

**Disciplina: POO**

**Professor: Genivan Silva**

**Discente: Marcos Antonio**

**Atividade 03 – Tratamento de exceções**

* Para responder a atividade utilize todos os recursos de orientação objetos vistos até o momento: herança, polimorfismo e encapsulamento.
* Em cada método efetue o tratamento de exceções de modo a diminuir a probabilidade de os cálculos serem realizados com erros.
  1. Crie uma classe chamada Forma Geométrica com um atributo área e o método calcula área. Esse método não executa nenhuma ação, possui apenas uma declaração e seu corpo pode ficar vazio.
  2. Crie três subclasses da classe Forma Geométrica: triângulo, círculo e trapézio. Cada uma dessas classes deve possuir atributos específicos para sua forma. Exemplo: o triângulo deve possuir uma base e uma altura, o círculo possui o raio e uma outra variável com o valor de PI, já o trapézio possui base menor, base maior e altura.
  3. Em cada uma das subclasses implemente o método calcula área, porém, cada uma das formas possui uma forma de implementação diferente. Pesquise como efetua o cálculo de área para as três formas.
  4. Dentro do mesmo arquivo teste a instanciação dos objetos e a execução dos métodos. Sua aplicação deve iniciar perguntando qual das formas o usuário deseja efetuar os cálculos. Em seguida o usuário informa os valores específicos para a forma escolhida e o programa apresenta o resultado do cálculo da área. Caso o usuário informe valores inválidos para qualquer um dos parâmetros, a aplicação precisa informar o erro e solicitar novo valor (use o próprio tratamento de exceções para isso).

class FormaGeometrica:

    def \_\_init\_\_(*self*):

*self*.\_area = 0

    @property

    def area(*self*):

        return *self*.\_area

    @area.setter

    def area(*self*, *area*):

*self*.\_area = *area*

class Tringulo(FormaGeometrica):

    def \_\_init\_\_(*self*, *base*, *altura*):

        super().\_\_init\_\_()

*self*.\_base = *base*

*self*.\_altura = *altura*

    @property

    def base(*self*):

        return *self*.\_base

    @property

    def altura(*self*):

        return *self*.\_altura

    def calculaArea(*self*):

*self*.area = (*self*.base \* *self*.altura)/2

class Circulo(FormaGeometrica):

    def \_\_init\_\_(*self*, *raio*):

        super().\_\_init\_\_()

*self*.\_raio = *raio*

*self*.\_pi = 3.14

    @property

    def raio(*self*):

        return *self*.\_raio

    @property

    def pi(*self*):

        return *self*.\_pi

    def calculaArea(*self*):

*self*.area = (*self*.raio\*\*2) \* *self*.pi

class Trapezio(FormaGeometrica):

    def \_\_init\_\_(*self*, *bMaior*, *bMenor*, *altura*):

        super().\_\_init\_\_()

*self*.\_bMaior = *bMaior*

*self*.\_bMenor = *bMenor*

*self*.\_altura = *altura*

    @property

    def bMaior(*self*):

        return *self*.\_bMaior

    @property

    def bMenor(*self*):

        return *self*.\_bMenor

    @property

    def altura(*self*):

        return *self*.\_altura

    def calculaArea(*self*):

*self*.area = ((*self*.bMaior + *self*.bMenor) \* *self*.altura)/2

def exibeMenu():

    print('################# MENU #################\n'

          '1 - Calcular área do Triangulo\n'

          '2 - Calcular área do Circulo\n'

          '3 - Calcular área do Trapezio\n'

          '4 - SAIR\n')

    try:

        opcao = int(input('Escolha uma opção (Digite o numero):'))

        return opcao

    except ValueError:

        print('Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

    except BaseException:

        print('EITA VOCÊ CAIU NO ERRO GENERICO')

opcao = -1

while opcao != 4:

    opcao = exibeMenu()

    if opcao == 4:

        break

    elif opcao == 1:

        while True:

            try:

                base = float(input('Digite o valor da base:'))

                while True:

                    try:

                        altura = float(input('Digite o valor da altura:'))

                        triangulo = Tringulo(base, altura)

                        triangulo.calculaArea()

                        print(f'BASE: {triangulo.base}\n'

                              f'ALTURA: {triangulo.altura}\n'

                              f'AREA: {triangulo.area}\n'

                              '###################################')

                        opcao == 0

                    except ValueError:

                        print('Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

                    except BaseException:

                        print('EITA VOCÊ CAIU NO ERRO GENERICO')

                    finally:

                        break

            except ValueError:

                print('Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

            except BaseException:

                print('EITA VOCÊ CAIU NO ERRO GENERICO')

            finally:

                print('REDIRECIONANDO PARA O MENU')

                break

    elif opcao == 2:

        while True:

            try:

                raio = float(input('Digite o valor do raio:'))

                circulo = Circulo(raio)

                circulo.calculaArea()

                print(f'RAIO: {circulo.raio}\n'

                      f'AREA: {circulo.area}\n'

                      '###################################\n')

                opcao == 0

            except ValueError:

                print('Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

            except BaseException:

                print('EITA VOCÊ CAIU NO ERRO GENERICO')

            finally:

                print('REDIRECIONANDO PARA O MENU')

                break

    elif opcao == 3:

        while True:

            try:

                bMaior = float(input('Digite o valor da base maior:'))

                while True:

                    try:

                        bMenor = float(input('Digite o valor da base menor:'))

                        while True:

                            try:

                                altura = float(input('Digite o valor da altura:'))

                                trapezio = Trapezio(bMaior, bMenor, altura)

                                trapezio.calculaArea()

                                print(f'BASE MAIOR: {trapezio.bMaior}\n'

                                    f'BASE MENOR: {trapezio.bMenor}\n'

                                    f'ALTURA: {trapezio.altura}\n'

                                    f'AREA: {trapezio.area}\n'

                                    '###################################\n')

                                opcao == 0

                            except ValueError:

                                print(

                                    'Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

                            except BaseException:

                                print('EITA VOCÊ CAIU NO ERRO GENERICO')

                            finally:

                                break

                    except ValueError:

                        print(

                            'Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

                    except BaseException:

                        print('EITA VOCÊ CAIU NO ERRO GENERICO')

                    finally:

                        break

            except ValueError:

                print('Parece que você digitou uma letra, DIGITE SOMENTE NUMERO!!!')

            except BaseException:

                print('EITA VOCÊ CAIU NO ERRO GENERICO')

            finally:

                print('REDIRECIONANDO PARA O MENU')

                break

    else:

        print('Digite uma opção valida')

        opcao == 0