

Exercicio 1:

Crie um objeto a partir da classe abaixo, chamado roc1, passando 2 parâmetros e depois faça uma chamada aos atributos e métodos

```
In [ ]: from math import sqrt

class Rocket():

    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def move_rocket(self, x_increment=0, y_increment=1):
        self.x += x_increment
        self.y += y_increment

    def print_rocket(self):
        print(self.x, self.y)

# coloque seu código aqui

roc1 = Rocket(3,5)
roc1.move_rocket(2,3)
roc1.print_rocket()
```

5 8

Exercicio 2:

Crie uma classe chamada Pessoa() com os atributos: nome, cidade, telefone e e-mail. Use pelo menos 2 métodos especiais na sua classe, onde faça uma apresentação de si mesmo e informe seus contatos. Crie um objeto da sua classe e faça uma chamada a pelo menos um dos seus métodos especiais.

```
In [ ]: # coloque seu código aqui
class pessoa():
    def __init__(self, _nome, _cidade, _telefone, _email): #metodo construtor
        self.nome = _nome
        self.cidade = _cidade
        self.telefone = _telefone
        self.email = _email

    def apresentar(self):
        print(f"olá, eu sou {self.nome}. moro em {self.cidade}. meu telefone de cor

gabriel = pessoa("gabriel", "diadema", 11999995555, "gabriel@gmail.com")
gabriel.apresentar()
```

olá, eu sou gabriel. moro em diadema. meu telefone de contato e 11999995555. e meu outro meio de contato e gabriel@gmail.com

Exercicio 3:

Crie a classe Smartphone com 2 atributos, tamanho e interface e crie a classe MP3Player com os atributos capacidade. A classe MP3player deve herdar os atributos da classe Smartphone.

```
In [ ]: # coloque seu código
class smartphone():
    def __init__(self, tamanho, interface):
        self.tamanho = tamanho
        self.interface = interface
    def apresentar(self):
        print(f"meu smartphone e {self.interface}.tem {self.tamanho}. gb")

class mp3plaer(smartphone):
    def __init__(self, tamanho, interface):
        super().__init__(tamanho, interface)
memoria = smartphone(512, "celular")
memoria.apresentar()
```

meu smartphone e celular.tem 512. gb

Exercicio 4:

Crie uma classe Livro que armazene o titulo, o isbn (código de armazenamento da Biblioteca), que ao iniciar imprima que foi criado o código com sucesso, e uma função informacoes para mostrar os dados do objeto. Crie dois objetos e faça a chamada do mesmo.

```
In [ ]: # coloque seu código aqui
class livro():
    def __init__(self, _titulo, _isbn):
        self.titulo = _titulo
        self.isbn = _isbn
        print(f"livro criado. titulo: {self.titulo}, isbn: {self.isbn}")
    def informacoes(self):
        print(f"titulo: {self.titulo}, isbn: {self.isbn}")
livro1 = livro("diario de um banana", "978-0-14-377843-1")
livro2 = livro("calvin e haroldo", "978-8-57-616521-7")

livro1.informacoes()
livro2.informacoes()
```

livro criado. titulo: diario de um banana, isbn: 978-0-14-377843-1
 livro criado. titulo: calvin e haroldo, isbn: 978-8-57-616521-7
 titulo: diario de um banana, isbn: 978-0-14-377843-1
 titulo: calvin e haroldo, isbn: 978-8-57-616521-7

Exercicio 5:

Crie uma classe chamada Veiculo, e 3 subclasses chamada Moto, Carro, Avião. As classes devem ter um método chamado Mover que deve imprimir "Correndo", exceto a classe Avião que deverá imprimir "Voando"

```
In [ ]: # coloque seu código aqui
class veiculo():
    def mover(self):
        print("corredor")
class moto(veiculo):
    pass
class carro(veiculo):
    pass
class aviao(veiculo):
    def mover(self):
        print("voando")
```

```

moto = moto()
carro = carro()
aviao = aviao ()
moto.mover()
carro.mover()
aviao.mover()

```

corredor
corredor
voando

Exercicio 6

Crie uma classe Pessoa, e duas classes Estudante e Funcionario que se estendem da classe Pessoa.

A classe pessoa deve conter nome, idade, e altura.

A classe Estudante, alem das originais deve conter Curso e Ano, uma função que retorna tempo para concluir (cursos de 5 anos) e a idade que deverá ser concluída.

A Classe funcionario deve conter alem das originais, deverá conter salário e cargo.

Crie dois estudantes fictícios e 3 funcionarios.

```

In [ ]: # coloque seu código aqui
class Pessoa():
    def __init__(self, nome, idade, altura):
        self.nome = nome
        self.idade = idade
        self.altura = altura

class Estudante (Pessoa):
    def __init__(self, nome, idade, altura, curso, ano):
        super().__init__(nome, idade, altura)
        self.curso = curso
        self.ano = ano
    def tempo_para_concluir(self):
        anos_restantes = 5 - self.ano
        return(f"Tempo para concluir: {anos_restantes} anos")
    def idade_conclusao(self):
        idade_conclusao = self.idade + (5- self.ano)
        return (f"Idade de conclusão: {idade_conclusao} anos")
    def aprenhaentar(self):
        print(f"ola eu sou {self.nome}. tenho {self.idade}. tenho {self.altura}. faço")

class Funcionario(Pessoa):
    def __init__(self, nome, idade, altura, salario, cargo):
        super().__init__(nome, idade, altura)
        self.salario = salario
        self.cargo = cargo
    def aprenhaentar(self):
        print(f"ola eu sou {self.nome}. tenho {self.idade}. tenho {self.altura}. ganho")

Estudante1 = Estudante("João", 20, 1.75, "Engenharia", 2)
Estudante2 = Estudante("Maria", 22, 1.68, "Medicina", 3)
Funcionario1 = Funcionario("Carlos", 30, 1.81, 5000, "Analista")
Funcionario2 = Funcionario("Ana", 28, 1.71, 6000, "Gerente")
Funcionario3 = Funcionario("Paulo", 35, 1.75, 4500, "Desenvolvedor")

```

```

Estudante1.aprenhaentar()
Estudante2.aprenhaentar()
Funcionario1.aprenhaentar()
Funcionario2.aprenhaentar()
Funcionario3.aprenhaentar()

```

ola eu sou João. tenho 20. tenho 1.75. faço curso de Engenharia. e estou fazendo 2 ano

ola eu sou Maria. tenho 22. tenho 1.68. faço curso de Medicina. e estou fazendo 3 ano

ola eu sou Carlos. tenho 30. tenho 1.81. ganho 5000. meu cargo e Analista

ola eu sou Ana. tenho 28. tenho 1.71. ganho 6000. meu cargo e Gerente

ola eu sou Paulo. tenho 35. tenho 1.75. ganho 4500. meu cargo e Desenvolvedor

Exercicio 7

Crie uma classe Circulo, que permita alterar o seu valor de raio, mesmo após ser inicializado. Deve haver função para calcular sua área e perímetro.

```

In [ ]: # coloque seu código aqui
class Circulo:
    def __init__(self, raio):
        self.raio = raio
    def alterar_raio (self, novo_raio):
        self.raio = novo_raio
    def calcular_area(self):
        area = 3.14 * (self.raio ** 2)
        return area
    def calcular_perimetro(self):
        perimetro = 2 * 3.14* self.raio
        return perimetro
# Exemplo de uso
circulo = Circulo(5)
print("Área:", circulo.calcular_area())
print("Perímetro:",
circulo.calcular_perimetro() )
circulo.alterar_raio(8)
print("Novo Raio:", circulo.raio)
print("Nova Área:", circulo.calcular_area())
print("Novo Perímetro:",
circulo.calcular_perimetro())

```

Área: 78.5

Perímetro: 31.400000000000002

Novo Raio: 8

Nova Área: 200.96

Novo Perímetro: 50.24