FT 013
Curso: UFCD 10793
UFCD/Módulo/Temática: UFCD 10793 - **Fundamentos de Python**
Ação: 10793_07/N
Formador/a: Sandra Liliana Meira de Oliveira
Data:
Nome do Formando/a:

Podemos capturar exceções específicas usando classes apropriadas, caso saibamos de antemão o tipo de erro que pode vir a ocorrer no script, como por exemplo erros de tipo incorreto ou a clássica divisão por zero.

Assim, podemos dar um tratamento adequado a cada caso de erro que ocorra.

A cláusula **else** permite executar um bloco de código caso o código dentro do bloco **try** tenha sido executado sem erros (ou seja, sem lançar nenhuma exceção).

Todo código inserido em uma cláusula **finally** é executado independentemente de ter havido ou não erros na execução do bloco **try**. **Geralmente, colocamos aqui código para libertar recursos externos, como conexões a bases de dados, acesso a recursos de rede ou um ficheiro aberto.**

1. Reproduz o seguinte código:

```
# Bloco finally
import sys
item = 22
try:
    divisao = 1/item
    print("Valor:", item)
except (TypeError):
    # Manipular tipo incorreto
    print("Você deve digitar apenas números!")
    print()
except (ZeroDivisionError):
    # Manipular divisão por zero
    print("Não é possível dividir por zero.")
    print()
except:
    # Manipular outras exceções
    print("Ocorreu a exceção", sys.exc_info()[0])
else:
    print("1 dividido por", item, "é", divisao)
finally:
    print("O valor fornecido foi:", item)
```

```
Valor: 22
1 dividido por 22 é 0.045454545454545456
O valor fornecido foi: 22
Não é possível dividir por zero.
O valor fornecido foi: 0
Você deve digitar apenas números!
O valor fornecido foi: Fábio dos Reis
```

2. Escreva um programa Python para manipular uma exceção **ZeroDivisionError** ao dividir um número por zero.

exception ZeroDivisionError:

Gerado quando o segundo argumento de uma operação de divisão ou módulo é zero. O valor associado é uma cadeia de caracteres que indica o tipo de operandos e a operação.

Python Code:

```python
def divide_numbers(x, y):
    try:
        result = x / y
        print("Result:", result)
    except ZeroDivisionError:
        print("The division by zero operation is not allowed.")
# Usage
numerator = 100
denominator = 0
divide_numbers(numerator, denominator)
```

Output:

```
The division by zero operation is not allowed.
```

3. Escreva um programa Python que solicita que o usuário insira um inteiro e gera uma exceção ValueError se a entrada não for um inteiro válido.

exception ValueError:

Gerado quando uma operação ou função recebe um argumento que tem o tipo certo, mas um valor inadequado, e a situação não é descrita por uma exceção mais precisa, como IndexError.

Python Code:

```
def get_integer_input(prompt):
        try:
            value = int(input(prompt))
            return value
        except ValueError:
            print("Error: Invalid input, input a valid integer.")
# Usage
n = get_integer_input("Input an integer: ")
print("Input value:", n)
```

Output:

```
Input an integer: abc
Error: Invalid input, input a valid integer.
Input value: None
```

```
Input an integer: 10.06
Error: Invalid input, input a valid integer.
Input value: None
```

```
Input an integer: 5
Input value: 5
```

4. Escreva um programa Python que abra um ficheiro e manipule uma exceção FileNotFoundError se o arquivo não existir.

exception FileNotFoundError:

Gerado quando um arquivo ou pasta é solicitada, mas não existe.

Python Code:

```
def open_file(filename):
    try:
        file = open(filename, 'r')
        contents = file.read()
        print("File contents:")
        print(contents)
        file.close()
    except FileNotFoundError:
        print("Error: File not found.")
# Usage
file_name = input("Input a file name: ")
open_file(file_name)
```

5. Escreva um programa Python que solicita ao utilizador dois números e gera uma exceção TypeError se as entradas não forem numéricas.

exception TypeError:

Gerado quando uma operação ou função é aplicada a um objeto de tipo inadequado. O valor associado é uma cadeia de caracteres que fornece detalhes sobre a incompatibilidade de tipo.

Python Code:

```python
def get_numeric_input(prompt):
    while True:
        try:
            value = float(input(prompt))
            return value
        except ValueError:
            print("Error: Invalid input. Please Input a valid number.")
# Usage
n1 = get_numeric_input("Input the first number: ")
n2 = get_numeric_input("Input the second number: ")
result = n1 * n2
print("Product of the said two numbers:", result)
```

Output:

```
Input the first number: a
Error: Invalid input. Please Input a valid number.
Input the first number: 10
Input the second number: b
Error: Invalid input. Please Input a valid number.
Input the second number: 12
Product of the said two numbers: 120.0
```

6. Escreva um programa Python que abra um ficheiro e manipule uma exceção PermissionError se houver um problema de permissão.

exception PermissionError:

Gerado ao tentar executar uma operação sem os direitos de acesso adequados - por exemplo, permissões do sistema de ficheiros.

Python Code:

```
def open_file(filename):
    try:
        with open(filename, 'w') as file:
            contents = file.read()
            print("File contents:")
            print(contents)
    except PermissionError:
        print("Error: Permission denied to open the file.")

# Usage
file_name = input("Input a file name: ")
open_file(file_name)
```

Output:

```
Enter the file name: employees.csv
Error: Permission denied to open the file.
```

7. Write a Python program that executes an operation on a list and handles an IndexError exception if the index is out of range.

exception IndexError:

Python Code:

```
def test_index(data, index):
    try:
        result = data[index]
        # Perform desired operation using the result
        print("Result:", result)
    except IndexError:
        print("Error: Index out of range.")


nums = [1, 2, 3, 4, 5, 6, 7]
index = int(input("Input the index: "))
test_index(nums, index)
```

Output:

```
Input the index: 0
Result: 1
```

```
Input the index: 4
Result: 5
```

```
Input the index: 9
Error: Index out of range.
```