

Exception handling

نستخدمه علشان نعالج الأخطاء اللى بتطلع أثناء تنفيذ البرنامج
وعلشان نمنع البرنامج أنه يتوقف فجأة

أنواع الأخطاء اللى تسبب exception:

1. DivideByZeroException: لما نقسم على صفر.
2. NullReferenceException: لما نحاول null نستخدم حاجه قيمتها
3. IndexOutOfRangeException: لما المستخدم يحاول يوصل لعنصر خارج حدود ال المصفوفة
4. FileNotFoundException: لما المستخدم يحاول يوصل لملف غير موجود.
5. FormatException: خطأ في تحويل 'نصوص إلى أرقام أو أنواع بيانات أخرى

كود ال exception handling يتكون من

```
try
{
    // الكود الى ممكن يسبب ايرور
}
catch (Exception ex)
{
    // الكود الى هنعالج بيه الايرور
}
finally
{
    // كود يتنفذ دايمًا
}
```

Edited 11:43 pm ✓

مكونات الباراميتير في catch:

1. (Exception Type):
بيحدد نوع الخطأ اللي عاوزين نتعامل معاه.
لو استخدمنا النوع العام Exception، معناها إنك بتعامل مع أي خطأ بشكل عام.
لو استخدمت أنواع فرعية (زي DivideByZeroException أو FileNotFoundException)، معناها إنك بتحدد نوع معين من الأخطاء.

2. (Parameter Name):
مثل الخطأ اللي حصل.
من خلاله أقدر أوصل لمعلومات بتصف الايرور

1. أنواع ال Exceptions

أ. Built-in Exceptions

دي Exceptions موجودة في مكتبة ال .NET Framework، وبتغطي معظم سيناريوهات الأخطاء اللى ممكن تحصل.

زي

DivideByZeroException

لما نقسم على صفر

ArgumentException

بتحصل لما تبعث قيمة غير صحيحة للميثود.

InvalidOperationException

بتحصل لما تحاول تعمل عملية غير منطقية

ب. Custom Exceptions (اللي بتعملها بنفسك):

لو عندنا حالات خاصة في البرنامج، ممكن تعمل Exception خاص بيبك

11:56 pm ✓