رقم 0 معناه Male  
رقم 1 معناه Female

رقم 2 معناه Unknow  
فانا دلوقتى المفروض هخزنه في الداتا بيز على انه انتيجر مع ان انا هحتاج اظهره كسترينج فهضطر ساعتها اعمله Switch cases ففي الحاله دى الاحسن طبعا استخدم ال Enum عشان الكود يبقى سهل التعديل فيه و سهل قرايته لان انا هستخدم الكلام ده على مستوى كبير جدا في المشاريع الكبيره فمش عاوز لغبطه بدل ما اعد اكتب ارقام و بتاع لا بكتب الحاجه بالاينم و هو الانم بيخزن الداتا بتعته على انها Enum فالمفروض انا احولها الى انتيجر لما اجى اخزنها في الداتا بيز عادى خالص اما لو هظهرها على الشاشه فبحولهاش ولا حاجه بستخدمها ذي ما هي عشان هي بتبين اسمها على طول من غير سويتش كيس و سويتش بتنجان يعنى بتبسط الكود خالص

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

Person[] PerObj = new Person[3];

PerObj[0] = new Person

{

Name="Mahmoud Badawy",

Gender = 0

};

PerObj[1] = new Person

{

Name = "Soso",

Gender = 1

};

PerObj[2] = new Person

{

Name = "Niazy",

Gender = 2

};

foreach (Person Per in PerObj)

{

Console.WriteLine("Name is {0} - Gender is : {1}", Per.Name, GetGender(Per.Gender));

}

}

public static string GetGender(int GenderID)

{

// Gender 0 Male , 1 Female , 2 Unknown

switch (GenderID)

{

case 0 :

return "Male";

break;

case 1:

return "Female";

break;

case 2:

return "Unknown";

break;

default:

return "Nothing";

}

}

}

class Person

{

public String Name { get; set; }

public int Gender { get; set; }

}

}

طيب انا عاوز الكود ده بقه يتكتب بال Enum عشان اوريك حلاوه ال Enum وفيدته

الكود بال Enum هتبقى بتستخدمه ذي ما انت شايف من غير تحويل لانتيجر اما لما تيجى هتخزن في الداتا بيز يا كبير ساعتها حول الى راجعلك من ال enum ل Int

namespace LearningCSharp

{

public enum GenderEnum : int

{

Male = 0,

Female = 1,

Unknow = 2

}

class Program

{

static void Main(string[] args)

{

Person[] PerObj = new Person[3];

PerObj[0] = new Person

{

Name="Mahmoud Badawy",

Gender = GenderEnum.Male

};

PerObj[1] = new Person

{

Name = "Soso",

Gender = GenderEnum.Female

};

PerObj[2] = new Person

{

Name = "Niazy",

Gender = GenderEnum.Unknow

};

foreach (Person Per in PerObj)

{

// In Save in SQL use Convert Enum to Int

Console.WriteLine("Name is {0} - Gender is : {1}", Per.Name, Convert.ToInt32(Per.Gender));

// In Showing it Normaly Do not Convert

Console.WriteLine("Name is {0} - Gender is : {1}", Per.Name, Per.Gender);

}

}

}

class Person

{

public String Name { get; set; }

public GenderEnum Gender { get; set; } // We change Gender From int to Enum

}

}

لاحظ برضوا ان انا ممكن ادي لل Enum ارقام انتيجر و ممكن اسيبه و مديلوش ارقام و ساعتها هو هياخد ارقام الانديكس باى ديفولت يعنى يبدأ أول عنصر يكون ب 0 و يزيد بمقدار واحد بقه كل اما اعمل عنصر داخله

namespace LearningCSharp

{

public enum GenderEnum : int

{

Male = 0,

Female = 1,

Unknow = 2

}

class Program

{

static void Main(string[] args)

{

int[] GenderEnumInt = (int[])Enum.GetValues(typeof(GenderEnum));

string[] GenderEnumString = Enum.GetNames(typeof(GenderEnum));

foreach (int Per in GenderEnumInt)

{

Console.WriteLine("Enum Value : {0} ", Per);

}

foreach (string Per in GenderEnumString)

{

Console.WriteLine("Enum String : {0} ", Per);

}

}

}

}

في الكود الى فات ده انا بعمل كونفيرت للاينم لانتيجر واخزنه في الاراى اوف انتيجر و مره بخزنه في الاراى اوف سترينح لو برجع الاسترينج بتاعه و بعد كده بظهره بص على الكود و هتفهم بقه

MB

################################################################

Type vs type members

------------------------------------

ال Types : الداتا تايب من الاخر و ينفع نقول عليها Class,Enum, Struct , Delegate, interface

sو دى الاكسيس موديفاير الخاصه بيها فقط internal , Public

ال Types members : اى حاجه جوه الكلاس ذي الفانكشنز البروبتيس الفيلدس و الكونستراكتور و دى تقبل جميع أنواع الاكسيس موديفاير internal , protected , internal protected , Private , public

MB

################################################################

Region

------------------------------------

ده ياعم بتعمل منطقه كده تبقى كاتب فيها كذا كود معين ليهم علاقه ببعض و بتديهم اسم معين عشان يسهل عليك السيرش عليهم او الوصول ليهم و السنتاكس بتاعه اهه

#region Enums

public enum GenderEnum : int

{

Male = 0,

Female = 1,

Unknow = 2

}

public enum jobTitleEnum : int

{

Jenior = 0,

senior = 1,

TeamLeader = 2,

}

#endregion

MB

##################################################################

Attribute

---------------------

بص يا سيدى دى عباره عن حاجه كده اتعملت عشان تعرفك بعض المعلومات التوضيحيه عن برنامجك و من الاخر احنا بنستخدمها كتير اننا نعمل بيها رولز معينه لبعض الميودز   
فهتلاقى فى اتربيوتس كتير جدا و اغلبهم بتكتبوا فوق الفانكشن كده فلما بكتبهم فوق الفانكشن الى بيحصل ساعتها ان تأثيرها بيبان على شغل الفانكشن دى على سبيل المثال عندنا اتربيوت اسمه

Obsolete و ده طبعا عشان تبقى فاهم فشخ و تعرف بعد كده تتبع الاتربيوت و تعرف بياخد ايه و بيخدش ايه بص يا سيدى اكتب الاتربيوت كده [Obsolete] او [ObsoleteAttribute] و بعد ما تكتبه دوس بقه على f12 عشان يوديك على الكلاس بتعته فتعرف الكونستراكتورز بتعته و بالتالى تعرف بياخد ايه بالظبط

الاتربيوت ده بستخدمه على فانكشن لما بيكون الفانكشن دى قديمه و مش هينفع امسحها بشكل نهائى حاليا و محتاج اعرف الديفولوبرز الى بعدى انها حاجه دبريكيتيد و الاحسن يشتغلوا بفانكشن تانيه انا عاملها جديده و اقولهم عليها

و اتربيوت ال Obsolete ده برضوا بياخد جواه براميترز هعرفك ايه فيدتها بس لما اكتب الكود أحسن

namespace CsharpLearning

{

class Program

{

static void Main(string[] args)

{

MyName1();

MyName2();

}

[Obsolete("MyName1() Func is Duprecated please Use new Func It is MyName2()" , true)]

static public String MyName1()

{

return "Mahmoud Badawy";

}

static public String MyName2()

{

return "Mostafa Badawy";

}

}

}

جرب البرنامج الى فوق هتلاقيه على فانكشن رقم واحد راح عاملك علامه الايرور و قالك الرساله بتاعت ان الفانكشن الاولى دبريكيتيد و ياريت تستخدم الفانكشن التانيه و بالمناسبه لو عاوز تخليه يعمل ورنينج بس خلى True ب false

MB

##################################################################

Reflection : Video 53 -54

--------------------------------

دى أهميتها انك تقدر تتعامل مع الكلاسيس و الفانكشنز و البروبرتيس و الكونستراكتورز اثناء ال Run Time و ده أهميته اك تقدر تغير في الحاجات دى او تعرضها و الابلكيشن شغل فطبعا هيبقى استخدام الريفليكشن بشكل اساسى في الديزاين بالسى شارب و في انك تعمل حاجه كده ذي ال visual studio هتلاقيهم عرضينلك البروبرتيس على جنب كده عشان تشوفها و كمان تقدر تغير فيها براحتك فتغير حاجات في الكود و انت بترنه

و الاستخدام التانى انك ممكن تستخدمه في ال Late Binding و ده معناه انك تقدر تعمل اوبجيكت من كلاس و ترنه و انت لسه مانشاتش الكلاس أصلا و يبقى بيتم انشاء الكلام ده اثناء الران تايم على العموم هنستفيض في شرحها بعدين

التعامل مع الريفليكشن بيبقى بشكل كبير تعامل مع الكلاس و الى جواه اثناء الران تايم و بمعىن انه دايما بيتعامل مع الاسيبملى فايل و ده كود بيوريك اذاى تعرف تعرض محتويات كلاس ما و انت شغال

ذي ما انت شايف الكود اهه بقدر اجيب بيه اسامى الفانكشنز و الكلاس و البروبتريس و الكونستراكتورز و الفيلدس و كل حاجه في الكود عادى و على فكره بقدر اعمل للكلام ده سيت برضوا بس انا مهندلتش الموضوع ده في الكود ابقى عيش و افهم انت بقه

using System;

using System.Reflection;

namespace LearningCSharp

{

class Program

{

static void Main(string[] args)

{

// It is Noraml Code That Use Early Binding

Customer C1 = new Customer();

Console.WriteLine(C1.MyName());

// It is Reflection Code That Use Late Binding

Type T = Type.GetType("LearningCSharp.Customer");

// Retern the FullName Of Class I will Use

Console.WriteLine(T.FullName); // return LearningCSharp.Customer

Console.WriteLine(T.Name); // return Class Name ( the Type) Customer

Console.WriteLine(T.Namespace); // return NameSpace LearningCSharp

Console.WriteLine("The Proprties are : "); // Properties Will Return in Array of PropertyInfo

PropertyInfo[] Properties = T.GetProperties();

foreach(PropertyInfo Prop in Properties)

{

Console.WriteLine("Property Name : {0} and Type : {1} " ,Prop.Name , Prop.PropertyType.Name);

}

Console.WriteLine("The Methods are : "); // Methods Will Return in Array of MethodInfo

MethodInfo[] Methods = T.GetMethods();

foreach (MethodInfo Meth in Methods)

{

Console.WriteLine("Method Name : {0} and Return Type : {1} ", Meth.Name, Meth.ReturnType.Name);

}

Console.WriteLine("The Constructors are : "); // Constructors Will Return in Array of ConstructorInfo

ConstructorInfo[] Constructors = T.GetConstructors();

foreach (ConstructorInfo Cnstr in Constructors)

{

Console.WriteLine("Constructor Name : {0}", Cnstr.ToString() );

}

}

}

public class Customer

{

public int X;

public String Y;

public int ID { get; set; }

public String Name { get; set; }

public Customer()

{

this.ID = this.MyID();

this.Name = this.MyName();

Console.WriteLine( this.FullName("Mostafa", "Badawy"));

}

public Customer(string X)

{

Console.WriteLine(X) ;

}

public string MyName()

{

return "Mahmoud Badawy";

}

public int MyID()

{

return 1;

}

public string FullName(string FName, string LName)

{

return string.Format("My Name is : {0} {1}", FName, LName);

}

}

}

MB

##################################################################

Late Binding

---------------------

بص يا سيدى الطبيعى انك بتعمل كلاس و بعد ما تعمل الكلاس بتروح تعمل منه اوبجيكت لو معملتش كلاس و جيت تعمل منه اوبجيكت فبيروح عاملك ايرور صح كده ؟؟ ايوه صح   
ده ياعم اسمه ال Early Binding

ال late binding ده يا صاحبى انك ممكن تعمل انستانس او ابوجيكت من الكلاس قبل ما تنشاه أصلا و يتم انشاء الكلاس في ال Assemply file و الكود بيترن لو عاوز تفهم الكلام ده شايف المثال الى فوق دهه انا مستخدم فيه ال Early and Late Binding جرب تعمل كومنت للكلاس customer هتلاقى الاوبجيكت الى معمول بالايرلى بايندينج عمل ايرور بينما باقى الكود شغال فل الفل مش هيطلع ايرور   
بس مع ذلك في الحالتين عشان الدنيا تشتغل صح لازم الكلاس يبقى موجود

تعالى بقه اديك مثال اوريك فيه اذاى تقدر تعمل اوبجيكت و تعمل فانكشنز و تعمل بروبرتيس في ال late Binding

using System;

using System.Reflection;

namespace CsharpLearning

{

class Program

{

static void Main(string[] args)

{

//Early Binding (Normal)

Person Obj = new Person(); // Create Instance

Obj.GetFullName("Mahmoud","Badawy"); // Call Function

//Late Binding

// Open Assembly Executing

Assembly Exec\_Assembly = Assembly.GetExecutingAssembly();

// Refer To Class you need to Get

Type PersonClassType = Exec\_Assembly.GetType("CsharpLearning.Person");

//Create Instance From Person Class

Object PersonInstance = Activator.CreateInstance(PersonClassType);

//Get Determined Method and Create It's Param

MethodInfo GetFullName = PersonClassType.GetMethod("GetFullName");

string[] Params = new string[2];

Params[0] = "Mostafa";

Params[1] = "Mahmoud";

string FullName = (string)GetFullName.Invoke(PersonInstance, Params);

Console.WriteLine(FullName);

}

}

class Person

{

public void GetFullName(string FName , string LName)

{

Console.WriteLine("Hi : {0} {1}", FName, LName);

}

}

}

MB

##################################################################

Generic

---------------------

بص يا سيدى نشوف بقه الجينيرك ده اتعمل ليه بس عن طريق الكود لان مش هيتفهم كده

اولا انا عاوز اعمل فانكشن يكون بقارن فيه رقمين او اتنين استرينج و اشوف لو الاتنين متساويين و ذى بعض اقوله تمام و لو الاتنين مختلفين اقوله لأ مش تمام يا حوده

طب دى تتعمل اذاى ؟؟ من الواضح انى مش هعرف اقوله ان البراميترز انتيجر لان انا ممكن ادخله سترنيج و كذلك مينفعش اقوله انهم سترينج لانى ممكن ادخله انتيجر

طب و العمل ؟؟

**استخدم البراميتر من نوع Object طيب دلوقتى بعد ما عملناه من نوع اوبجيكت بقت الدنيا هيصه و بقى ينفع اقارن انتيجر ب انتيجر و استرينج ب استرينج و اى حاجه باى حاجه**

**بس فى دلوقتى مشكله انى ممكن ادخل براميتر من نوع سترينج و التانى من نوع انتيجر**

**احيه ده كده هبل ياعم طب احل المشكله دى اذاى ؟؟**

استخدم يا باشا الجينيرك لان الجينيرك هيتيح ليك انك تدى للفانكشن اى نوع انت عاوزه بس بشرط ان يكونوا كلهم من نفس النوع مش حاجه كده و حاجه كده لانك بمجرد ما بتكتبله انه الحاجه دى انتيجر بيروح شغال على انتيجرز بس او تقوله سترينج يروج شغال استرينج بس

امتى استخدم الجينيرك ؟؟ لما يكون عندى فانكشن و الفانكشن دى كل البراميتارز الى داخله ليها متشابهه فى الداتا تايب   
و لما يكون محتاج ان الفانكشن دى ميبقاش داخل ليها نوع معين من الداتا تايب يعنى عاوز ادخلها مره البراميتارز من نوع انتيجر و مره من نوع سترينج و مره من نوع دوبل و مره من نوع اراى و مره من نوع كلاس معقد

static void FuncName(int X, int Y){}

static void FuncName(string X, string Y) {}

static void FuncName(double X, double Y) {}

static void FuncName(CustomTypeCls X, CustomTypeCls Y) {}

امتى استغنى عن الجينيرك و استخدم بداله نوع Object ؟؟

لما يكون المدخلات او البراميتراز بتاعت الفانكشن عاوز ادخلها مختلفه فى الداتا تايب

يعنى الاكس يبقى انتيجر و الواى سترينج او اى داتا تايبس مختلفه يعنى او حتى متشابهه عادى مثلا ذى كده

static void FuncName(int X, string Y){}

static void FuncName(string X, int Y){}

static void FuncName(string X, string Y){}

ساعتا هلجأ ل دى

static void FuncName(Object X, Object Y) {}

و مش هلجأ للجينيرك لان الجينيرك بيبقى T و T دى لما اقول انها من نوع int فالكومبايلر بيشوف اى حاجه احنا حطينها ب T انها int يعنى بيوحد كل ال T بالنوع الى انا اشرت ليه

طريقه كتابه الفانكشن بقه بالجينيرك بتبقى كده

static void FuncName<T>( T X, T Y) {}

يعنى اسم الفانكشن ثم حرف T وسط <> و ده المكان الى بحدد منه بعد كده الفانكشن دى الداتا تايب الى هتخش فى البراميتارز الى نوع T هتبقى نوعها ايه بعد كده

ثم بحط البراميتارز الى جوه الفانشكن و اشوف مين عاوزه يبقى حاجه جينيرك و بروح حاطط T بدل الداتا تايب العاديه

لما باجى اناديه بقه بناديه كده

FuncName<int>(1, 1);

< int >

ده معناه انى بقوله يا فانكشن الى هيجيلك فى البراميتارز الجينيرك T حاجات int

مينفعش ساعتها الحاجه الى كاتب الداتا تايب بتعها T انى احط فيها اى حاجه تكون مش انتيجر لأ لازم تبقى طالما كانت واخده T يبقى تاخد انتيجر ذى ما حددنا للجينيرك

طب عملنا الفانكشن كده

FuncName <string>("1", "1");

<string>

ده معناه انى بقوله يا فانكشن الى هيجيلك فى البراميتارز الى الجينيرك T حاجات string

طب لو انا عملت الفانكشن كده

static void FuncName<T>( T X, int Y) {}

ده معناه ان X جينيرك ينفع ادخل فيها اى نوع انما ال Y دائما int

ملحوظه اخيره ال T دى هى الجينرك داتا تايب و دى ينفع اكتبها اى كلام بدل ال T طالما نفس السنتاكس يعنى ينفع اكتب الفانكشن الجينيرك كده

static void FuncName<MBadawyType>( MBadawyType X, MBadawyType Y) {}

class Program

{

static void Main(string[] args)

{

AreEqual\_int(1, 1); // Equal

AreEqual\_string("1", "1"); // Equal

AreEqual\_Object(1, 1); // Equal

AreEqual\_Object("1", "1"); // Equal

AreEqual\_Object("1", 1); // not Equal

// AreEqualByGeneric<int>(1, "1"); // Can not Writen With Diffrent Types

AreEqualByGeneric\_T<int>(1, 1); // Equal

AreEqualByGeneric\_T<string>("1", "1"); // Equal

// AreEqualByGeneric<int>(1, "1"); // Can not Writen With Diffrent Types

AreEqualByGenericFirstParam<string>("1", 1); // not Equal

AreEqualByGenericFirstParam<int>(1, 1); // Equal

}

static void AreEqual\_int(int X, int Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

static void AreEqual\_string(string X, string Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

static void AreEqual\_Object(Object X, Object Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

static void AreEqualByGeneric\_T<T>(T X, T Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

static void AreEqualByGenericFirstParam<T>(T X, int Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

}

**خلى بالك ان ينفع اعمل الريتيرن بتاع الفانكشن هو كمان جينيرك يعنى اقدر ابعت جينيرك و اعمل ريتيرن بحاجه جينيرك و الفاانكشن ساعتها بتتكتب بالشكل ده**

static T FuncName<T>( T X, T Y) {}

و لما اجى اناديها بناديها كده

int First = FuncName <int>(1, 1);

بما انى حطيت فى اقواس الجينيرك انتيجر فده معناه انى هرجع انتيجر فى الجينيرك ريتيرن

string second = GenericReturn<string>("1", "1");

اما لو حطيت استرينج يبقى لازم ارجع استرينج

طب دلوقتى انا جوه الفانكشن هقوله رجعلى انتيجر ولا رجعلى استرينج اذاى ما هو هيبقى بيقولى رجع جينيرك ؟؟

و انت بتعمل ريتيرن هتحول النتيجه من جينيرك للحاجه الى انت عاوز ترجعها

و ادى مثال يا باشا بقه

class Program

{

static void Main(string[] args)

{

int First = GenericReturn<int>(1, 1);

Console.WriteLine(First);

string second = GenericReturn<string>("1", "1");

Console.WriteLine(second);

int[] Arr1 = { 10, 20, 30, 40 };

int[] Arr2 = { 3, 4 };

int[] third = GenericReturn<int[]>(Arr1, Arr2);

Console.WriteLine(third);

}

static T GenericReturn<T>(T X, T Y)

{

if (X.GetType()==typeof(string))

{

return (T)Convert.ChangeType("String", typeof(T));

}

else if (X.GetType() == typeof(int))

{

return (T)Convert.ChangeType(123, typeof(T));

}

else

{

int[] Arr1 = { 3, 4 };

return (T)Convert.ChangeType(Arr1, typeof(T));

}

}

}

طبعا ينفع اعمل البراميتارز انوعهم عاديه خالص و الريتيرن هو الى اخليه جينيرك برحتى

ذى كده

static T FuncName<T>( int X, string Y) {}

و كده الريتيرن بس هو الى جينيرك يا اسطى

**ملحوظه يمكن استخدام الجينيرك على الكلاسيز و الانترفيسيز و الديلجيتس و بالتالى لو هطبها هنا على الكلاس الكود هيتكتب كده هيتغير خد بالك**

**عمتا اقواس الجينيرك بتتكتب بعد اسم الفانكشن او بعد اسم الكلاس عشان تبقى واخد بالك بس**

class Program

{

public static void Main(string[] args)

{

bool Res = Calculator<int>.AreEqualByGenericParam(1, 1);

Console.WriteLine(Res);

Res = Calculator<bool>.AreEqualByGeneric\_All(true, true);

Console.WriteLine(Res);

int xRes = Calculator<int>.AreEqualByGeneric\_Ret(1, "5");

string sxRes = Calculator<string>.AreEqualByGeneric\_Ret(2, "5");

Console.WriteLine(xRes);

Console.WriteLine(sxRes);

}

}

public class Calculator<T>

{

public static bool AreEqualByGenericParam(T X, T Y)

{

return X.Equals(Y) ? true : false;

}

public static T AreEqualByGeneric\_Ret(int X,string Y)

{

if(X == 1)

{

return (T)Convert.ChangeType(X, typeof(T));

}

else

{

return (T)Convert.ChangeType(Y, typeof(T));

}

}

public static T AreEqualByGeneric\_All(T X, T Y)

{

bool Res = X.Equals(Y);

return (T)Convert.ChangeType(Res, typeof(T));

}

}

من هنا نفهم ان قوس الجينيرك لما يتحط فى الكلاس ده معناه انى كانى حطيته بعد كل الفانكشنز الى فى الكلاس و بالتالى اقدر اكتب جوه الفانكشنز دى التايب الجينيرك على طول من غير ما اكتب اقواس الجينيرك بعد اسم الفاناكشن و طبعا اقدر اعمل فانكسنز عاديه تمام مش بالضروره ابقى بكتب فانكشنز جينيرك و بس يعنى

class Program

{

public static void Main(string[] args)

{

Calculator<string> x = new Calculator<string>();

string Res = x.NormalFunc(1, 1);

Console.WriteLine(Res);

bool boolRes = x.Generic\_Func("1", "1");

Console.WriteLine(boolRes);

}

}

public class Calculator<T>

{

public string NormalFunc(int X, double Y)

{

return "Hello string";

}

public bool Generic\_Func(T X, T Y)

{

return X.Equals(Y);

}

}

ملحوظه مهمه جدا بقه لما تيجى تعمل شغل جينيرك اعمل يا اما البراميتار بس هى الى جينيرك و الريتيرن حاجه انت محددها او اعمل الريتيرن هو الى جينيرك و البراميتراز حاجات انت محددها

انما متعملش الاتنين ذى بعض الا لو انت متاكد ان نوع الحاجه الى هيدلوخوها هى هى نفس نوع الحاجه الى هيخرجوها فى كل الاحوال

و الشائع عمتا اننا بنستخدم الجينيرك مع البراميتار فى الانبوت و الريتيرن بنحدده و مبنعملوش جينيرك عشان ميبقاش اللعب كتير فى الحته دى

MB

##################################################################

Why should you override ToString ?

---------------------

عشان فى اوقات بنبقى عاوزين نحول الاوبجيكت كله مره واحده لاسترينج فده مش هيحصل الا لو انا عملت اوفر رايد للفانكشن يا باشا و ادى الى انا اقصده عشان تفهم

using System;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

Person Per = new Person();

Console.WriteLine(Per.ToString());

}

}

public class Person

{

public string FName = "Mahmoud";

public string LName = "Badawy";

public override string ToString()

{

return this.FName + " " + this.LName;

}

}

}

MB

##################################################################

Why should you override Equals ?

---------------------

ده لان انا لو جبت اتنين ابوجيكت من نفس الكلاس و اديت الفيلدز بتعتهم نفس البيانات و جيت اقارن بينهم هو مش هيقارن بالفاليو هو هيقارن بالريفرنس الى هما بصين عليه فهيلاقيهم مختلفين لان معروف ان الاوبجيكت ريفرنس تايب مش فاليو تايب فهدينى False حتى لو هما متساويين و عشان احل المشكله دى لازم اعمل اوفرايد للاكوال ميثود بنفسى

using System;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

Person Per1 = new Person();

Per1.FName = "Mahmoud";

Per1.LName = "Badawy";

Person Per2 = new Person();

Per2.FName = "Mahmoud";

Per2.LName = "Badawy";

Console.WriteLine(Per1.Equals(Per2));

}

}

public class Person

{

public string FName ;

public string LName ;

public override bool Equals(object obj)

{

if(obj == null)

{

return false;

}

else if ( !(obj is Person))

{

return false;

}

else

{

return this.FName == ((Person)obj).FName && this.LName == ((Person)obj).LName;

}

}

public override int GetHashCode()

{

return this.FName.GetHashCode() ^ this.LName.GetHashCode() ;

}

}

}

MB

##################################################################

. Convert ToString VS ToString

---------------------

Convert.ToString() أحسن من ToString() ليه بقه يا سيدى ؟؟

لان الكونفيرت توسترينج لو جالها حاجه ب Null بتحول ال Null الى “” بينما Tostring بتضرب البرنامج و تعمل ايرور على طول

و ادى كود مكتوب فيه الاتنين واحده هتعدى و واحده هتضرب عشان تصدق كلامى

using System;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

Person Per1 = new Person();

string Res = Convert.ToString(Per1.FName); // Null Will be ""

Console.WriteLine("it is : " + Res);

Res = Per1.FName.ToString(); // Null will Make Error

Console.WriteLine("it is : " + Res);

}

}

public class Person

{

public string FName;

public string LName;

}

}

MB

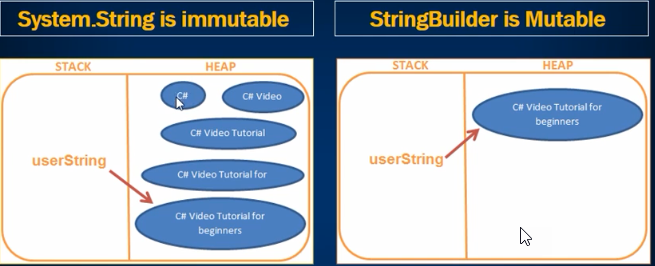
##################################################################

String VS StringBuilder

---------------------

String : بص يا سيدى الاسترينج العادى لو جيت تعمل كونكاتيونيشن او تعديل فهو بيفضل يخزن فى اماكن كتير جدا فى الميمورى يعنى لو هتعمل تعمل كونكاتيونيشن او تعديل بين مليون كلمه سترينج فكده كل تعديله او كونكتينسشن بيحجز مكان فى الميمورى و بالتالى هو حجز مليون مكان فى الميمورى ففرتك الميمور

StringBuilder : ده بقه لو عملت 100 مليون كونكاتيونيشن هو بيجمعهم كلمهم فى مكان واحد و يخزنهم فيه و بعد كده يشاور عليه يخده و بالتالى هو احسن مليون مره فى البيرفورمانس



اذاى اكتب الاسترينج و الاسترينج بيلدر بقه فى الكود ؟؟

using System;

using System.Text;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string FullName = "My ";

FullName += "Name is Mostafa ";

FullName += "Badawy";

Console.WriteLine(FullName);

StringBuilder MyStr = new StringBuilder("My ");

MyStr.Append("Name ").Append("is ").Append("Mahmoud ");

MyStr.Append("Badawy");

Console.WriteLine(MyStr);

}

}

}

MB

##################################################################

. Partial classes

---------------------

دى يا صديقى فايدتها ان يكون عندى مثلا كلاس واحده بس عاوز اقسم كتابته فى كذا فايل فبستخدم البرشيال كلاس دى و اكتر استخدامها فى الفريم وركس الى ذى ال web form هتلاقى دايما الكلاس ال cs بتعتها بارشيال كلاس لان الكلاس هنا بتنقسم الى 3 فايلات

فايل بنتكب فيه ال aspx و فايل بتكتب فيه Cs و فايل بنتكتب فيه designer.cs

كل دول بيبقوا اصلا كلاس واحده بس متقسمه على 3 فايلات و بما انها كلاس واحده و متقسمه فى فيلات عديده فده معناه ان الكلاس دى partial Class

مثال ده كلاس عادى فى فايل واحد اسمه Geeks.cs

public class Geeks

{

private string Authour\_name { get; set; }

private int Total\_articles { get; set; }

public Geeks(string a, int t)

{

this.Authour\_name = a;

this.Total\_articles = t;

}

public void Display()

{

Console.WriteLine("Author's name is : " + Authour\_name);

Console.WriteLine("Total number articles is : " + Total\_articles);

}

}

هقسمه بقه على فايلين Geeks1.cs & Geeks2.cs

فى الفايل الاول Geeks1.cs

public partial class Geeks

{

private string Authour\_name { get; set; }

private int Total\_articles { get; set; }

public Geeks(string a, int t)

{

this.Authour\_name = a;

this.Total\_articles = t;

}

}

فى الفايل التانى Geeks2.cs

public partial class Geeks

{

public void Display()

{

Console.WriteLine("Author's name is : " + Authour\_name);

Console.WriteLine("Total number articles is : " + Total\_articles);

}

}

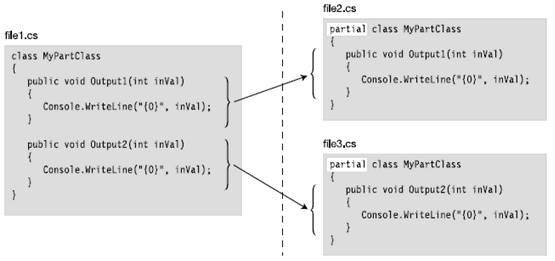
و انا بناديهم فى اى مكان بقه بنادى كلاس واحد الى هو Geeks ده و ده لان اثناء الكومبيليشن الكومبيلر بيجمع الفايلين و يخليهم كانهم فايل واحد بالظبط فيه كلاس واحده اسمها Geeks

طبعا

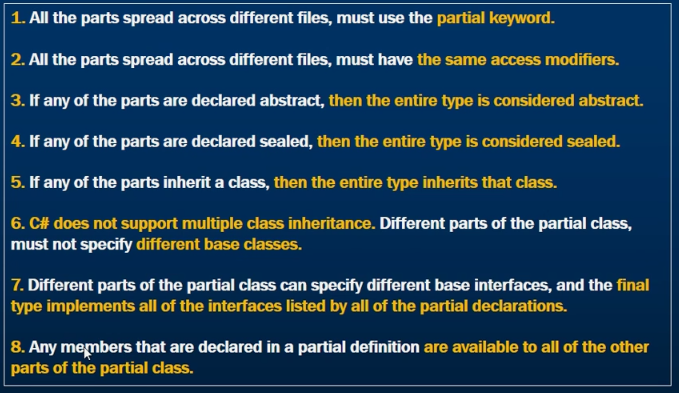
شروطه ان فى الفايلين لازم كل صفات الكلاس دى فى الفايلين تبقى متشابهه تماما يعنى لو الاكسيس موديفاير فى واحد بابليك لازن فى التانى تبقى بابليك و لو برايفيت تبقى فى التانى برايفت

و لو واحد منهم ورث فيعتبر التانى ورث و مينفعش يورث كلاس كمان عشان ساعتها هيبقى كان الكلاس بيورث اتنين كلاس و ده مينفعش فى السى شارب و كذلك لو فايل منهم اتكتب فيه ان الكلاس ابستراكت اصبح ابستركت فى الفايلين و كذلك لو بقى واحد فيهم سيلد

و دى صوره بتوضح كيفيه عمل البرشيال كلاس



و دى بتوضح الشروط



بس يا سيدى دى كل حاجه عنه و فى بقه بارشيال ميثودز و دى بنستخدمها لما نكون عاوزين نعمل كلاس بارشيال

فبنعمل فانش بارشيال فى الفايل الاول و بتكون عباره عن دكليريشن بس من غير اى امبليمنتيشن

و فى الفايل التانى بنكمل الفانكشن و بنكتب جواها امبليمنتيشن و لما بنيجى ننده عليها بننده عليها داخل فانكشن تانيه تكون public بما ان البرشيال ميقود اجبارى تبقى private

ليه بنعمل كده ؟؟ عزل او isolation للكود يا صديقى

مثال اهه

**Circle1.cs**

public partial class circle

{

// declaration of partial method

partial void area(int p);

}

**Circle2.cs**

public partial class circle

{

// declaration of partial method

public void newarea(int a)

{

area(a);

}

// This is the definition of

// partial method

partial void area(int r)

{

double A = 3.14 \* r \* r;

Console.WriteLine("Area is : {0}", A);

}

}

هنده بقه للفانكشن كده لما احتاج ارنها

class Program

{

public static void Main(string[] args)

{

circle x = new circle();

x.newarea(5);

}

}

MB

##################################################################

Constant

---------------------

Constant : متغير يتم تعريفه و اعطاءه intial Value مره واحده فقط و لا يمكن التعديل عليها

using System;

class Program

{

static void Main(string[] args)

{

const double Tax = 0.03; // const

// Tax = 0.50; Syntax Error

double Salary = 10\_000 - (10\_000 \* Tax); // variable

Console.WriteLine(Salary);

Salary = 5000 - (5000 \* Tax); // variable

Console.WriteLine(Salary);

Salary = 3500; // variable

Console.WriteLine(Salary);

}

}

من المثال ستفهم انه يمكن تغيير الفاريبول فى اى وقت لاى قيمه اخرى و لكن لا يمكن تغيير قيمه ال const تماما