C# Summery part-1 By Mahmoud Badawy

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

الملخص يحتوى على

- Design Patterns

المستخدمه فى الشرح   
<https://refactoring.guru/design-patterns/singleton>

MB

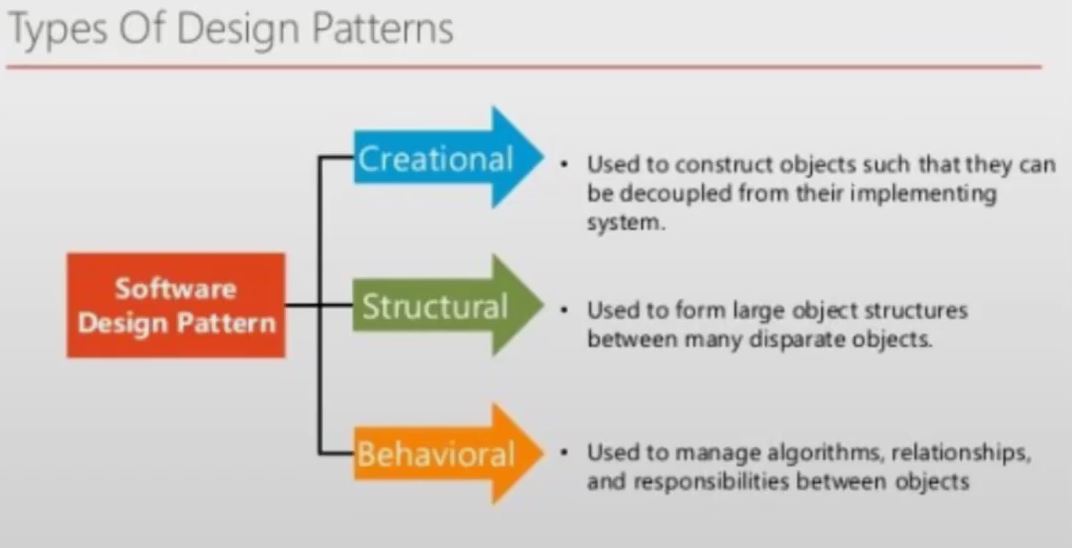
##################################################################

Design Pattern Importance

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

الناس لما عملت ال OOP و ابتدوا يشتغلوا بيها و يعملوا مشاريع كبيره ابتدت تظهر مشاكل سواء فى ال Creation لل instance او سلوكه او طريقه تعامله مع الانستانس المختلفه عنه و ده الى خلاهم فكروا يعملوا حلول باستاندرد محترم يخلى الكود نضيف و فى نفس الوقت يحل المشكله الى بيوجهوها .و هى دى اهميه ال Design Pattern

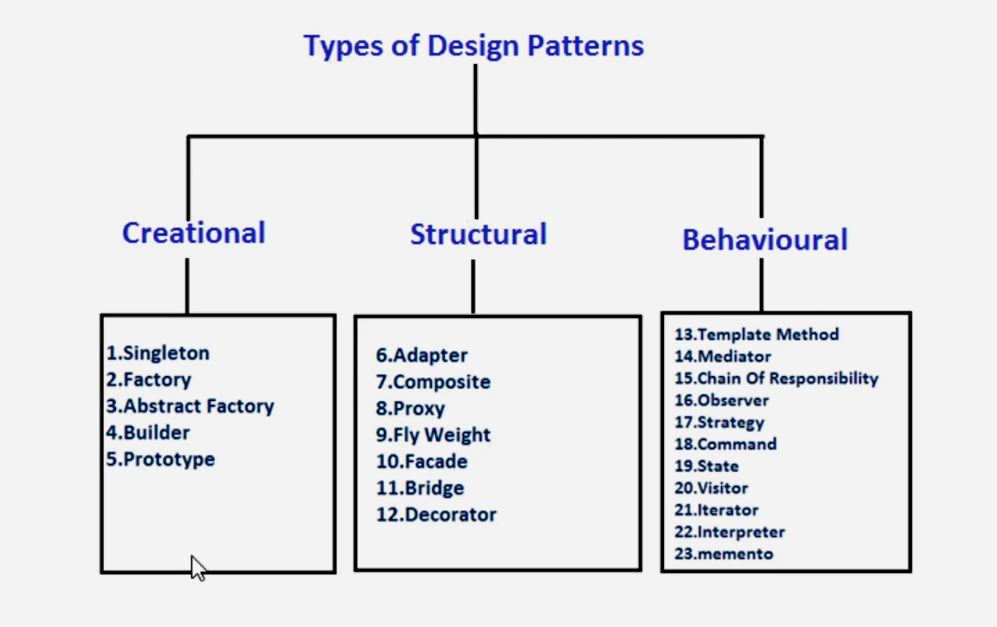
فى تلات انواع للديزاين باترن



ال creational : ده المسؤول عن انشاء ال objects .

ال Structural : ده المسؤول عن العلاقه بين ال objects المختلفه .

ال Behavioral : ده المسؤول عن السلوك الى بيحصل بين ال objects دى الى هو بيبقى سبب فى نقل البيانات بينهم ذى مثلا ال events , Algorithms , states



MB

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

Singleton DP

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

قبل ما يعملوا ال singleton كان الواحد ممكن يعمل كلاس ما و يبتدى يستخدمه فيعمل منه new instance ثم فى مكان تانى يعمل new instance تانى و هكذا لحد ما ممكن يكون بيعمل ملايين ال instances و المشكله ان كل دول محفوظين فى الميمورى و واخدين مساحه كبيره منها و مش بيتمسحم و بالتالى الريسوريس بتاعت الجهاز بيتم استهلاكها بدون داعى و من هنا طلع فكره ال Singleton علشان تحل المشكله دى



ال Singleton بقه فكرتها ان انا هعمل only One Object من الكلاس و بالتالى هيبقى استهلاك الميمورى قلبل جدا جدا و بالتالى الكود هيبقى سريع جدا .

تعالى بقه اوريك المشكله قبل ال SingleTon و الحل لما طبقناه عمل ايه

MB

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

Before Singleton DP

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

المشكله قبل ال singleton هو انى عندى كلاس و كل اما اعمل new instance منه بيروح يخزن ال new instance ده فى الميمورى و للاسف مبيكونش مسح القدام و بالتالى كده انا بعمل instances كتير جدا جدا من class واحد و ممكن مبقاش محتاج كل ده و بالتالى ده استهلاك لموارد الكمبيوتر بشكل كبير و هيبطء البرنامج الى انا بعمله جدا و ده الكود الى بيوضح الفكره

using System;

namespace Before

{

internal class Program

{

static void Main(string[] args)

{

// with no Singleton it will create New 5 Instance in Memory

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

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

Counter ObjA = new Counter();

Counter ObjB = new Counter();

Counter ObjC = new Counter();

Counter ObjD = new Counter();

Counter ObjE = new Counter();

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

Console.WriteLine($"Number Of Intances {Counter.NumberOfInstance}");

Console.ReadKey(true);

}

}

}

هنا انا بنادى على الكلاس 5 مرات و كل مره من ال 5 مرات دول بينشأ instance new من كلاس ال counter

using System;

using System.Collections.Generic;

using System.Text;

namespace Before

{

internal class Counter

{

public int Count = 0; //it is not shared so it is repeated in any new intance

public static int NumberOfInstance = 0; //it is shared between all instances to Count Instrances

public Counter()

{

Count++;

NumberOfInstance++;

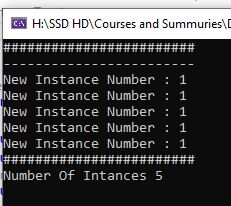
Console.WriteLine($"New Instance Number : " + Count);

}

}

}

و الدليل على انه بينشأ New instance كل مره بعمل منه object هى نتايج البرنامج و اهه علشان تبقى فاهم



**طيب انا بقه مش عاوز كده انا عاوز انى اعمل Creation لل instance مره واحده و فى اى وقت اناديله تانى يروح رايح واخد نفس ال instance و ميعملش غيرها و بالتالى انا كده هتعامل مع only One Instance و بالتالى استهلاك الميمورى هيكون اقل بكتير و بالتالى سرعه البرنامج هتبقى كبيره جدا خصوصا لما نستخدم الباترن ده مع الفانكشنز الى بتتعامل مع heavy Data**

اول حاجه هنحل المشكله الى فوق دى بال Singleton DP و ده بقه بيتعمل اذاى ؟؟؟

1- اول حاجه نخلى ال Constractor برايفيت Private و ده علشان فى ال Client side محدش يعرف يعمل new instance of class

private Counter()

{

Count++;

NumberOfInstance++;

Console.WriteLine($"New Instance Number : " + Count);

}

2- ثم بعد كده هعرف جوه الكلاس بتاعى ال instance من نوع public static Counter ذى كده و هو ده هيبقى ال instance الوحيد الى بتعامل معاه علشان كده عملته Static يا ريس

public static Counter Instance = null;

3- المفروض بقه هعمل فانكشن كل شغلانتها انها بتشوف لو ال static instance ده ب null فتروح تملاه و تديله new instance اما لو مش ب null معنى كده انه فيه instance فكده ياباشا تروح قيلاله خد القديم استعمله يابا مش هعملك حاجه جديده

public static Counter GetInstance()

{

if (Instance == null)

{

Instance = new Counter();

}

return Instance;

}

و الفانكشن دى اتعملت static علشان تبقى Shared و علشان اقدر اناديها باسم الكلاس و طبعا هى بترجع حاجه من نوع الكلاس Counter Class لان هى دى المسؤوله عن انشاء الاوبجيت اصلا او انها تديك الاوبجيكت القديم الى معمول اساسا بدلا من انها تعملك حاجه جديده فتهلك الميمورى

و بالتالى الكود بالكامل اهه

using System;

namespace After\_Simple\_Singleton

{

internal class Program

{

static void Main(string[] args)

{

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

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

Counter ObjA = Counter.GetInstance();

Counter ObjB = Counter.GetInstance();

Counter ObjC = Counter.GetInstance();

Counter ObjD = Counter.GetInstance();

Counter ObjE = Counter.GetInstance();

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

Console.WriteLine($"Number Of Intances {Counter.NumberOfInstance}");

Console.ReadKey(true);

}

}

}

ثم الكلاس بقه

using System;

using System.Collections.Generic;

using System.Text;

namespace After\_Simple\_Singleton

{

internal class Counter

{

public int Count = 0; //it is not shared so it is repeated in any new intance

public static int NumberOfInstance = 0; //it is shared between all instances to Count Instrances

public static Counter Instance = null;

private Counter()

{

Count++;

NumberOfInstance++;

Console.WriteLine($"New Instance Number : " + Count);

}

public static Counter GetInstance()

{

if (Instance == null)

{

Instance = new Counter();

}

return Instance;

}

}

}

النتيجه اهه يا صديقى بقى مش بيعمل غير One Instance من الكلاس

