Tratamento de Exceções

Exceção é um evento que ocorre durante a **execução de um programa** e que **interrompe o fluxo normal** de instruções, por exemplo:

- Divisão por zero
- Operação matemática inválida
- Tentativa de acesso a uma posição inválida em um vetor
- Tentativa de acesso a um objeto que não foi criado
- etc.

Tratamento de Exceções

Quando uma exceção ocorre dentro de um método, é criado um objeto que é entregue para o ambiente de execução

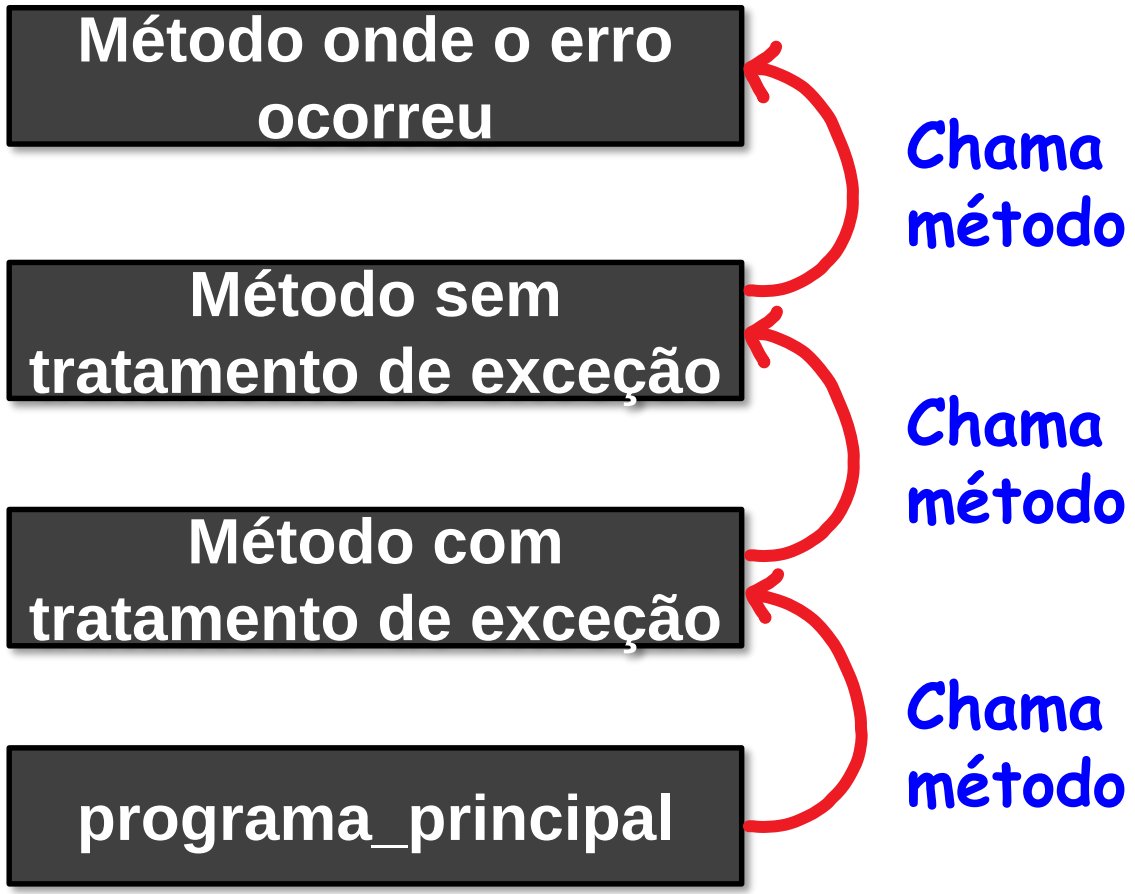
- Este comportamento é denominado levantar ("**raise**") uma exceção
- O objeto criado é chamado de **objeto de exceção** e contém informações sobre a exceção, incluindo seu tipo e o estado do programa quando a exceção ocorreu

https://docs.python.org/3/library/exceptions.html

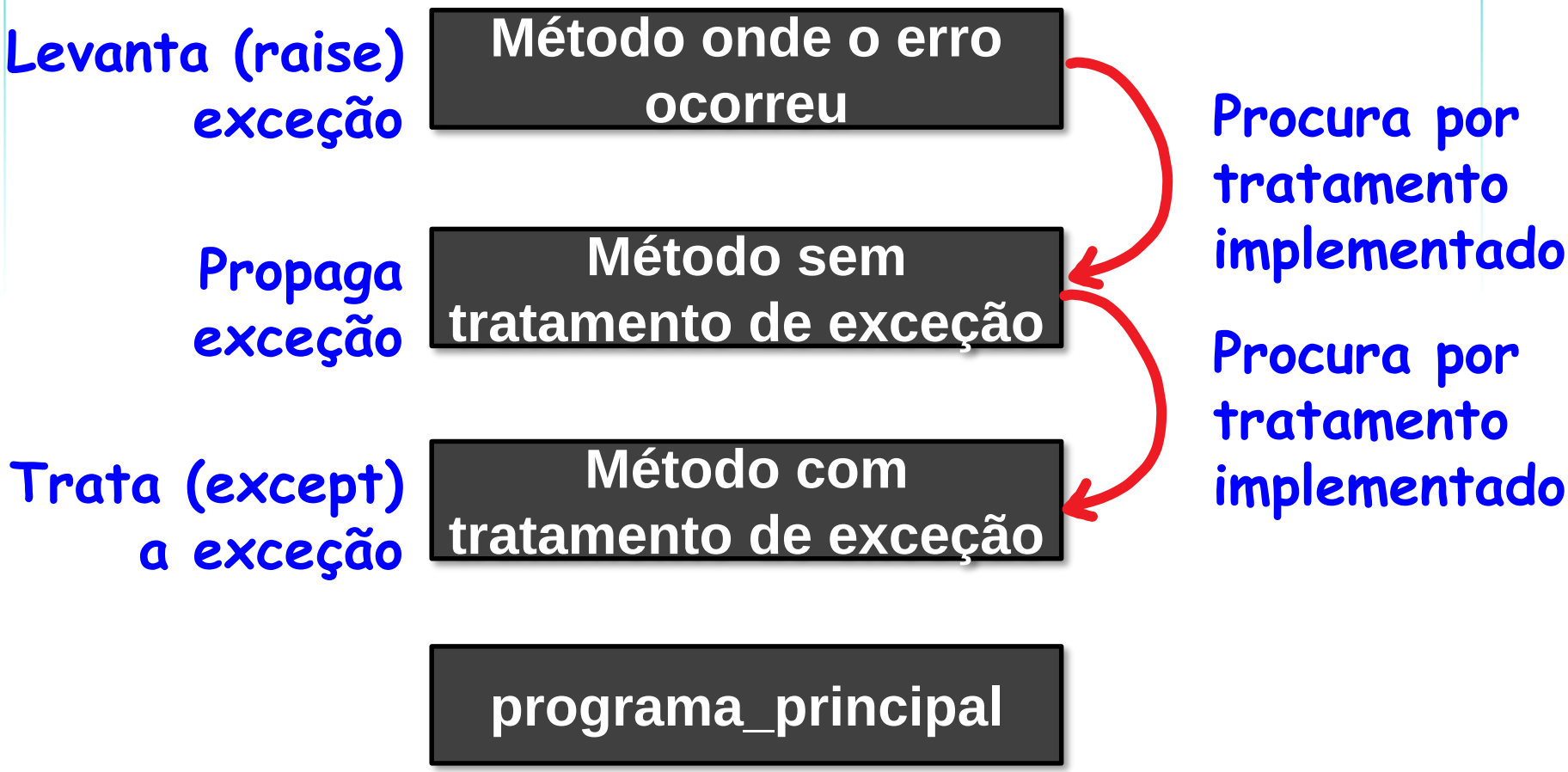
Tratamento de Exceções

- Depois que um método levanta (*raise*) uma exceção, o ambiente de execução tenta encontrar algum tratamento para a exceção
- A sequência de possíveis tratamentos para manipular a exceção é a pilha dos métodos que foram chamados para chegar até o método onde o erro ocorreu
- Antes de uma cláusula de exceção ser executada, os detalhes sobre a exceção são armazenados no módulo *sys* e podem ser acessados via *sys.exc_info()*

Tratamento de Exceções



Tratamento de Exceções

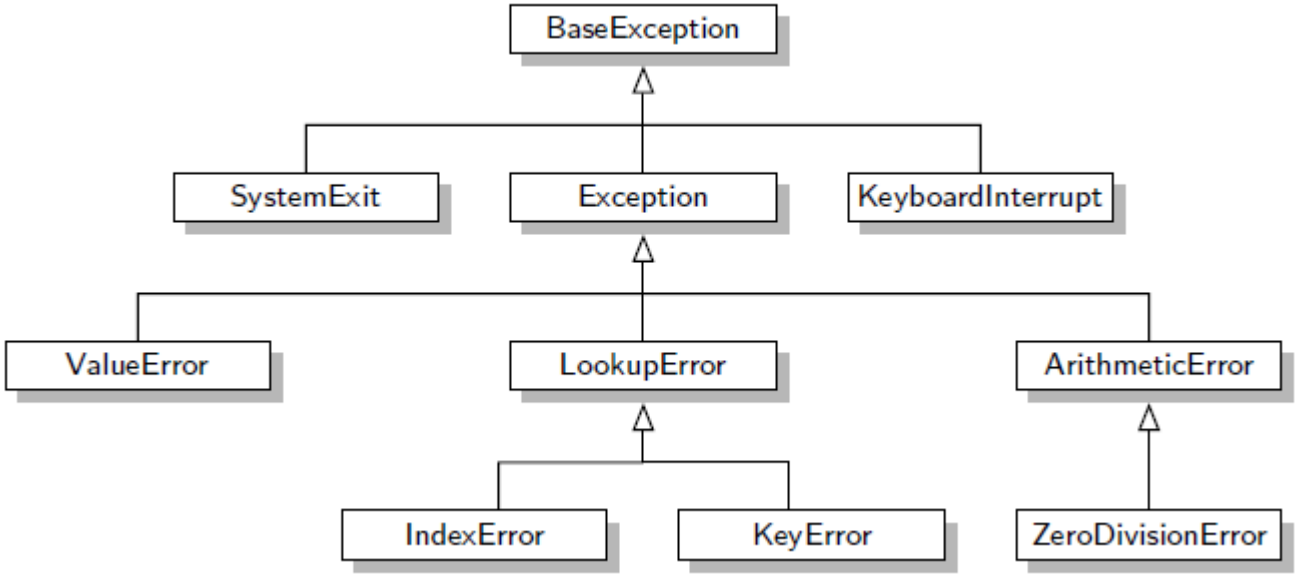


try ... except

```
try:
    1/0
except Exception:
    print("Erro!")
else:
    print("Nao deu erro!")
finally:
    print("Sempre executa!")
```

1. O código controlado pela cláusula ***try*** é executado
2. Se **ocorrer uma exceção**, o controle é desviado para a cláusula ***except***
3. Se a exceção ocorrida **estiver sendo tratada** pela cláusula ***except***, o código de tratamento da exceção é executado.
4. Se **não ocorrer uma exceção**, o código da clausula ***else*** é executado logo após o código do *try* e o tratamento da exceção **não é executado**
5. Ocorrendo ou não exceção, o código dentro da cláusula ***finally*** é **sempre executado**

Tipos de Exceções em Python

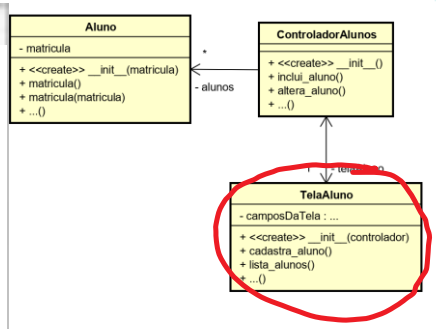


https://docs.python.org/3/library/exceptions.html

[http://www.devinline.com/]

Continuando com o Código da Tela

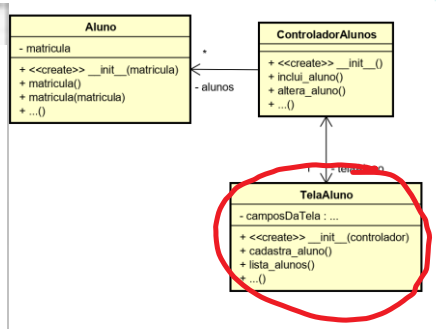
```
class TelaAluno:

    def __init__(self, controlador):
        self.__controlador = controlador

    def le_num_inteiro(self, mensagem: str = "", inteiros_validos: [] = None):
        while True:
            valor_lido = input(mensagem)
            try:
                inteiro = int(valor_lido)
                if inteiros_validos and inteiro not in inteiros_validos:
                    raise ValueError
                return inteiro
            except ValueError:
                print("Valor incorreto: Digite um valor num
                if inteiros_validos:
                    print("Valores validos: ", inteiros_valid

    def mostra_tela_opcoes(self):
        print("--------- CADASTRO ALUNOS ---------")
        print("1 - Incluir")
        print("2 - Alterar")
        print("3 - Excluir")
        print("4 - Listar")
        print("0 - Voltar")
        opcao = self.le_num_inteiro("Escolha a opcao: ", [1, 2, 3, 4, 0])
        return opcao
```



Também é possível disparar uma Exceção intencionalmente

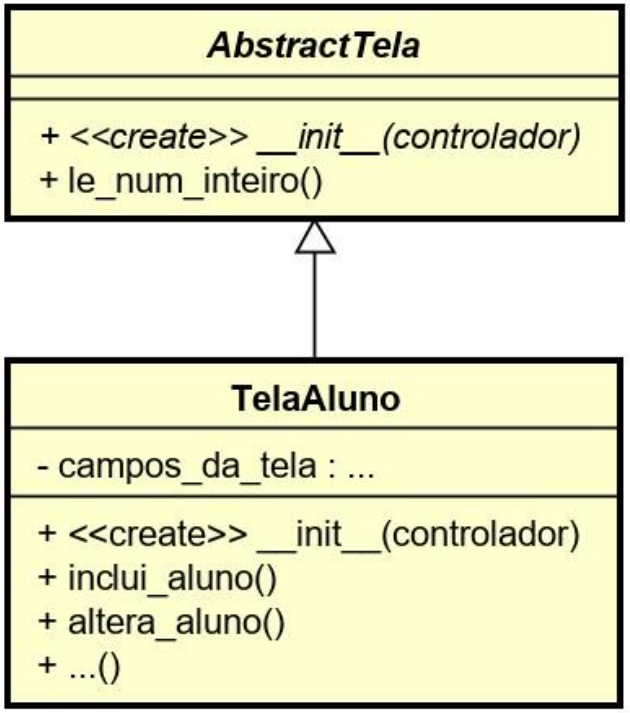
Continuando com o Código da Tela

```
class TelaAluno:

    def __init__(self, controlador):
        self.__controlador = controlador

    def le_num_inteiro(self, mensagem: str = "", inteiros_validos: [] = None):
        while True:
            valor_lido = input(mensagem)
            try:
                inteiro = int(valor_lido)
                if inteiros_validos and inteiro not in inteiros_validos:
                    raise ValueError
                return inteiro
            except ValueError:
                print("Valor incorreto: Digite um valor numerico inteiro valido")
                if inteiros_validos:
                    print("Valores validos: ", inteiros_validos)

    def mostra_tela_opcoes(self):
        print("--------- CADASTRO ALUNOS ---------")
        print("1 - Incluir")
        print("2 - Alterar")
        print("3 - Excluir")
        print("4 - Listar")
        print("0 - Voltar")
        opcao = self.le_num_inteiro("Escolha a opcao: ", [1, 2, 3, 4, 0])
        return opcao
```



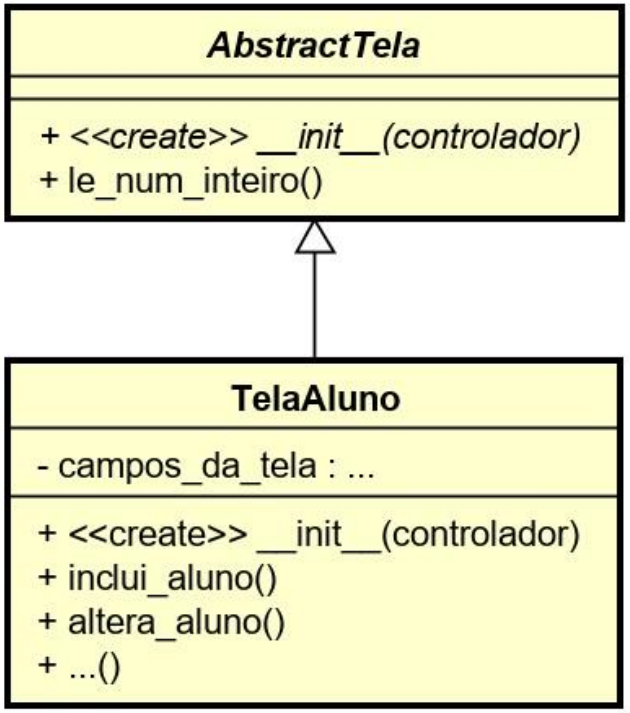
Este código será utilizado por outras telas?

Herança utilizada para Telas



Que tal então uma hierarquia de telas?

Herança utilizada para Telas



Que tal então uma hierarquia de telas?

Pode ser interessante também nos Controladores que tenham características comuns entre eles!

Agradecimento

Agradecimento ao prof. Marcello Thiry pelo material cedido.

marcello.thiry@gmail.com

Licença

Atribuição-Uso-Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

Você pode:
- copiar, distribuir, exibir e executar a obra
- criar obras derivadas

Sob as seguintes condições:
Atribuição — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.
Uso Não-Comercial — Você não pode utilizar esta obra com finalidades comerciais.
Compartilhamento pela mesma Licença — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite http://creativecommons.org/licenses/by-nc-sa/2.5/br/ ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.