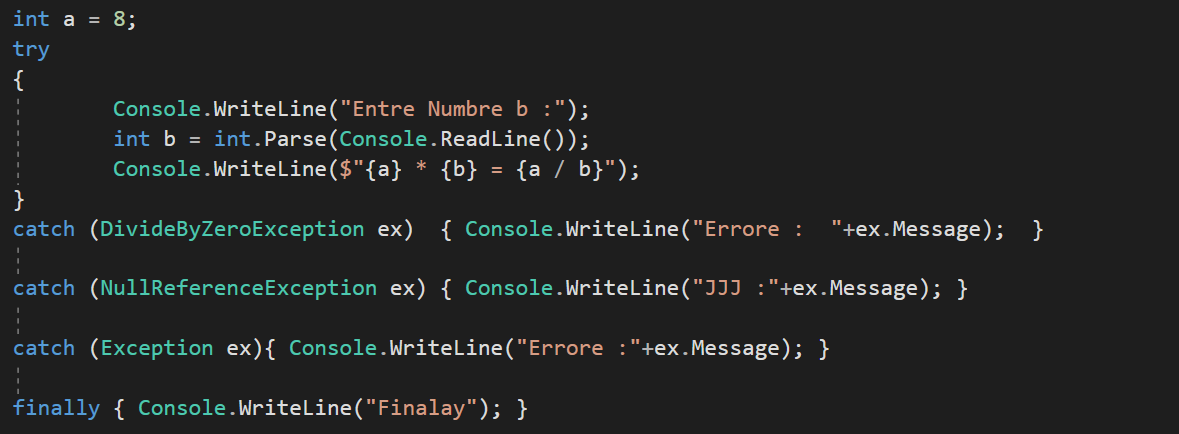
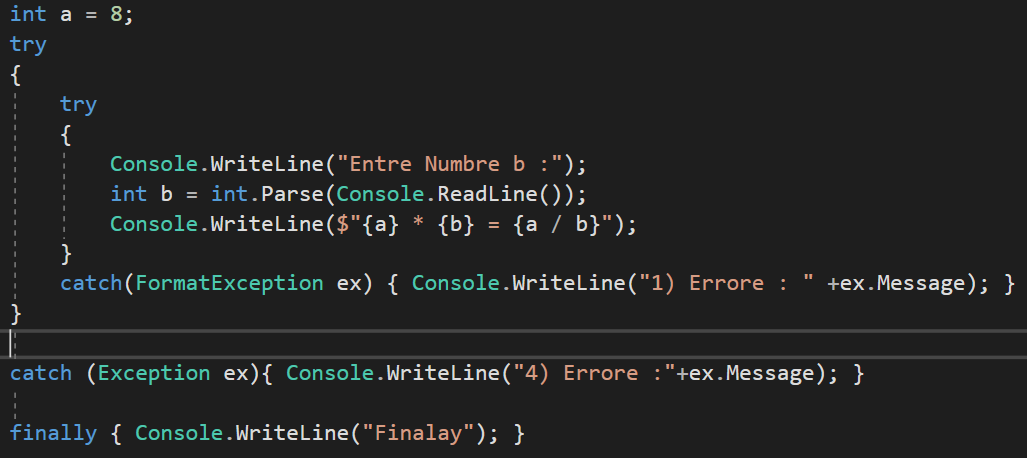
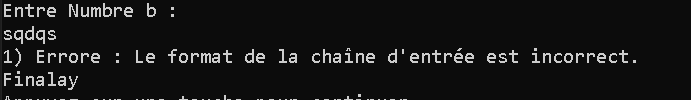
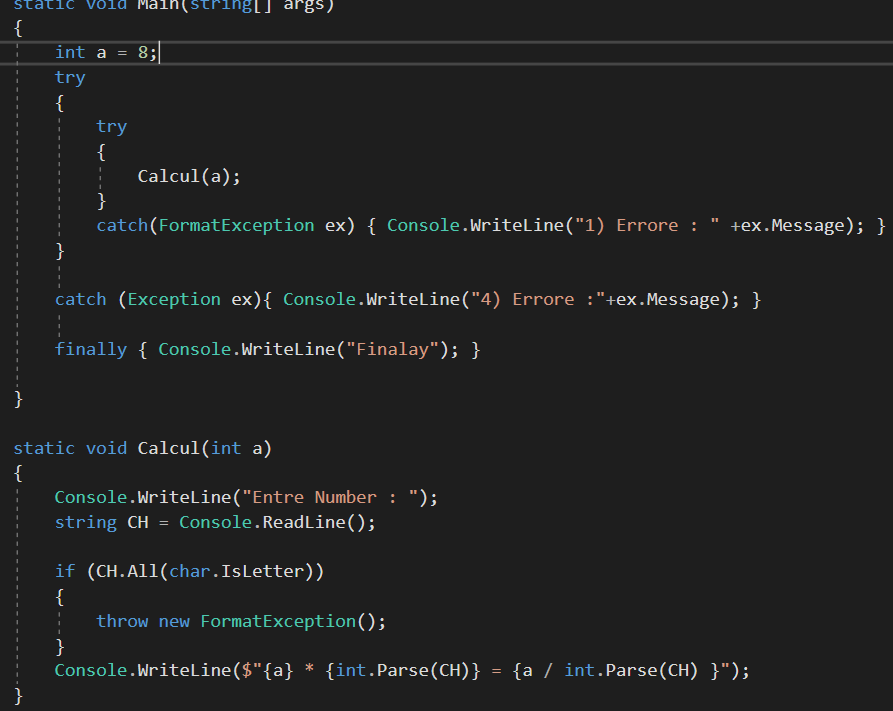
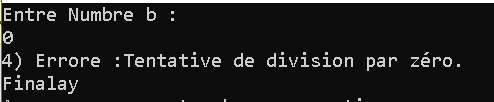
Exception :

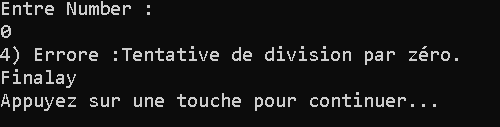
* Catch = catch (Exception ex)
* La classe Exception est la superclasse de toutes les classes Exception et peut donc gérer tous les types d'exceptions lancées dans le bloc try.



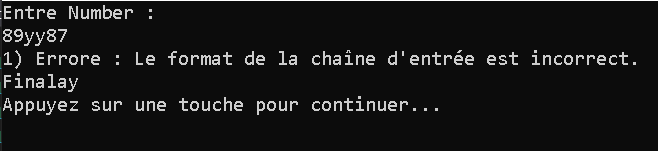
Nested try-catch

throw keyword

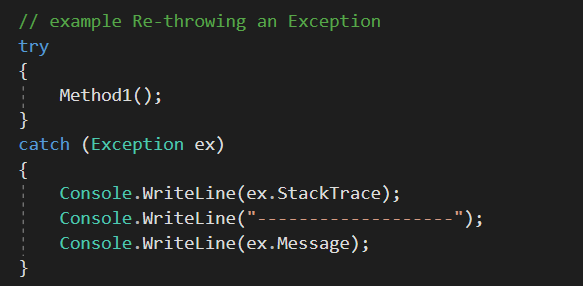
Diviser par 0 : exception passe catch(Exception ex)

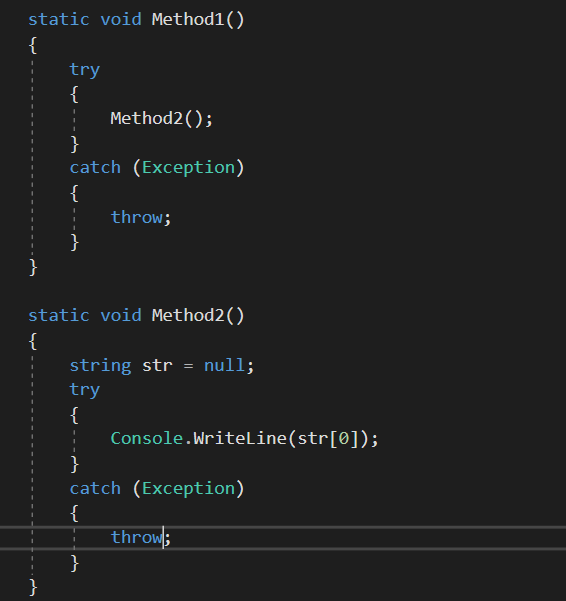
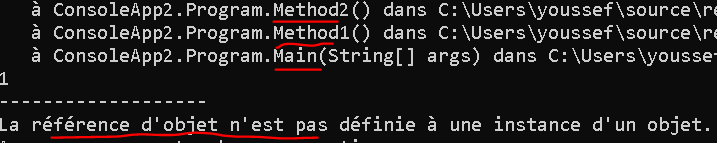


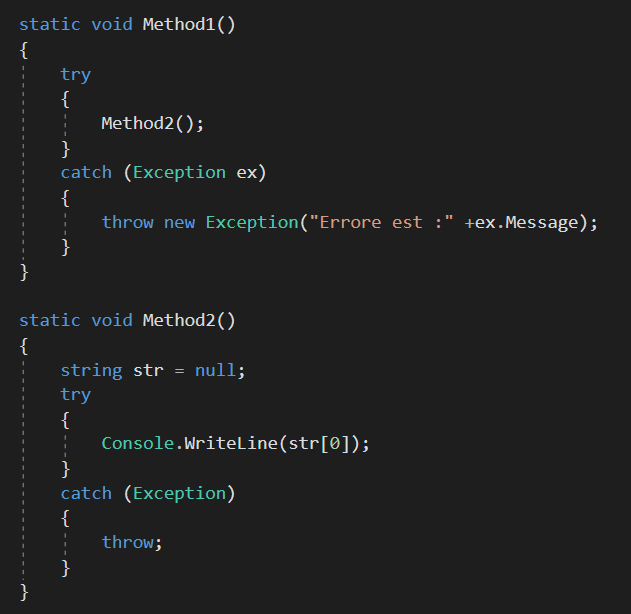
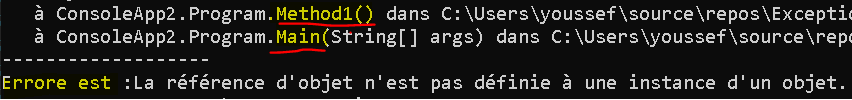
Entre chaine donc passe FormatException()

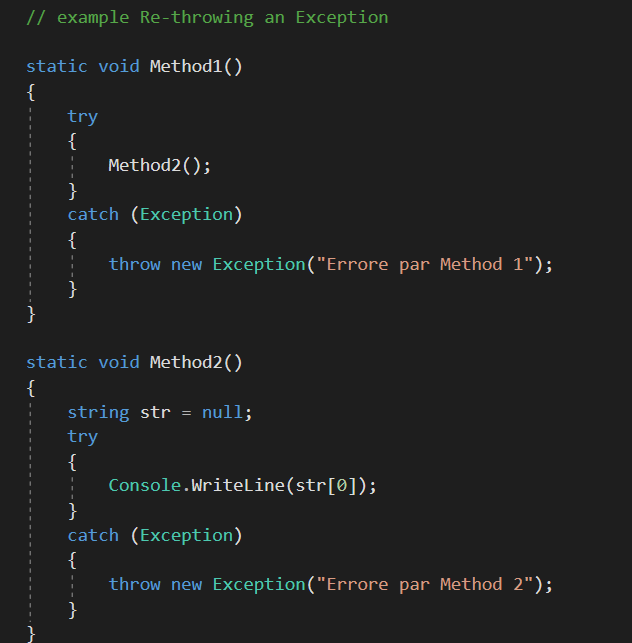


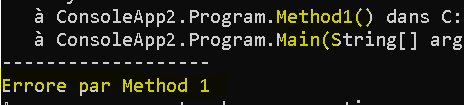
Re-throwing an Exception





* Si vous relancez une exception à l'aide du paramètre exception, il ne conservera pas l'exception d'origine et créera une nouvelle exception.





***Exception personnalisée en C :***

* Deux type exception :

Exception Système : déclenchée implicitement sous un programme par le gestionnaire d'exceptions en raison d'erreurs logiques.

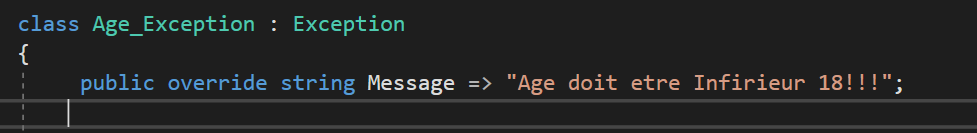
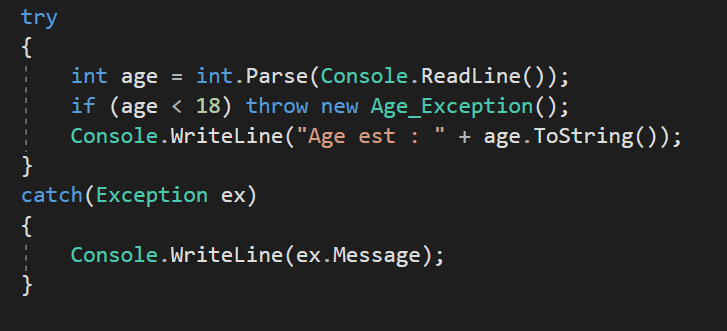
DivideByZeroException

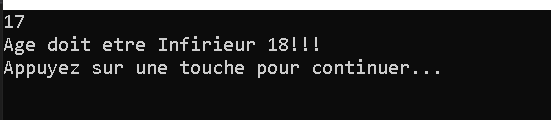
IndexOutOfRangeException

FormatException.

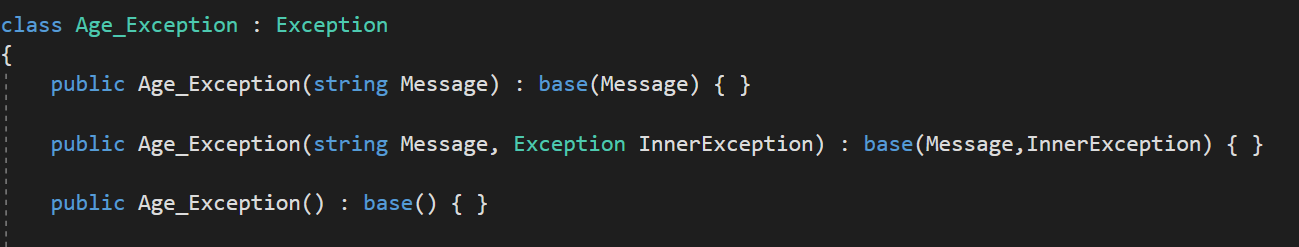
Exception d’application : déclenchée explicitement sous un programme basé sur notre propre condition.

* Créer une nouvelle classe héritant de la classe prédéfinie Exception class
* Override the virtual message .



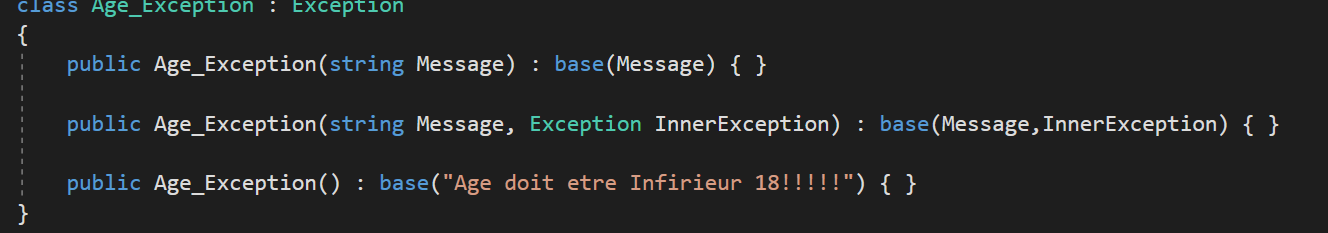


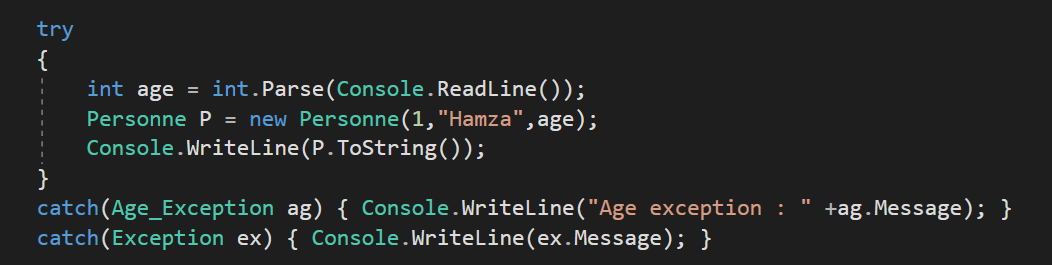
2) Method

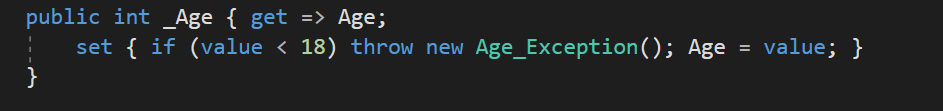


Autre method :

*Classe Age\_Exception*

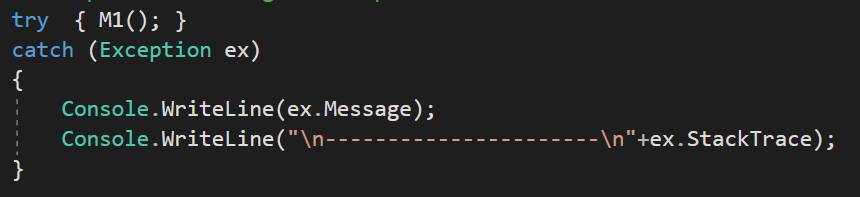
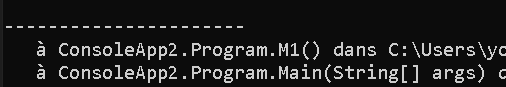
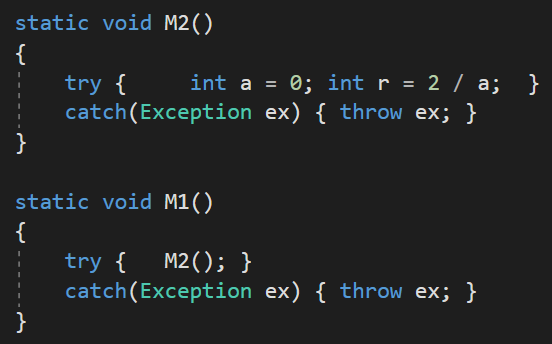


 ***Classe Programme :***

 ***Classe Age :***

Differenet entre throw ; et throw ex;

1. Cas thorw ex;



1. ***Cas thorw***

