C# Interview Questions Summery By Mahmoud Badawy

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

MB

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

اول حاجه كده ملخص هحط فيه كل الاساله و الاجوبه الخاصه بال C# و Dot net للانترفيوهات

و نبتدى معاكم اول سؤال

MB

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

In Debugging What is Different Between (Autos , Locals , Watches)

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

A screenshot of a computer

Description automatically generated  
Locals : دى و انت بتعمل ديباج بتجيبلك كل قيم الفاريبولزو البراميتارز حتى الى فى فانكشن بتناديها و الى عديت عليها و الى انت فى نفس الاسكوب بتعها و بتعرفك انت فى انهى اسكوب دلوقتى و اى تغيير فى اى فاريبول من الفاريبولز دى بتعرفها لان لون الفاريبول و قيمته بيتغيروا للون الاحمر و ده معناه ان حصل تغيير بعد الخطوه الى انت مشيتها دى فبتعرف التغييره حصلت امتى من متابعه ال locals دى لانها بتتعامل مع الاسكوب و بالتالى اى مشكله خاصه بقيمه ما تم تغييرها و انت مش فاهم اتغيرت فين و امتى بتعرفه عن طريق ال locals

Autos : نفس الى اللوكال بتعمله بس الفرق انها بتراقب الفاريبولز فقط و ملهاش دعوه باى براميتر مبتشوفهمش هى بتركز على الفاريبولز فقط و التغيرات الى بتطرأ عليها كل شويه سواء فى نفس الاسكوب او فى اسكوبس مختلفه

Watches : دى بقه يا سيدى بتشتغل بمزاجك انت يعنى تحطلها فاريبول معينه و تتبعها هى على وجه الخصوص علشان متوجعش دماغك بالباقى و عادتا احنا بنستخدم دى بدلا من اللوكالز علشان نعرف نركز على الفاريبول الى بيحصل فيها المشكله بسرعه

MB

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

Ternary Operator Vs If-Else

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

الفرق الواضح جدا ان if – else مش لازم يعمل ريتيرن لقيمه - ينفع نحط جواها Retern للقيم او ممكن نعمل كود مبيعملش Return لاى قيمه و يبقى مجرد قراءه للداتا او تغيير فى قيمه فاريبول بس مش هعمله ريتيرن عادى

اما ال Ternary Operator : ده لازم يعمل return لقيمه فلو معملتش ريترين لقيمه منه هيديك ايرور

A screen shot of a computer

Description automatically generated

MB

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

Do While Vs While

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

While : مش هيشتغل لو الى جواه مبيحققش الشرط

Do While : اول مره هيشغل الكود الى جوه ال do حتى لو كان مش محقق الشرط الى جوه while بعد كده يبتدى يطبق الشروط لما يخش بقه فى ال while

var x = 11;

while (x < 11 && x >=0)

{

Console.WriteLine("Hello from While");

--x;

}

var x2 = 11;

do

{

Console.WriteLine("Hello from Do While ");

--x2;

}

while (x2 < 11 && x2 >= 0);

MB

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

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

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

طيب ايه الفرق بينهم ؟؟

ال Value Type ذى مثلا ال ال

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

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

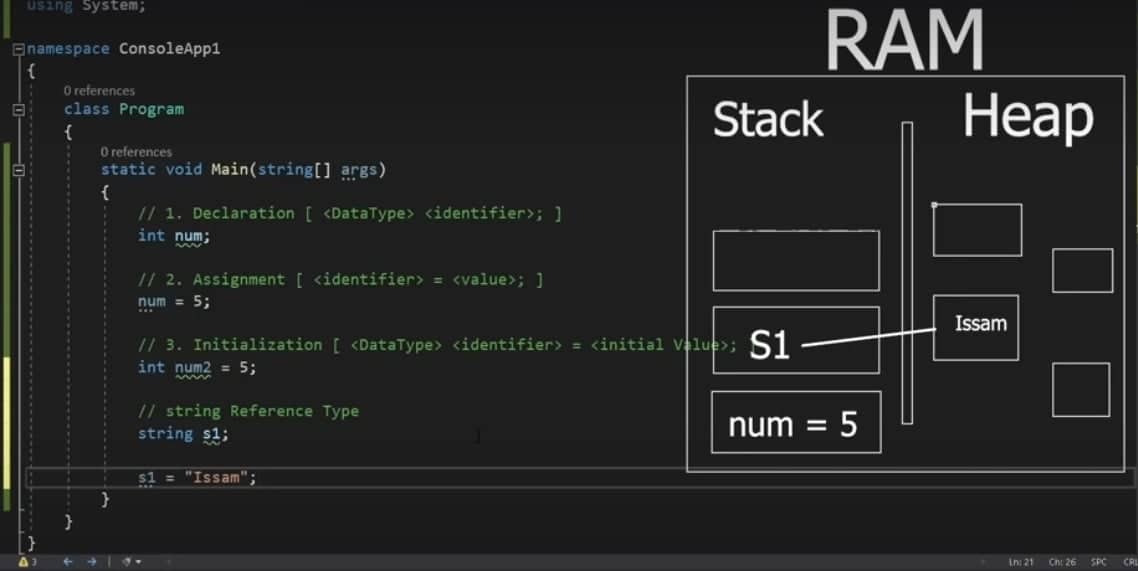
يعنى لما باجى اعرفها ذى ما فى الابلكيشن ده كده

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

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

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

و الصوره دى هتوضح اكتر للناس علشان تفعم طريقه التخزين



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

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

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

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

void ValueType()

{

int num1 = 5;

int num2 = num1;

num1 = 10;

Console.WriteLine(num1); // 10

Console.WriteLine(num2); // 5

}

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

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

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

void RefrenceType()

{

int[] num1 = { 5 };

int[] num2 = num1;

num1[0] = 10;

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

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

}

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

MB

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

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

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

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

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

و دى مثال للكلام ده

using System;

class Program

{

static void Main(string[] args)

{

var IntNum = 5;

Console.WriteLine(IntNum); // 5

IntNum = "5"; // syntax Error

}

}

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

بينما فى ال dynamic هيشتغل عادى خالص

using System;

class Program

{

static void Main(string[] args)

{

dynamic IntNum = 5;

Console.WriteLine(IntNum); // 5

IntNum = "5";

Console.WriteLine(IntNum); // 5

}

}

MB

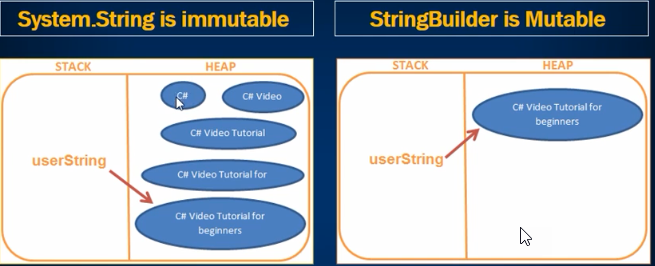
##################################################################

String VS StringBuilder

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

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

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



اذاى اكتب الاسترينج و الاسترينج بيلدر بقه فى الكود ؟؟

using System;

using System.Text;

namespace CsharpLearning

{

class Program

{

public static void Main(string[] args)

{

string FullName = "My ";

FullName += "Name is Mostafa ";

FullName += "Badawy";

Console.WriteLine(FullName);

StringBuilder MyStr = new StringBuilder("My ");

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

MyStr.Append("Badawy");

Console.WriteLine(MyStr);

}

}

}

MB

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

Implicit Convert VS Explicit Convert

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

Implicit Convert : بص يا سيدى ده التحويل الى الكومباير بيعمله Automaticly بدون ما يحتاج اى كود زياده منك و ده لان اوقات بينفع انى اخزن مثلا ال int جوه ال Double لان الانتيجر اقل فى المساحه من الدوبل فبينفع اخزنه فيه بس هو الكومبايلر بيخزن ال int ده جوه ال double بعد ما بيحوله فى الخفاء كده الى double و لكن لو جينا نقلب الايه و نحط ال Double جوه ال int فالكومبيلر هيرفض و ده لانه مش قادر يحول ال double الى int علشان يخزنه جوه ال int و ده راجع لان ال double مساحته اكبر فممكن اكون مخزن فيه رقم كبير و علشان احوله ل int فوارد يحصل Date lose و بالتالى يحصل مشكله كبيره و علشان كده الكومبايلر بيطلع ايرور فى الحاله دى و يقولك لأ مش نافع

Explicit convert : ده بقه يا سيدى تحويل Manual و ده بقه يا سيدى كانك كده بتقوله انا مسؤول على الى انا بعمله فحاولى ال Double الى int عادى و مش مشكله لو حصل Data lose بس علشان النوعيه دى تشتغل اكيد محتاج تكتب كود يفعل التحويل ده علشان تبقى انت كتبتها بأيدك و مسؤول عن اى داتا هيتم فقدها يا صديقى

static void Main(string[] args)

{

int IntNum = 10;

double Num = IntNum; // implicitly Convert

Console.WriteLine(Num);

IntNum = Num; // Syntax Error (Can not implicitly Convert From double to int)

IntNum = (int)Num; // Explicit Convert

Console.WriteLine(IntNum);

}

MB

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

Compilation Process

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

A screen shot of a computer

Description automatically generated

أول حاجه احنا بنكتب الكود فده الى اسمه ال Source Code ده الكود بتعنا الى بنفهمه ده اول ما بنعمل Build الى بيحصل ان ال Compiler بيبدأ يشتغل علشان يحول ال Source Code ل Machine Code   
  
طيب خطوات حدوث الكلام ده ايه بقه ؟؟

اول حاجه عندى Souce Code فعمل بيلد للبروجيكت او Run الى هيحصل ان الكومبايلر هيششتغل و ياخد الكود يحوله الى لغه وسيطه IL او MSIL او CIL و اللغه الوسيطه دى هى كوبرى بين ال Machine language , C# language و لذلك بعد تحويل السورس كود الى لغه وسيطه عن طريق الكومبايلر بيتم تحويل اللغه الوسيطه الى machine Code عن طريق ال CLR & JIT

طيب ايه هو ال Source code ؟

ده الكود الى احنا بنكتبه و بنفهمه انا و انت و اى ديفلولوبر

طب ايه هو ال IL ؟

ده ال ddls & exe files الى هو مكتوب عن طريق ال intermediate language لسه متحولش ل machine language

طب ايه هو ال CLR ؟

ده بقه يا باشا لما بتدوس على Exe file علشان يشغل البرنامج : الى بيحصل ساعتها ان ال exe file بيتعمله Compile عن طريق ال Just in time Compiler ليتم تحويله ال Machine Code و الى بيتم التعامل معاه عن طريق ال CLR الى هى هنا بنعتبرها ال Virtual Machine بديله قبل مرحله البروسيسور

يعنى مجرد ما ندوس على ال exe file بيشغله

ال CLR , JIT = Common language Runtime – just in time Compilation و الفايل الى مكتوب ب IL ده بيروح متحول فى الران تايم الى Machine Code البروسيسور يقدر يفهمه بقه بدون التركيز فى موضوع المعماريه يعنى يشتغل على اى بروسيسور عادى لان اعتماده على virtual CPU

MB

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

Manage code VS Un manage Code

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

ال C# هى Manage Code و ده لان هى مش بتطلع ال machine Code بشكل مباشر و ده لانها بتستهدف Virtual Machine كبديل لل Processor علشان لو المعماريه بتاعت البروسيسور مختلفه عن المكتوب فى الكود ميحصلش مشكله لان البروسيسور و البرنامج معماريتها مختلفه فاحينانا هتلاقى حاجه بتحتاج برامج 64 و تانيه عاوزه 32 و لذلك ال manage Code بيحل المشكله دى بانه يعمل intermediate language وسيطه تخلى الماشينج كود تنفع لاى حاجه

على عكس ال unmanage Code : دى و انا بعمل البرنامج لازم ابقى محدد البروسيسور الى هتتعامل معاه بالظبط و اكتبله هو الكود مخصوص لان لواتكتب و اتبنى بمعماريه مختلفه عن البروسيسور مش هيشتغل على البروسيسور ده ابدا و ده لان الكود بيتعامل بيتحول بشكل مباشر الى machine Code فمفيش وسيط يظبط الماشين كود قبل ما تتعمال مع البروسيسور

MB

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

Manage code VS Un manage Code

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