DÉCOUVREZ
LES FORMATIONS
DE L'OFPPT

OFPPT

FILIÈRES
CONDITIONS D'ACCÈS
PROGRAMME
DIPLÔMES

PRÉSENTATION SUR LA GESTION DES EXCEPTIONS EN POO {PHYTON} :

❖ ***Realise par :***

- ✓ *Ilham barakat*
- ✓ *Assia hamilou*
- ✓ *Adam moussaoui*
- ✓ *Hatim harchane*

Annee scolaire ***2023 2024***

Formateur *othmane haqay*

Filiere *devloppement digital 104*

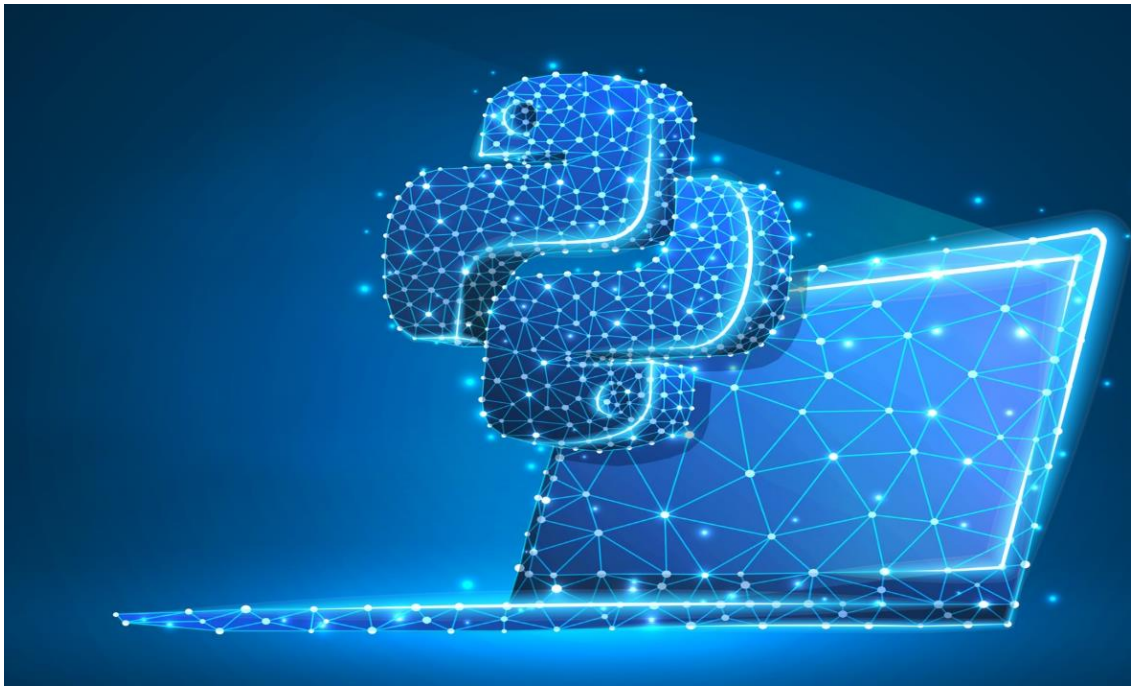
Centre de formation *ntic 2 sidi maarouf*

OFPPT

SOMMAIRE :

1)INTRODUCTION :

- **La gestion des exceptions est un aspect crucial de la programmation, permettant de gérer les erreurs et les situations inattendues qui peuvent survenir pendant l'exécution d'un programme. En Python, la gestion des exceptions est mise en œuvre à l'aide des blocs try-except, et elle est étroitement intégrée à la programmation orientée objet.**



2)types d’erreur et expérimentation :

Exemple

```
rixHT = int(input("QUELLE EST LE PRIX?"))
prixTOT = prixHT + prixHT * 0.2
print(prixTOT)

# pour résoudre le problème d'utilisation != int
while True:
    try:
        prixHT = int(input("QUELLE EST LE PRIX?"))
        prixTOT = prixHT + prixHT * 0.2
        print(prixTOT)
        break
    except ValueError:
        print("Veuillez entrer un nombre.")
```

Explication :

- Erreur de Valeur Possible :

L'erreur de valeur potentielle dans ce code est liée à la conversion de l'entrée utilisateur en un entier (int). Si l'utilisateur entre une valeur qui ne peut pas être convertie en entier (par exemple, des lettres ou des caractères spéciaux), Python lèvera une exception ValueError.

Gestion des Exceptions:

- Pour résoudre ce problème, le code utilise une boucle while True avec un bloc try-except :

Bloc try :

Le programme essaie de convertir l'entrée utilisateur en un entier avec prixHT = int(input("QUELLE EST LE PRIX?")).

Si la conversion réussit, il calcule le prix total avec la taxe : prixTOT = prixHT + prixHT * 0.2.

Il imprime ensuite le prix total avec print(prixTOT).

La boucle se termine avec break, car l'entrée était valide.

Bloc except ValueError :

Si l'entrée utilisateur ne peut pas être convertie en entier, une exception ValueError est levée.

Le programme capture cette exception et imprime un message d'erreur : print("Veuillez entrer un nombre.").

La boucle continue, demandant à l'utilisateur de réessayer jusqu'à ce qu'une entrée valide soit fournie.

- Ce mécanisme permet de gérer de manière robuste les entrées invalides de l'utilisateur, en s'assurant que le programme ne plante pas et que l'utilisateur est invité à entrer une valeur correcte. En capturant ValueError, le code empêche les erreurs de conversion et garantit que le calcul du prix total ne se fait qu'avec des valeurs entières valides.

➢ainsi qu ill y en a plusieurs tupes d erreur qui peuveant etre resolue avec les exceptions comme :

- **Erreurs de Syntaxe (Syntax Errors)**
- **Erreurs de Type (Type Errors)**
- **Exception : TypeError**
- **Erreurs de Valeur (Value Errors)**
- **Exception : ValueError**
- **Erreurs d'Index (Index Errors)**
- **Exception : IndexError**
- **Erreurs de Clé (Key Errors)**
- **Exception : KeyError**
- **Erreurs d'Attribut (Attribute Errors)**
- **Exception : AttributeError**
- **Erreurs de Division (Division Errors)**
- **Exception : ZeroDivisionError**
- **Erreurs d'Entrée/Sortie (I/O Errors)**
- **Exception : IOError ou OSError**
- **Erreurs d'Importation (Import Errors)**
- **Exception : ImportError ou ModuleNotFoundError**
- **Erreurs de Mémoire (Memory Errors)**
- **Exception : MemoryError**
- **Erreurs d'Arrêt du Programme (SystemExit)**
- **Exception : SystemExit**
- **Erreurs de Clavier (Keyboard Interrupt)**
- **Exception : KeyboardInterrupt**
- **Erreurs de Protocole (EOFError)**
- **Exception : EOFError**
- **Erreurs de Connexion (Connection Errors)**
- **Exception : ConnectionError**

3)TYPES DES EXCEPTIONS :

Exceptions de syntaxe	se produisent lorsque le code source ne respecte pas la syntaxe du langage de programmation utilisé.{une parenthèse manquante, d'une virgule incorrecte ou d'un mot-clé mal orthographié}
Exceptions d'exécution	Se produisent pendant l'exécution du programme en raison de conditions imprévues ou de situations anormales. {Une division par zéro}
Exceptions de logique	se produisent lorsque le programme ne fonctionne pas d'une mauvaise logique de programmation . {une boucle infinie qui ne se termine jamais ou un algorithme qui produit des résultats incorrects}

Exceptions d'entrée/sortie (E/S)	Lorsqu'il y a un problème avec les opérations d'entrée/sortie, telles que la lecture ou l'écriture de fichiers, l'accès à des périphériques, etc.
Exceptions réseau	Se produisent lorsqu'il y a un problème avec les opérations réseau, telles que l'établissement d'une connexion, l'envoi ou la réception de données sur un réseau.
Exceptions de temps d'exécution	Se produisent lorsqu'une erreur se produit pendant l'exécution du programme, mais qu'elle n'est pas liée à une erreur syntaxique ou logique spécifique.
Exceptions personnalisées	Ces exceptions sont définies par le développeur pour gérer des situations spécifiques dans son application. Les exceptions personnalisées peuvent être utilisées pour signaler des erreurs métier, des conditions spéciales ou des situations exceptionnelles qui ne sont pas couvertes par les exceptions standard du langage.

4)Gestion des exceptions :

- La gestion d'exceptions est une pratique essentielle qui permet de gérer les erreurs et les situations imprévues de manière élégante et robuste, ainsi que la manière de gérer les erreurs lors de l'exécution d'un programme. En programmation orienté objet, les classes et les objets sont des composants clés, et la gestion des exceptions permet de gérer les erreurs de manière spécifique à chaque classe ou méthode. En utilisant des blocs try-except, les développeurs peuvent intercepter les erreurs et mettre en œuvre des actions de secours pour maintenir la stabilité du programme.

-En Python, une exception est une anomalie qui se produit pendant l'exécution d'un programme. Les exceptions peuvent être déclenchées lorsqu'une erreur se produit, comme une division par zéro, un accès à un index inexistant dans une liste, ou lorsqu'une erreur système survient, comme une impossibilité d'ouvrir un fichier.

La Structure de la Gestion d'Exceptions :

La gestion d'exceptions repose sur le mécanisme des blocs try, except, finally, et raise.

try: Ce bloc contient le code où des exceptions peuvent être levées. Il est suivi d'un ou plusieurs blocs except ou d'un bloc finally.

except: Ces blocs capturent les exceptions spécifiques levées dans le bloc try et spécifient comment le programme doit réagir à ces exceptions.

finally: Ce bloc est utilisé pour exécuter du code quel que soit le résultat de l'exécution du bloc try. Il est souvent utilisé pour nettoyer les ressources ou effectuer des actions de clôture.

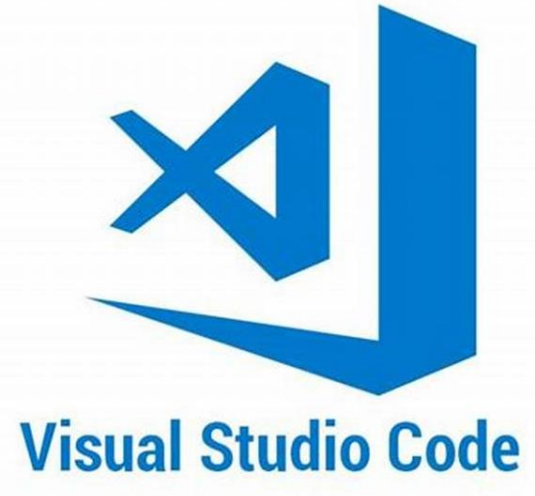
raise: Ce mot-clé permet de lever explicitement une exception à un endroit spécifié dans le code.

5)EXEMPLE :

```
try :
     a = int(input("entrez un nombre entier:"))
     b = 6/a
     print("la division est : ",b)

     '''   le résultat est :
     entrez un nombre entier:2
     la division est :  3.0    '''



except ValueError :
    print("je vous demande de saisir un entier ")

    '''   le résultat est :
    entrez un nombre entier:'bonjour'
    je vous demande de saisir un entier '''


    '''   le résultat est :
    entrez un nombre entier:2.5
    je vous demande de saisir un entier '''




except ZeroDivisionError :
    print("impossible de diviser par zero")

    '''   entrez un nombre entier:0
        impossible de diviser par zero'''


except :
    print("erreur!")

else :
    print("ça marche bien !")

finally:
    print("fin du programme")
```

6-CONCLUSION :

La gestion des exceptions en Python, combinée à la programmation orientée objet, offre un moyen puissant de gérer les erreurs et les situations inattendues dans les programmes. En utilisant les blocs try-except de manière appropriée, les développeurs peuvent écrire des applications robustes et résilientes.



Merci Pour Votre Attention

FJN