C# Summery part-1 By Mahmoud Badawy

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

ملخص سى شارب الجزء الاول

الملخص يحتوى على

- Basics

- OOP

- 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

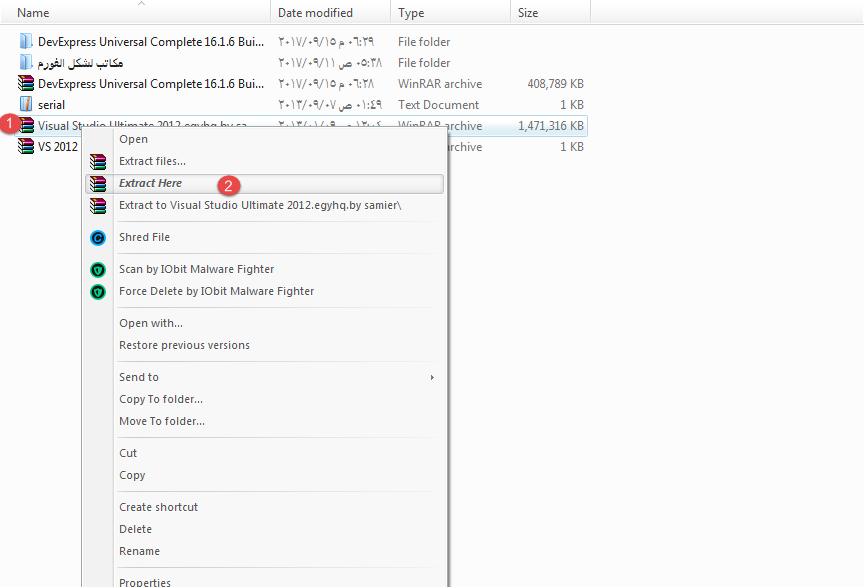
<http://www.kunal-chowdhury.com/2015/07/download-visualstudio-2015.html>

MB

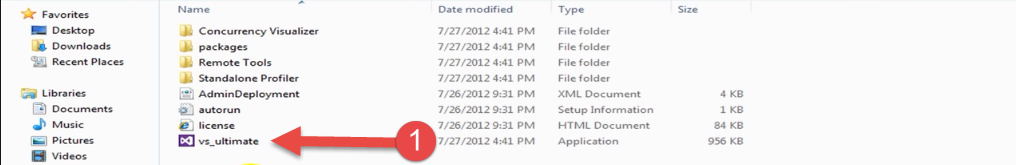
#######################################################################

اذاى بقه نسطبه

1. اول حاجه هتفك الضغط بتاعه هيظهرلك مجموعه ملفات

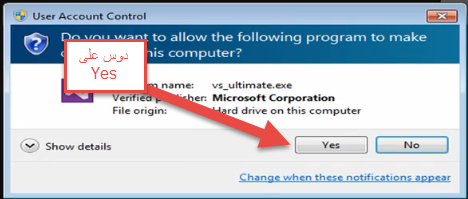


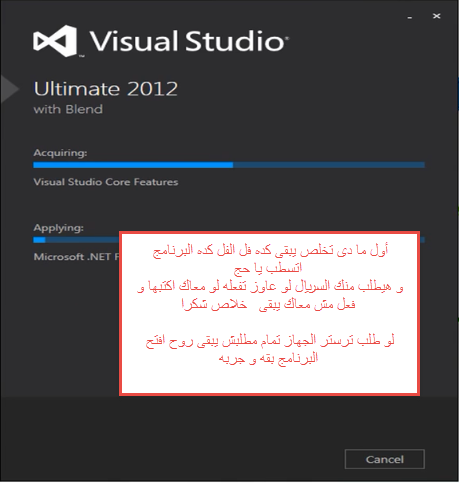
1. تانى حاجه نبدا نسطب بقه ندوس على تسطيب الابليكيشن بقه

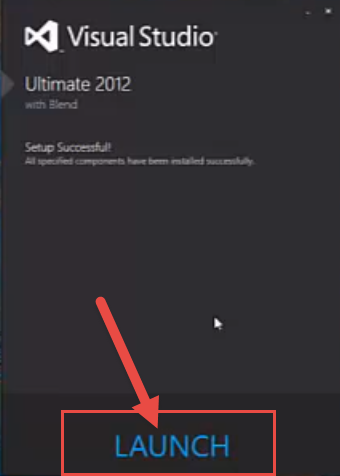


1. يبدأ من هنا التسطيب الفعلى و ده هنشرحه بالصور الى جايه بقه   
     
   



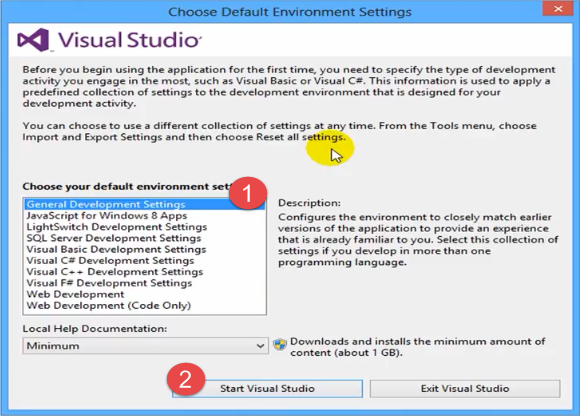


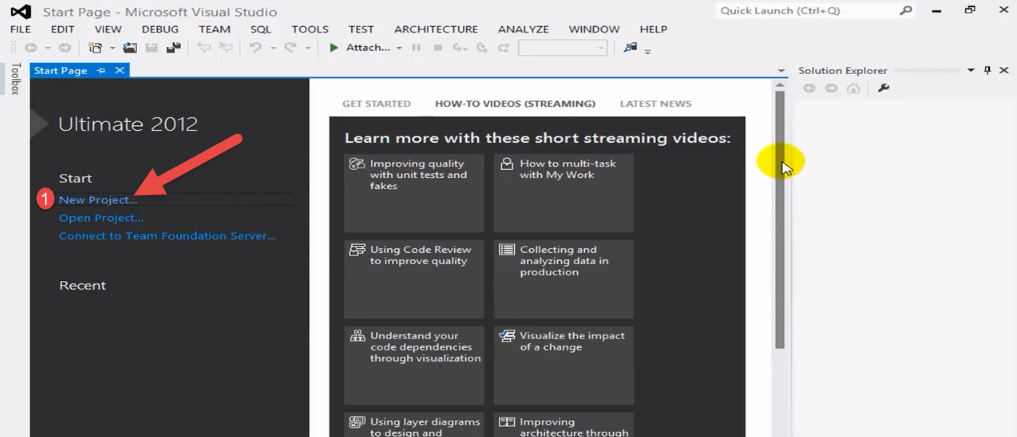


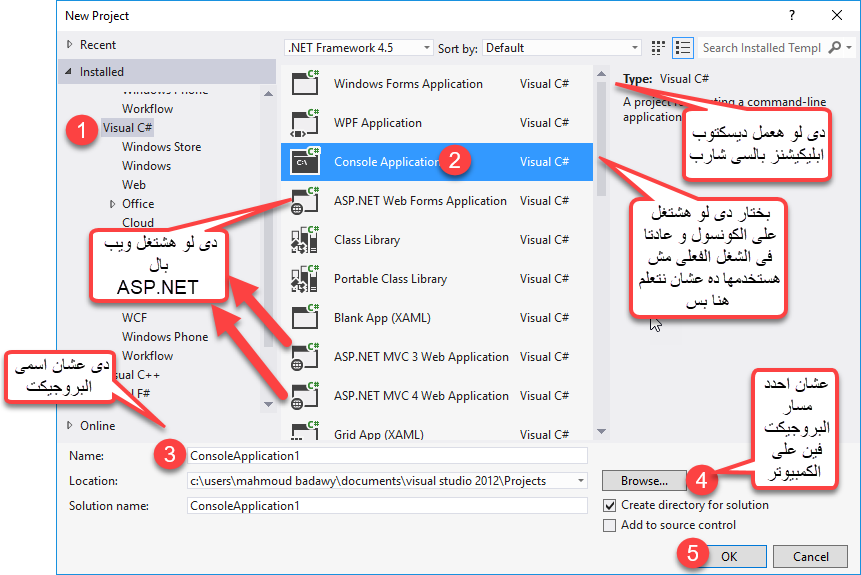


هيفتحلك قائمه كده تحط فيها السيريال نمبر  
 لو معاك لو مش معاك دوس   
Cancel   
 معاك حطها و دوس   
Next



روح افتح البرنامج بقه هيظهرلك اول مره اختيارات  






MB

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

البنيه الاساسيه لاى برنامج فى ال C#

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

Using System;  
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

using System;

namespace CSharpConsAppTest

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello Mahmoud Badawy");

}

}

}

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

* 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

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

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";

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);

}

}

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

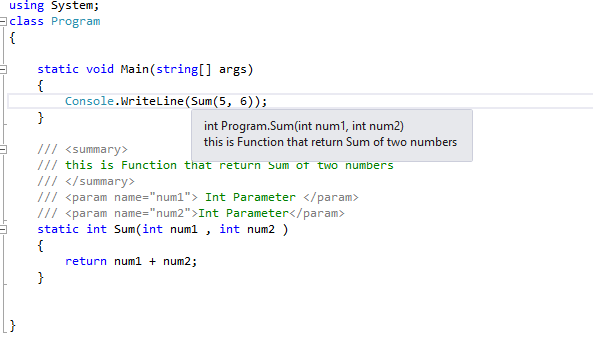
MB

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

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



MB

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

Coverting Type Methods

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

الاول عن طريق الكاستينج   
عاوز احول من فلوت لانتيجر مثلا

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");

}

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

MB

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

Array

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

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]);

}

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;

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]);

}

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

**طريقه كتابه اسهل فى الاراى**

static void Main(string[] args)

{

//type[] name = {Values here}

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

Console.WriteLine(arr[2]);

}

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);

}

}

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)";

}

}

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

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

MB

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

While Loop

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

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

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

static void Main(string[] args)

{

int Num = 1 ;

while (Num<=15)

{

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

Num++;

}

}

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

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

while (true)

{

Console.WriteLine("Hello Mahmoud badawy");

}

}

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

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

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));

}

}

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 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

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

NameSpace

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

بص يا سيدى دى يعتبر معموله للتنظيم عشان احط فيها كلاسات و انموريشنز و ديليجيتس براحتى و تبقى تابعه للنيم سبيس الفلانى و يعتبر ال dll بيتعمل باسمها برضوا و بيبقى لامم كميه كبيره من الكلاسات و الانترفيسيس و اى حاجه ينفع تتحط جوه نيم سبيس عمتا

لاحظ ان ينفع تحط نيم سبيس جوه نيم سبيس و عشان انادى اى حاجه من جوه نيم سبيس طبعا بعتبرها مكتبه هجيب منها حاجه و بالتالى بقوله using و اكتبله اسم النيم سبيس و افضل اتتبع الحاجه لحد ما اجيبها بقه لان ممكن يبقى الحاجه جوه كلاس و الكلاس جوه نيم سبيس و النيم سبيس جوه نيم سبيس و كمان ممكن يبقى في نيم سبيسيس باسامى متشابهه فساعتها بعمل Aliases عشان افرق بينهم و هيصه كده ربنا الى عالم بيها و عمتا عشان تفهم ادينى عملت مثال

using PATA = ProgectA.TeamA; // namespace have Alias PATA

using PATB = ProgectA.TeamB; // namespace have Alias PATB

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

ProgectA.TeamA.MyCls.Myfunc();

ProgectA.TeamB.MyCls.Myfunc();

PATA.MyCls.Myfunc();

PATB.MyCls.Myfunc();

}

}

}

namespace ProgectA

{

namespace TeamA

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamA > MyCls ");

}

}

}

}

namespace ProgectA

{

namespace TeamB

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamB > MyCls ");

}

}

}

}

MB

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

NameSpace

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

يمكن كتابه الكود الى فات بطريقتين الطريقه الأولى

النيم سبيس أحيانا لما بيبقى تحت السوليوشن على طول بيبقى عباره عن بروجيكت الى جوه السوليوشن و طبعا انت عارف ان السوليوشن ممكن يشيل 5000000 بروجيكت عادى

using PATA = ProgectA.TeamA; // namespace have Alias PATA

using PATB = ProgectA.TeamB; // namespace have Alias PATB

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

ProgectA.TeamA.MyCls.Myfunc();

ProgectA.TeamB.MyCls.Myfunc();

PATA.MyCls.Myfunc();

PATB.MyCls.Myfunc();

}

}

}

namespace ProgectA.TeamA

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamA > MyCls ");

}

}

}

namespace ProgectA.TeamB

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamB > MyCls ");

}

}

}

الطريقه التانيه ان النيم سبيس الى متشابه في الاسم اجمع فيه كل النيم سبيسيس و الكلاسات الخاصه بيه بدل ما انطور اسمه في كل حته على الفاضى و خلاص

using PATA = ProgectA.TeamA; // namespace have Alias PATA

using PATB = ProgectA.TeamB; // namespace have Alias PATB

namespace LearnProg

{

class Program

{

static void Main(string[] args)

{

ProgectA.TeamA.MyCls.Myfunc();

ProgectA.TeamB.MyCls.Myfunc();

PATA.MyCls.Myfunc();

PATB.MyCls.Myfunc();

}

}

}

namespace ProgectA

{

namespace TeamA

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamA > MyCls ");

}

}

}

namespace TeamB

{

class MyCls

{

public static void Myfunc()

{

Console.WriteLine("Hello From ProgectA > TeamB > MyCls ");

}

}

}

}

MB

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

Classes

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

بص يا سيدى دى يعتبر اهم حاجه في ال OOP و دى يعتبر كده الاستراكتشار بتاع اى حاجه هتعملها جديده كان معاك اوضه و بتفسمها براحتك او كان هي دى اللوحه الى بتحط فيها تصميمك و تمشى الناس عليه من الاخر كده الكلاسات هي الى بتنظم المشاريع كلها

كل كلاس بيحتوى على مجموعه بروبرتيس الى هي عباره عن مدخلات و مخرجات و كذلك مجموعه من الميثودز و دى الى هي وظايف الكلاس ده بيعمل ايه و كذلك يحتوى على كونستراكتور و ده ياعم الحاجه الى بتربط اى كلاس باى كلاس اخر و بتنقل بيه الداتا من كلاس لكلاس اخر و ده دايما ملازم الاوبجيكت الى بتعمله من الكلاس و يمكن عمل اكتر من كونستراكتور للكلاس الواحد عادى المهم تغيير عدد البراميتر الى دخلاله عشان نعرف نفرق بين كل واحد (اوفرلودينج يعنى ) و وانت بتعمل الاوجيكت بتختار بقه تعمل الاوبجيكت بانهى كونستراكتور ساعتها

و فى الديستراكتور و ده بقه لما يكون الاوبجيكتات الى انت عاملها في الكلاس كلها خلصت و خلاص هيطلع من الكلاس و مبقاش ليه شغل فيه ساعتها بيشوف انت انشات كام اوبجيكت من الكلاس و على أساسه بيعمل ديستراكت لكل الاوبجيكتات دى (لاحظ ان الكونستراكتورو الديستراكتور بيخدوا نفس اسم الكلاس بس الديستراطتور بيبقى قبل اسمه ~)

namespace ConsoleApplication1

{

public class Customer

{

// Class Properties

public string \_FirstName;

public string \_LastName;

// Class Constracor

public Customer(string FirstName, string LastName)

{

this.\_FirstName = FirstName;

this.\_LastName = LastName;

}

// Class Methods

public void FullName()

{

Console.WriteLine("My name is : {0} {1}", this.\_FirstName, this.\_LastName);

}

~Customer()

{

Console.WriteLine("Destracor Works Successfully");

}

}

class Program

{

public static void Main(string[] args)

{

// Class Cusomer Obj (instance)

Customer Obj1 = new Customer("Mahmoud", "Badawy");

Customer Obj2 = new Customer("Mostafa", "Badawy");

// Call Method from Cusomer Class By Obj (instance)

Obj1.FullName();

Obj1.FullName();

Obj1.FullName();

Obj2.FullName();

}

}

}

MB

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

Static and instance

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

الفرق بين ستاتيك و انستانس في الكلاسات   
الانتستانس كلاس هو انك تنادى الكلاس بأوبجيكت او انستانس منه و على أساس الاوبجيكت بتقدر تستدعى بيه الى جوه الكلاس   
الاستاتيك كلاس بقه انت مش بتنشأ من الكلاس اى اوبجيكت خالص لا بتناديه عن طريق انك تكتب اسمه دوت اى حاجه جواه بقه

ملاحظات هامه

الانستانس كلاس تقدر تعمل منه ألاف الاوبجيكتات و بالتالى كل مره بتعمل فيه اوبجيكت و تنادى بيه اى حاجه بتستخدم مساحه من الميمورى للاوبجيكت ده و كل ابويجيكت ذي ما انت عارف بيشور على مكان في الميمورى و بياخد القيمه عن طريق الادرس و كده فلفه سوده يعنى ههههههههههه

الاستاتيك كلاس بقه هو بيتم استخدام هو بس و بالتالى مهما استخدمت فيه اى حاجه جواه فهو بيتخزن في الميمورى مره واحده بس و اى تغيير بيبقى في المكان ده بس في الميمومورى

اذا لما يكون حاجه انت متأكد انها مش هتحتاج تغيرها كل شويه فريح دماغك و اعملها ستاتيك احسن

طبعا ينفع تعمل الكلاس ستاتيتك و ينفع تعمل الفيلدس او المدخلات او الفاريبولز او البروبرتيس او الفانكشنز او الميسودز الى جواه من نوع ستاتيك

ستتيك معاناه شيرابول يعنى من الاخر كده هو موجود في كل الاوبجيكتات بنفس العنوان في الميمورى و بالتالى مش محتاج تعرفه جوه اى اوبجيكت لانك معلمه بالكلاس يا صديقى

أول ملحوظه الكلاس الى فات في الشرح الى فات ده كل شغل انستانس كلاس و ميثودز و بروبرتيس

تعالى بقه اوريك اذاى تعمل الكلاس ستاتيك و اذاى تعمل الكلاس عادى و الفانكشنز او البروبرتيس الى جواه ستاتيك

ده كود فيه كلاس ستاتيك و من الواضح ان اى كلاس ستاتيك لا يقبل بداخله الا بروبرتيس او ميثودز و تكون من نوع ستاتيك هي كمان

اما الانستانس كلاس العادى بيقبل ان جواه يتحط ستاتيك ميثود او ستاتيك بروبرتيس عادى

الاستاتيك كونستراكتور لازم يبقى من غير اى اكسيس موديفاير لان مينفعش يتقرأ بره الكلاس بتاعه لان الاستاتيك أصلا المفروض انه مش بيتنادى باى اوبجيكت أساسا فسواء عملته جوه ستاتيك كلاس او انستانس كلاس يبقى متدلوش اى اكسيس موديفاير بقه

أدى الكود بص عليه بقه و هتفهم الفرق بين الانستانس و الاستاتيك كلاس كويس ان شاء الله

لاحظ ان اول حاجه الكومبايلر بيقراها في الكلاس هو الاستاتيك كونستراكتور عشان يروح يحط قيم الفيلدز على طول هناك الأول و يملاها و بعد كده يروح يقرأ اى حاجه بقه ستاتيك ثم يقرأ الحاجات العاديه بقه مهما كان ترتيب سطور الكود يا اسطى

namespace ConsoleApplication1

{

// instance Class

public class Customer

{

// instance Class Properties

public string \_FirstName;

public string \_LastName;

// static Class Field (Propert)

static Double PI ;

// Class Constracor

public Customer(string FirstName, string LastName)

{

this.\_FirstName = FirstName;

this.\_LastName = LastName;

}

static Customer()

{

Customer.PI = 3.41;

Console.WriteLine("Static Customer Constracor");

}

// Class Methods

public void FullName()

{

Console.WriteLine("My name is : {0} {1}", this.\_FirstName, this.\_LastName);

// Calling static Method

Customer.GetPI();

}

// static Class Method

static void GetPI()

{

// Calling static Field (Propert)

Console.WriteLine("PI = {0}",Customer.PI);

}

~Customer()

{

Console.WriteLine("Destracor Works Successfully");

}

}

public static class PersonalInf

{

// Static Class Can not Accept any prop or Methods if it is not static

public static string Name;

// Static Constracor

static PersonalInf()

{

Console.WriteLine("Static PersonalInf Constracor");

}

public static void MyName() {

Console.WriteLine("My name is : {0} ", PersonalInf.Name);

}

}

class Program

{

public static void Main(string[] args)

{

// Class Cusomer Obj (instance)

Customer Obj1 = new Customer("Mahmoud", "Badawy");

Customer Obj2 = new Customer("Mostafa", "Badawy");

// Call Method from Cusomer Class By Obj (instance)

Obj1.FullName();

Obj1.FullName();

Obj1.FullName();

Obj2.FullName();

// Call Methods and Properties from Static Class PersonalInf

PersonalInf.Name = "Mesho MeBO";

PersonalInf.MyName();

}

}

}

MB

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

Inheritance

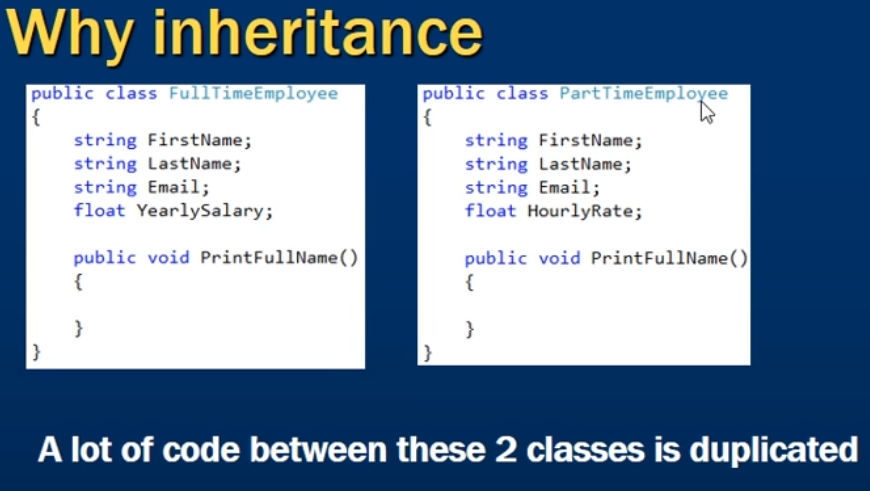
----------------------------

الوراثه و دى كونسيت مهم جدا في الاو او بى و هو ان اذاى كلاس يورث من كلاس كل خواصه من اول البروبرتيس للميثودز و الحاجات الى فيه عمتا كان ابن بيورث ابوه فبياخد منه كل صفاته

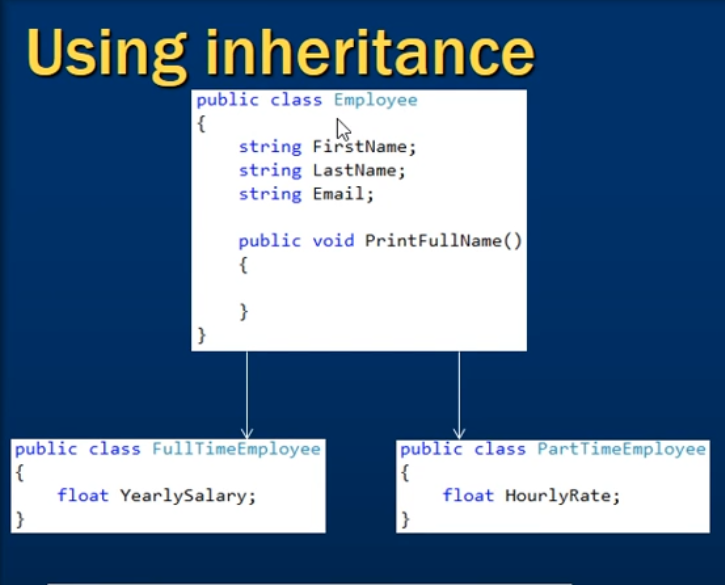
بفرض ان كلاس PersonINFO فيه بروبرتى الاسم و النوع و العمر و الشغل و فيه ميثود بيحسب المرتب و بيحسب السن باليوم مثلا

و جبت كلاس تانى اسمه Person و قولتله اورث يا با من الكلاس PersonINFO ساعتها هيورث منه الحاجات الى فيه و هي بروبرتى الاسم و النوع و العمر و الشغل و فيه ميثود بيحسب المرتب و بيحسب السن باليوم كانهم عنده بالظبط بس هو مش كاتبهم هو ورثهم من صاحبنا التانى الى هو اب يعتبر و الوريث هو الكلاس الابن

دى صور بتوضح اهميه الوراثه



بدل العك الى فوق ده المفروض نعمله كده شوف توفير الكود عامل اذاى



تعالى اوريك مثال بالكود بقه و عشان تعرف الوراقه بتتكتب اذاى بالكود كمان

namespace ConsoleApplication1

{

public class EmployeeParent\_CLS

{

public string FirstName;

public string LastName;

public void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName , this.LastName);

}

}

// HourlyEmp inherits from EmployeeParent\_CLS

public class HourlyEmp : EmployeeParent\_CLS

{

public float HourlyEmpSalary = 30;

public void Salary()

{

Console.WriteLine("EMP Salary is per month : {0} ", this.HourlyEmpSalary \* 8 \* 30);

}

}

// YearlyEmp inherits from EmployeeParent\_CLS

public class YearlyEmp : EmployeeParent\_CLS

{

public float YearlyEmpSalary = 60000;

public void Salary()

{

Console.WriteLine("EMP Salary is per month : {0} ", this.YearlyEmpSalary / 12);

}

}

class Program

{

public static void Main(string[] args)

{

HourlyEmp objHourlyEmp = new HourlyEmp();

objHourlyEmp.FirstName = "Mahmoud";

objHourlyEmp.LastName = "Badawy";

objHourlyEmp.printFullName();

objHourlyEmp.Salary();

YearlyEmp objYearlyEmp = new YearlyEmp();

objYearlyEmp.FirstName = "Mostafa";

objYearlyEmp.LastName = "Badawy";

objYearlyEmp.printFullName();

objYearlyEmp.Salary();

}

}

}

لاحظ ان الكلاسات الى ورست ورست كل حاجه من الاب طالما الحاجه الى في الاب بابليك او بروتيكتيد انما لو برايفيت الكلاسات الى بتورث مش هتورثها أصلا + الابن يقدر يورث الحاجه و يعدل عليها طالما هو بيتعامل مع اوبجيكت بتاعه من نفسه يعنى لو الاوبجيكت ده بتاع الابن الى بيورث فليه حريه انه يورث الحاجه و يعدل عليها عادى يعنى ما هي بتعته خلاص + لو لقيت فانكشن جواها مختلف عن الى بيورث منه اشطه اعمل اوفر رايتينج جوه كلاس الابن بحيث الابن ينادى على حاجته و ملوش دعوه بالى وارثه + الوراثه انك بتكتب الكلاس الابن ثم نقتطين فوق بعض ثم كلاس الاب + في السى شارب الكلاس بينفع يورث من كلاس واحد انما بينفعش يورث من كذا كلاس في سطر واحد و انما ينفع الابن يورث من الاب ثم ابن صغنن حفيد يعنى يورث من الابن الوسط و بكده يبقى الحفيد ورث صفات ابوه و جده

MB

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

Member or Method Hiding

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

دلوقتى لما يكون في كلاس بيورث من كلاس تانى و الكلاس الاب فيه فانكشن و الابن فيه نفس الفانكشن بالحته الى بيحصل بقه ان الاوبجيكت بتاع الابن بيجى ياخد الفانكشن دى فبيستقرب فبيروح واخد بتاعته و مبيورثسش بتاعه ابوه و ده بقه بيتسمى في البرمجه ان الفانكشن بتاعت الابن عملت اوفارايت او هايدينج للفانكشن الى وارثها من الاب و طبعا متنساش كلمه new في فانكشن الابن

namespace ConsoleApplication1

{

Public class EmployeeParent\_CLS

{

public string FirstName;

public string LastName;

public void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName , this.LastName + " - From EmployeeParent\_CLS ");

}

}

public class HourlyEmp : EmployeeParent\_CLS

{

// Method Hiding

public new void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName, this.LastName + " - From HourlyEmp ");

}

}

class Program

{

public static void Main(string[] args)

{

HourlyEmp HourlyEmpObj = new HourlyEmp();

HourlyEmpObj.FirstName = "Mahmoud";

HourlyEmpObj.LastName = "Badawy";

HourlyEmpObj.printFullName();

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

HourlyEmp HourlyEmpObj1 = new HourlyEmp();

HourlyEmpObj1.FirstName = "Mahmoud";

HourlyEmpObj1.LastName = "Badawy";

// Convert Hourly Obj to EmployeeParent\_CLS to Cancel OverWritting

((EmployeeParent\_CLS)HourlyEmpObj1).printFullName();

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

// EmployeeParent\_CLS Point to HourlyEmp

EmployeeParent\_CLS HourlyEmpObj2 = new HourlyEmp();

HourlyEmpObj2.FirstName = "Mahmoud";

HourlyEmpObj2.LastName = "Badawy";

HourlyEmpObj2.printFullName();

}

}

}

طيب لو عندى نفس الاتنين فانكشن و مش عاوز الابن يجيب الفانكشن بتاعته و عاوزه يجيب فانكشن الاب بتاعه اعمل ايه ؟؟

* الى بيحصل ان بعمل اثناء و انا بعمل كول للفانكشن بروح عامل كاستينج للاوبجيكت من نوعه لنوع ابوه عشان يشغل فانكشن ابوه يا حج
* الحل التانى انى اعمل اوبجيكت من الاب عامل ريفرنس او بوينتينج للكلاس الابن و ده هيتيح انى اتعامل مع حاجات الاب و الابن و الاب يبقوا مشتركين في نفس مكان التخزين في الميمورى عادى لانه اوبجيكت الاب بيبقى عامل بوينتج لحاجات الابن و ده مفهوم البوليمرفيزم بتبلمر الحاجه بحيث تزاول الهاكر ميبقاش عارف انهى اوبجيكت بيبص على انهى حاجه هههههههههههههههههههه :D
* ده من الاخر بتعمل اوبجيكت من الاب و بتخزن في أماكن الابن و بتبص على الى انت خزنته بقه من اوبجيت الاب عادى مع انه متخزن في ميمورى الابن ههههههههه حد فاهم حاجه يا جدعان

مثال اخر للتوضيح اكثر عن جزء ال **Polyemarphism**

في الكود ده هتلاحظ ان فكره الهايدينج ملغيه تماما و كل الاوبجيكت بتقرأ من الاب و سايبه الأبناء عشان انا عامل بوليمرفيزم

namespace ConsoleApplication1

{

public class EmployeeParent\_CLS

{

public string FirstName;

public string LastName;

public void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName , this.LastName + " - From EmployeeParent\_CLS ");

}

}

public class HourlyEmp : EmployeeParent\_CLS

{

public new void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName, this.LastName + " - From HourlyEmp ");

}

}

public class yearlyEmp : EmployeeParent\_CLS

{

public new void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName, this.LastName + " - From yearlyEmp ");

}

}

class Program

{

public static void Main(string[] args)

{

EmployeeParent\_CLS [] Obj = new EmployeeParent\_CLS[3];

Obj[0] = new EmployeeParent\_CLS();

// start Polyimervism

Obj[1] = new HourlyEmp();

Obj[2] = new yearlyEmp();

// End Polyimervism

foreach (EmployeeParent\_CLS I in Obj)

{

I.FirstName = "Mahmoud";

I.LastName = "Badawy";

I.printFullName();

}

}

}

}

طيب في الحاله دى لو عاوز اعمل Override هعمل ايه يا اسطى بقه ساعتها ؟؟

هكتب قبل الحاجه الى عاوز اعملها اوفررايد في الاب كلمه virtual

و قبلها في الابن كلمه Override كده

**و هو ده مفهوم البوليمرفيزم الصح انى استدعى الميثودز بتاعت الابن عن طريق اوبجيكت الاب و ده بيتم كما بالكود الاتى**

MB

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

Member or Method Override

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

namespace ConsoleApplication1

{

public class EmployeeParent\_CLS

{

public string FirstName;

public string LastName;

public virtual void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName , this.LastName + " - From EmployeeParent\_CLS ");

}

}

public class HourlyEmp : EmployeeParent\_CLS

{

public override void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName, this.LastName + " - From HourlyEmp ");

}

}

public class yearlyEmp : EmployeeParent\_CLS

{

public override void printFullName()

{

Console.WriteLine("EMP name is : {0} {1}", this.FirstName, this.LastName + " - From yearlyEmp ");

}

}

class Program

{

public static void Main(string[] args)

{

EmployeeParent\_CLS [] Obj = new EmployeeParent\_CLS[3];

Obj[0] = new EmployeeParent\_CLS();

// start Polyimervism

Obj[1] = new HourlyEmp();

Obj[2] = new yearlyEmp();

// End Polyimervism

foreach (EmployeeParent\_CLS I in Obj)

{

I.FirstName = "Mahmoud";

I.LastName = "Badawy";

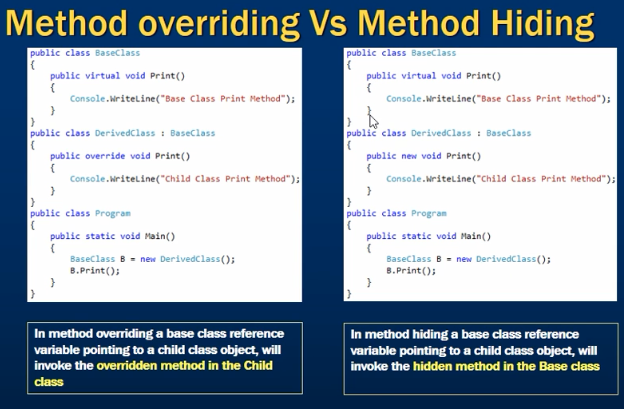
I.printFullName();

}

}

}

}



الميثود هايدينج لو انت عامل بوليمرفيزم او كاسيتنج للاوبجيكت مش هينادى للميثود الى جوه الابن لا هينادى للى جوه الاب و يبقى كده فكره الهايدينج ادمرت ذى الصوره الى ناحيه اليمين   
بينما فى الصوره الى فى ناحيه الشمال بيحصل فيها اوفر رايد مهما كان عامل بقه كاستينج عامل بوليمرفيزم عامل بتنجان هو الميثود الى جوه الابن هى الى بيتم منادتها

خد بالك كويس ان على طول الاوبجيكت بتاع الاب هو الى بيعمل بوينتينج للكلاس بتاع الابناء مش العكس

MB

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

Method Overloading

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

هى انك تعمل نفس الميثود بنفس الاسم ولكن مختلفه اما فى عدد البراميترز او نوع الداتا تايب انتيجر ولا فلوت بتاعت البراميتر او نوع الفاريبول نفسه انبوت ولا اوت بوت ولا فاليو ولا ريفرنس   
مثال

namespace ConsoleApplication1

{

class Program

{

public void MyFunc()

{

Console.WriteLine("Hello From My Function With No Param");

}

public void MyFunc(int Num)

{

Console.WriteLine("Hello From My Function With one Integar Param : " + Num);

}

public void MyFunc(string Num)

{

Console.WriteLine("Hello From My Function With one string Param : " + Num);

}

public void MyFunc(int Num , int Num2)

{

Console.WriteLine("Hello From My Function With Two Integar Input Param : " + Num + " & " + Num2);

}

public void MyFunc(int Num, out int Num2)

{

Num2 = Num;

Console.WriteLine("Hello From My Function With Two Integar input and output Param output Num2 = " + Num2);

}

static void Main(string[] args)

{

Program Obj = new Program();

int x;

Obj.MyFunc();

Obj.MyFunc(2);

Obj.MyFunc("3");

Obj.MyFunc(2,3);

Obj.MyFunc(2,out x);

}

}

}

MB

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

Why Properties ??

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

بص يا صاحبى فى الغالب انت لما تيجى تعمل كود من غير بروبرتيس بيبقى الكود سمك لبن تمر هندى الى هو ممكن تدخل درجات طالب بالسالب مع ان المفروض ان مفيش درجات بالسالب اصلا و ممكن متدخلش فيها حاجه اصلا و تسيبها ب Null و برضوا الكلام ده غلط ما هو مستحيل واحد هيبقى درجته مفيش هههههههه و فى اوقات بتبقى عاوز تعمل الحاجه مينفعش حد يغيرها يعنى و ليكن مثلا المفروض ان درجه النجاح 50 و انت عاوز ان مينفعش حد يعدلها فالمفروض انك بتعملها Read Only و طبعا ده مش متوفر فى الفاريبولز العاديه   
و عشان كده اتعمل ال Properties   
لأن هى ذى ما تقول كده بتعمل قوانين و فالديشنز للفاريبولز   
يعنى عن طريق البروبرتيس تقدر تخلى الفاريبول لا تقبل الا الموجب بس و مينفعش تبقى null و كمان تخليها Read only او حتى تخليها Write Only من الاخر كده بتمسك الفاريبول تظبه على كيفك

ده كود بيبين مساوئ عدم استحدام البروبرتيس

namespace ConsoleApplication1

{

class Program

{

public int ID;

public int PassMark;

public string StudentName;

static void Main(string[] args)

{

Program Obj = new Program();

Obj.ID = -20;

Obj.PassMark = -50;

Obj.StudentName = null;

Console.WriteLine("ID : {0} - Name : {1} - PassMark: {2}", Obj.ID, Obj.StudentName, Obj.PassMark);

}

}

}

تعالى نظبط الاى دى مثلا بفانكشنز عاديين يشبهوا السيتر و الجيتر

namespace ConsoleApplication1

{

class Program

{

public int ID;

public string StudentName;

public int PassMark = 50;

public void SetID(int id)

{

if (id < 0)

{

throw new Exception("ID should be greater zero");

}

this.ID = id;

}

public int GetID()

{

return this.ID;

}

public void SetSN(string SN)

{

if (string.IsNullOrWhiteSpace(SN))

{

throw new Exception("Student name should not be null");

}

this.StudentName = SN;

}

public string GetSN()

{

return this.StudentName;

}

// Make PassMark Read Only

public int GetPassMark()

{

return this.PassMark;

}

static void Main(string[] args)

{

Program Obj = new Program();

// -101 for ID or null For SN will Make Exception So I Comment them

//Obj.SetID(-101);

//Obj.SetSN(null);

Obj.SetID(101);

Obj.SetSN("Mahmoud Badawy");

Console.WriteLine("ID : {0} - Name : {1} - PassMark : {2} " , Obj.GetID(), Obj.GetSN(),Obj.GetPassMark());

}

}

}

فى الحاله ان الاى دى بالسالب او اسم الطالب فاضى الكود هيضرب لان انا عامل اكسيبشن فى الحالات الى انا عاملها الى هى الاى دى بالسالب او اسم الطالب بال Null و طبعا الباس مارك كده خلاص بقى ريد اونلى

طبعا الى فات ده بلح الى فات ده مجرد فانكشنز عاديه بستخدمها لو اللغه مفيهاش البروبرتيس فبطر استخدم الفانكشن سيت و جيت

انما لغه ذى السى شارب فيها اى ابن كلب يبقى لازم نشرح البروبرتيس بقه و نشوف بقه حته ال Encapsualtion

و طريقه عمل البروبرتى بسرعه حاجه من اتنين يا اما بال insert snip ctrl + k , Ctrl + C  
او اعمل الفاريبول برايفيت و اروح اقف عليه و ادوس Ctrl + . هيعمل سيت و جيت جوه بروبرتى بابليك هروح حاطط اللوجيك جوه السيت و اخده بقه بالجيت ذى ما هتشوف فى الكود ده بالملى

namespace ConsoleApplication1

{

class Program

{

private int \_ID;

private string \_StudentName;

private int \_PassMark = 50;

public int ID

{

// Read And Write

set

{

if (value < 0)

{

throw new Exception("ID should be greater zero");

}

this.\_ID = value;

}

get

{

return this.\_ID;

}

}

public string StudentName

{

// Write Only

set

{

if (string.IsNullOrWhiteSpace(value))

{

throw new Exception("Student name should not be null");

}

\_StudentName = value;

}

}

public int PassMark

{

// Read Only

get

{

return \_PassMark;

}

}

static void Main(string[] args)

{

Program Obj = new Program();

Obj.ID = 101;

Obj.StudentName = "Mahmoud Badawy"; // Write only so i can not read it in console

// I can not write in PassMark So it will Shown only in console

Console.WriteLine("ID : {0} - PassMark : {1} " , Obj.ID, Obj.PassMark);

}

}

}

ملحوظه مهمه جدا فى شغلك لازم اى فاريبول هتعمله تعمله بروبرتى عشان ده هيخلى شغلك معموله انكبسوليشن و مفيهوش ثغرات و هيبقى شغلك نضيف حتى لو البروبتى مفيهاش اى لوجيك استخدمها رخامه

MB

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

Structs

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

الاستراكت فى طريقه الكتابه تعتبر متطابقه للكلاس و لكن هناك اختلافات بين الاستراكت و الكلاسيس هنقول عليها بس الاول تعالى اوريك اد ايه الاستراكت شبه الكلاس فى الكتابه يعنى مثلا هتلاقى فيها فيلدس و بروبرتيس و ميثودز و و كونستراكتور و كان هتلاقى ان بيتعمل منها اوبجيكت عادى و فل الفل

الكود

namespace ConsoleApplication1

{

struct Person

{

// fields and Properties

private string name;

private int age;

public string Name

{

get{return name;}

set{name = value;}

}

public int Age

{

get{return age;}

set{age = value;}

}

//Constructor

public Person(String \_Name , int \_Age)

{

this.name= \_Name;

this.age = \_Age;

}

//Methods

public void MyFunc()

{

Console.WriteLine("My Name is : {0} , My Age is : {1} ", this.name, this.age);

}

}

class Program

{

static void Main(string[] args)

{

Person Obj1 = new Person("Mahmoud Badawy", 28);

Obj1.MyFunc();

Person Obj2 = new Person();

Obj2.Name = "Mosatafa Badawy";

Obj2.Age = 2;

Obj2.MyFunc();

}

}

}

MB

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

Struct Vs Class

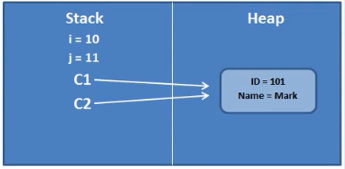
--------------------------------

هناك اختلافات كثيره بين الاستراكت و الكلاس و اهمهم ان ان مفيش ديستراكتور فى الاستراكت و كمان لو حبيت اعمل فيه كونستراكتور لازم الكونستراكتور يبقى بيخشله بارميتر و الا مش هيرضى يتعمل على عكس الكلاس بيتعمل عادى

الاستراكتشار لا يرث الا من الانترفيس بينما الكلاس يرث من الكلاسيز و الانترفيسيز عادى

الاستراكتشار value Type بينما الكلاس Reference type  
و بالتالى الاستراكت بيتخدن فى الاستاك بينما الكلاس بتخزن الاوبجيكت فى الاستاك و بتشاور على القيم بتعتهم فى ال heap و دول اماكن فى الميمورى

و بالتالى الاستراكت بيتخزن فى الميمورى بشكل مباشر فى نفس المكان الى هو اتعرف فيه بينما الكلاس بيخزن الداتا بتعته فى الميمورى جوه الاوبجيكت و بالتالى بيبقى متخزن فى مكان يقدر هو يشاور عليه يجيبه بمعنى اصح و انت بتعرف الاوبجيكت كده التعريفه دى فى الاستاك و لما تيجى تدى الاوبجيكت قيمه بيخدها فى مكان تانى فى الهيب بس بيبقى فى حلقه وصل يعنى الاوبجيكت فى الاستاك بيشاوى على مكان فى الهيب



و بالتالى الفرق بين الفاليو و الريفرنس فاريبول هو ان الفاليو مجرد ما تطلع من الاسكوب بتاعه بيتمسح من الميمورى   
بينما الريفرنس بقه الريفرنس فاريبول نفسه بيتمسح من الاستاك بمجرد الخروج من الاسكوب بينما الحاجات الى جوه الهيب الى هى الا بجيكت و الحاجات الى جواه بتتمسح لما الجاربيدج سيشن يخلص

الفرق بقه بين الفاليو و الريفرنس فاريبول فى مثال بالكود

namespace ConsoleApplication1

{

class Numbers\_CLS

{

public int Num\_1 , Num\_2;

}

class Program

{

static void Main(string[] args)

{

int num\_1, num\_2;

num\_1 = 10;

num\_2 = num\_1;

num\_2 = num\_2 + 1;

Console.WriteLine("Value type : num1 = {0} , num2 = {1} ",num\_1 , num\_2);

Numbers\_CLS Obj1 = new Numbers\_CLS();

Obj1.Num\_1 = 10;

Obj1.Num\_2 = 11;

//Refrence Values Obj 2 Refer to Obj 1

Numbers\_CLS Obj2 = Obj1;

Obj2.Num\_1 = 100;

Obj2.Num\_2 = 110;

Console.WriteLine("Refrence type : Obj1.Num\_1 = {0} , Obj1.Num\_1 = {1} ", Obj1.Num\_1 , Obj1.Num\_2);

Console.WriteLine("Refrence type : Obj2.Num\_1 = {0} , Obj2.Num\_1 = {1} ", Obj2.Num\_1, Obj2.Num\_2);

}

}

}

لاحظ ان الفاليو تايب اى تغيير فى اى حته مش بيأثر على التانيه مع انى واخد من نامبر وان حاطط فى نامبر تو

اما فى الريفرنس فاريبول انا شاورت على الاوبجيت 1 و انا بعرف اوبجيكت 2 فبقى اى حاجه بتحصل فى اوبجيكت 1 او اتنين يروحوا الاتنين مأثرين على بعض و ده الى بيسموه pointing و الريفرنس فاريبولز

MB

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

Sealed Class

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

ده بيتعمل لما تكون مش عاوز حد يورث من الكلاس ده خالص

اذا نقدر نعتبر الاستراكتشار بقه انه سيلد كلاس ههههههههههه لانه بورسش حد حاجه

MB

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

interfaces

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

الانترفيسيز دى من الاخر معموله عشان السينيور يحط فيها الديفينيشنز و الرولز الى الجينيور يمشى عليها و اجبره انه يحطها ذى ما انا حاططها من غير ميجود و لذلك هى يمكن وضع فيها ميثودز و ايقينتس و ديليحيتس بس مجرد ديفينيشنز فقط ولا يمكن وضع اى امبليمينتيشن بداخله و مبيتحطش فيها اى اكسيس موديفايرز هههههه :D

الامبلمينتيشن بقه بيبقى فى الكلاس او الاستراكت الى وارث من الانترفيس ده على فكره الكلاس مينفعش يورث من اتنين كلاس بس ينفع يورث من اتنين انترفيس :D و يعملهم امبليمينيشن و كذلك الاستراكت مينفعش يورث من اى حاجه اصلا الا من الانترفيس و ينفع يورث كذا انترفيس فى نفس الوقت

ملحوظه مهمه برضوا الانترفيس ينفع يورث من انترفيس و الاتنترفيس التالث يورث من التانى و هكذا بس الى هيعمل امبليمينتيشن للحفيد بقه لازم يعمل امبليمينتيشن لكل حاجه فيهم كلهم بقه يعنى بفرض انترفيس 3 ورث من 2 و اتنين ورث من 1 و انا جيت عملت كلاس بيورث من الانترفيس 3 ساعتها لازم اكتب جوه الكلاس كل فانكشنز انترفيس 1 و 2 و 3

namespace ConsoleApplication1

{

interface I\_one

{

void Print();

}

interface I\_Two

{

void Display();

}

class Employee\_Cls : I\_one , I\_Two

{

public void Print()

{

Console.WriteLine("Hello From InterFace 1 in Cls");

}

public void Display()

{

Console.WriteLine("Hello From InterFace 2 in Cls");

}

}

struct Employee\_Struct : I\_one, I\_Two

{

public void Print()

{

Console.WriteLine("Hello From InterFace 1 in Struct");

}

public void Display()

{

Console.WriteLine("Hello From InterFace 2 in Struct");

}

}

class Program

{

static void Main(string[] args)

{

Employee\_Cls Cls\_Obj = new Employee\_Cls();

Cls\_Obj.Print();

Cls\_Obj.Display();

Employee\_Struct Struct\_Obj = new Employee\_Struct();

Struct\_Obj.Print();

Struct\_Obj.Display();

}

}

}

MB

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

Explicit interfaces

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

بص يا سيدى فى مشكله دلوقتى ان انا ممكن اعمل اتنين انترفيس و الاتنين انترفيس ممكن يبقى فيهم نفس المثود بنفس الاسم بنفس البراميترز من الاخر نسخه من بعض طب دلوقتى انا كمحمود افرق قى الامبليمينتيشن بينهم اذاى ؟؟؟  
هنا بقه يجى دور الاكسبليسيت انترفيس و ده كود عشان تفهم بيتعمل اذاى

namespace ConsoleApplication1

{

interface I\_one

{

void Print();

}

interface I\_Two

{

void Print();

}

class Program : I\_one , I\_Two

{

void I\_one.Print()

{ Console.WriteLine("Print() From InterFace 1 "); }

void I\_Two.Print()

{ Console.WriteLine("Print() From InterFace 2 "); }

static void Main(string[] args)

{

Program obj = new Program();

((I\_one)obj).Print(); // Obj Casting to Interface 1 Type

((I\_Two)obj).Print(); // Obj Casting to Interface 2 Type

// Another Solution By Pointing

I\_one I\_one\_Obj = new Program();

I\_one\_Obj.Print();

I\_Two I\_Two\_Obj = new Program();

I\_Two\_Obj.Print();

}

}

}

MB

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

Abstract Class

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

الابستراكت كلاس ده من الاخر Base Class ده معمول عشان يبقى اب للكلاسات العاديه و مش بيتعمل منه اوبجيكت خالص و ينفع ابستراكت يورث من ابستراكت بس مينفعش يورث من كلاس عادى او انترفيس بس المؤكد انه كلاس بيورث الكلاسات العاديه

من مميزات الابستراكت كلاس انه بيجمع بين الانترفيس فى جزئيه انك تعمل فيه الديفينيشن بس و كمان ممكن تحط فيه ميثودز كامله عادى

و ده بيتم بانك لو عاوز تعمل ميثود اجباريه فى الكلاسات الى هتورث منه بتعمل الكلاس دى من نوع Abstract  
و لما تيجى تعمل الامبليمينتيشن بتعمها بتعملها انها Override على الى فى الابستراكت كلاس   
بينما لو عاوزها عاديه اشطه يبقى اكتبها بشكل عادى و قشطه و فل الفل و عنب تعالى اوريك بقه فى الكود الكلام ده

و هنا هتلاحظ ان الفيلدس الى انت عاملها ديكليريشن فى الابستراكت كلاس مش بتحتاج تكتبهم تانى فى الكلاس الابن الى بيورث و كذلك الفانكشنز العاديه بيتم وراثتها اوتوماتيك

اما بقه الفانكشنز الى معموله انها ابستراكت فى دى مجبر انك تعملها امبليمنتيشن فى الكلاس الوريث (الابن)

namespace ConsoleApplication1

{

public abstract class Abstract\_Cls

{

public string Name;

public abstract void Print();

public void Display()

{

Console.WriteLine("Hello In Normal Func in Abstract Class");

}

}

class Program : Abstract\_Cls

{

public override void Print() {

Console.WriteLine("Hello In Override Func inherted from Abstract Class");

}

static void Main(string[] args)

{

Program Obj = new Program();

Obj.Display();

Obj.Print();

Obj.Name = "Mahmoud";

Console.WriteLine("My Name From Abstract Field : {0}", Obj.Name);

}

}

}

MB

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

Abstract Class VS Interfaces

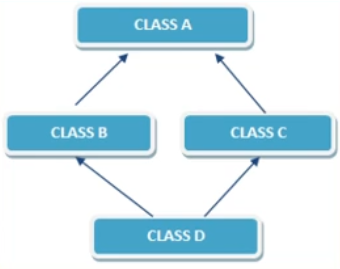
------------------------------------------

|  |  |  |
| --- | --- | --- |
| م | Abstract | Interfaces |
| 1 | ينفع اعمل امبلمينتيشن داخل الكلاس الابستراكت يعنى ينفع احط جواه فانكشن كامله عادى | مينفعش |
| 2 | ينفع احط جواها فيلدس و بروبرتيس | مينفعش |
| 3 | يرث من كلاس عادى او ابستراكت اخر و يرث من انترفيس دون اى مشكله | يرث من انترفيس فقط |
| 4 | بيتكتب جواها اكسيس موديفاير | بيتكتبش فيها اكسيس موديفاير |
| 5 | بنعتبره ذي اى كلاس بس الى بيميزه انه  1- مش بيتعمل منه اوبجيكت لانه معمول عشان يتورث منه بس  2- بيقبل انى اعمل فيه امبلمنتيش لميثودز عاديه او انى اعمل جواه ميثودز ابستراكت و احط فيها فقط الدفنيشن و ابقى اعملها امبلمينتشن في الوريث | دى بنعتبرها تمبلت بيتورث منه فقط عشان يربط الجينيورز انهم ميهبدوش بمزاجهم  و طبعا مينفعش اعمل منه اوبجيكت برضوا |
| 6 | يمكن لاى كلاس وراقه كلاس واحد ابستراكت في اللاين الواحد لان الابستراكت في الاخر كلاس و السى شارب مينفعش اورث كلاسين في لاين واحد | يمكن وراثه 10000000 انترفيس في لاين واحد و اكتر عادى يعنى |

MB

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

Problems of multiple class inheritance in Arabic

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

دلوقتى انا عندى كلاس A فيه فانكشن و جيت في كلاس C عملت وراثه من A و عملت اوفرايد للفانكشن الى في ال A داخل ال C و نفس الكلام عملته مع كلاس B   
يعنى دلوقتى B , C وارثين من كلاس A و عاملين اوفرايد للفانكشن بتاعته الاتنين   
جينا بقه عملنا كلاس جديد خالص اسمه D و قولنا نعمل وراثه من C بس مش هيقول لأ   
نعمل وراثه من B بس برضوا مش هيقول لأ   
اما بقه لو قولناله اورث من الاتنين ساعتها هيقول لأ   
ليه بقه لانه هيبقى مش فاهم ياخد انهى فانكشن الى في C ولا الى في ال B

طب حل المشكل هدى ايه بقه

1. استخدم انترفيس اكيد و لما تيجى تعمل اوبجيكت روح قوله من انهى انترفيس بالظبط بالبوينتينج

و هوريك مثال الصفحه الى جايه

MB

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

Multiple class inheritance using interfaces

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

أولا كود المشكله بتاعت الوراثه من اكتر من كلاس

namespace LearningCSharp

{

public class A

{

protected void Method()

{

Console.WriteLine("Hello from CLS A");

}

}

public class B : A { }

public class C : A { }

public class D : B,C { } // this is the Error Line Test it ..

class Program

{

static void Main(string[] args){ }

}

}

حل المشكله

namespace LearningCSharp

{

interface IA

{

void Method();

void AMethod();

}

interface IB

{

void Method();

}

public class AB : IA, IB {

public void AMethod()

{

Console.WriteLine("Hello AMethod from CLS AB");

}

public void Method()

{

Console.WriteLine("Hello Method from CLS AB");

}

}

class Program

{

static void Main(string[] args)

{

AB Obj = new AB();

Obj.AMethod();

Obj.Method();

}

}

}

هو الكود ده فعلا حل المشكله بس انا برضوا لسه معرفتش هو جاب انهى Method(); و كده حصل Ambigus بس مش مأثر على الكود و هيأثر على المدى البعيد و ككود نضيف مينفعش

طب اذاى احل المشكله دى بحيث ابقى كاتب كود نضيف جدا   
يبقى امشى ذي الكود ده بالظبط

namespace LearningCSharp

{

interface IA {void AMethod();}

public class A : IA { public void AMethod(){

Console.WriteLine("Hello AMethod from CLS A");} }

interface IB {void BMethod();}

public class B: IB { public void BMethod(){

Console.WriteLine("Hello BMethod from CLS B");} }

public class AB : IA, IB {

A objA = new A();

B objB = new B();

public void AMethod() { objA.AMethod();}

public void BMethod() { objB.BMethod();}

}

class Program

{

static void Main(string[] args)

{

AB Obj = new AB();

Obj.AMethod();

Obj.BMethod();

}

}

}

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 هدى و امنه 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

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

access modifiers

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

|  |  |  |  |
| --- | --- | --- | --- |
| Access Modifier | وصفه | types | Type Members |
| Public | ده يا سيدى انت بتعمله و المشروع كله بيقدر يشوفه يعنى الكلاس بتاعه و الى بيورث منه و الى بيورسش كمان و غريب عنه و اى كلاس معدى في الشارع بيقدر يشوفه و كمان لو بتكلم مشروع تانى المشروع التانى بيشوفه برضوا | yes | yes |
| private | ده يا كبير بيتعمل في الكلاس بتاعه و محدش بيشوفه غير الكلاس بتاعه بس | NO | Yes and Default |
| protected | الكلاس بتاعه بيشوفه و الكلاس الى وارث منه | NO | yes |
| internal | ده ذي البابليك بس الفرق ان ده لو عندك مشروعين المشروع التانى مش هيشوفه ابدا فقط مشروعه و الكلاس الى فيه هي الى بتشوفه اما اى مشروع تانى مبيشوفوش | Yes and Default | yes |
| Protected Internal | ده بيجمع بين صفات الانترنال و البروتيكتيد و بالتالى هو بيتشاف في اى مكان في مشروعه اما لو مشروع تانى ف في الحاله دى المشروع التانى مش بيشوفه الا في حاله واحده بس ان يبقى كلاس من المشروع التانى بيورث من الكلاس بتاعه الى في المشروع الأول بس كده | NO | yes |

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

using System;

namespace CsharpLearning

{

class Program

{

static void Main(string[] args)

{

AreEqual(1, 1); // true

AreEqual("1", "1"); // true

AreEqual("1", 1); // false

AreEqualByGeneric<int>(1, 1); // true

AreEqualByGeneric<string>("1", "1"); // true

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

}

static void AreEqual(Object X, Object Y)

{

if (X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

static void AreEqualByGeneric<GenericType>(GenericType X, GenericType Y)

{

if(X.Equals(Y))

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

}

}

**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**

using System;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

bool Res = Calculator<int>.AreEqualByGeneric(1, 1); // true

if (Res)

{

Console.WriteLine("Equal");

}

else

{

Console.WriteLine("Not Equal");

}

}

}

public class Calculator<GenericType>

{

public static bool AreEqualByGeneric(GenericType X, GenericType Y)

{

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

}

}

}

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

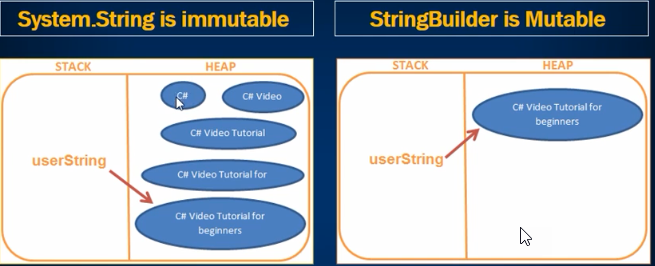
##################################################################

String VS StringBuilder

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

String : بص يا سيدى الاسترينج العادى لو جيت تعمل كونكاتيونيشن فهو بيفضل يخزن فى اماكن كتير جدا فى الميمورى يعنى لو هتعمل كونكاتيونيشن بين مليون كلمه سترينج فكده حجز مليون مكان فى الميمورى

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);

}

}

}