برمجة متقدمة ١

المحاضرة الثامنة

EXCEPTION

الأخطاء التي قد تنتج عن البرنامج

النوع الأوّل : هي أخطاء أثناء التصميم، أي أثناء كتابة الشيفرة، وهي أخطاء يمكن اكتشافها بسهولة من خلال مترجم سي شارب الذي يتعرّف عليها ويعطيك وصفًا عنها، وقد يزوّدك أيضًا ببعض المقترحات للتخلّص منها.

النوع الثاني من الأخطاء: هي التي تنتج أثناء تنفيذ البرنامج runtime errors .

عند حدوث مثل هذه الأخطاء ترمي بيئة التنفيذ المشتركة CLR استثناءً exceptionيحتوي على معلومات بخصوص الخطأ الذي حدث،

اما ان نقوم بالتقاط هذا الاستثناء اوسيتوقّف البرنامج عن العمل بصورة مفاجئة.

الإستثناء : هو خطأ يظهر أثناء تشغيل البرنامج سواء كان خطأ مفتعل أو نتيجة سهو المبرمج عن وضع قيود محدد للتعامل مع البرنامج مثلا حقل نص يستخدم لادخال الارقام وسها المبرمج عن وضع قيد يجبر المستخدم على ادخال ارقام بدل نصوص.او مثل القسم على صفر.

يقسم معالجة الاستثناءات الى اربع اقسام.

try وفي هذا البلوك يتم وضع الكود الذي يجب معالجة استثناء له عند حدوث الاخطاء ويجب ان يتبعه اما واحد من catch او عدة جمل Catch.

catch وهي تشير الى وقوع الخطأ وتم العثور عليه وهنا يتم معالجة الخطأ عند حدوثه.

finally وتستخدم سواء حدثجب الخطأ او لم يحدث سوف يتم تنفيذ الكود الذي تحتويه مثل إغلاق ملف. نفترض اننا قمنا بفتح قاعدة بيانات او ملف وحدث خطأ فهنا يجب إغلاق الملف سواء حدث الخطأ أو لم يحدث.

throw وهي تقوم برمي الاخطأ اي بمعنى تجاهل الخطأ والانتقال الى الكود التالي لتنفيذه وغالبا ما يستخدم من قبل المبرمجين المبتدائين وذوي قلة الخبرة ولكن لا يمنع ذوي الاحتراف استخدامه خاصة لكتابة الاخطأ في ملف log.

TRY-CATCH

عبارة try-catch تتألّف هذه العبارة من قسمين:

القسم الأوّل هو قسم المراقبة try ،

القسم الثاني هو قسم الالتقاط catch الشكل "الأبسط" لهذه العبارة هو التالي:

```
Try
 {
// عبارة برمجيّة مريبة
}
 catch(Exception exp)
{
//
هنا يُلتقط الاستثناء وتتمّ معالجته
 }
```

نقوم بالتقاط استثناء من خلال هذه العبارة إذا صادفتك عبارة برمجيّة تحتوي على عمليّة حسابيّة أو على استدعاء لتابع آخر فمن الممكن مراقبتها أثناء تنفيذ البرنامج باستخدام

ملاحظات

- يمكن أن يحوي قسم try على عبارات برمجيّة
- عندما يصادف البرنامج أثناء التنفيذ خطأً ما، سيتوقّف التنفيذ عند العبارة التي سبّبت الخطأ، ثم ينتقل فورًا إلى قسم الالتقاط catch
- قسم catch يُمرَّر إليه وسيط من الصف Exception
- الصف Exception موجود ضمن نطاق الاسم System
- وهو الصف الأب لجميع الاستثناءات إذ أنّ أي استثناء مهما كانت صفته (بما فيها الاستثناءات التي يمكنك أن تكتبها أنت) يجب أن ترث من هذا الصف.

بعد حدوث الاستثناء والانتقال إلى قسم catch ،

يحتوي الوسيط exp على معلومات حول مكان حدوث الاستثناء وسبب حدوثه،

وغيرها من المعلومات التي قد تكون مفيدة لمستخدم البرنامج.

ملاحظة: ان البرنامج لن يدخل إلى القسم catch أبدًا ما لم يحدث استثناء ضمن القسم.try

مثال:

```
using System;
namespace exception
{
class Program
{
static void Main(string[] args) {
int x = 5;
int y = 0;
int result;
try  {
result = x / y; }
catch(Exception exp)  {
 Console.WriteLine("The following error has occurred:");
Console.WriteLine(exp.Message);
} Console.WriteLine("Good Bye!");  } } }
```

```
 using System; using System.IO;
 namespace taghr {
class Program
{ static void Main(string[] args)
{  string contents;
 try  {
using (StreamReader sr = new StreamReader("myfile.txt")) { contents = sr.ReadLine();
 }
Console.WriteLine("This file contains {0} characters.", contents.Length);  }
catch(NullReferenceException nullExp)
{ Console.WriteLine("The file does not contain any data.");  }
catch(FileNotFoundException notFoundExp)
  {  Console.WriteLine("File: {0} not found!", notFoundExp.FileName); }}}  }
```

ما هو الخرج المتوقع للكود السابق؟

عبارة TRY-CATCH-FINAL

يمكن إضافة قسم أخير لعبارة try-catch اسمه final

هذا القسم يمكن له أن يحتوي على عبارات برمجيّة سيتمّ تنفيذها بعد أن يدخل البرنامج إلى القسم try العائد له.

وذلك سواءً أحدث استثناء ضمن try أم لم يحدث.

فائدة وجود هذا القسم أنّه قد نواجه أحيانًا بعض الحالات التي تتطلّب إجراء بعض المهام عندما نفرغ من قسم try مثل إغلاق بعض المصادر المفتوحة، أو تحرير الذاكرة بشكل فوري وغيرها.

الشرط الوحيد لاستخدام هذا القسم الاختياري هو أن يكون آخر قسم في عبارة try-catch.

؟؟؟؟؟؟؟؟؟؟