1- 6

```
from Veiculo import Veiculo

class Moto (Veiculo):
  def __init__(self, modelo, ano, cambio):
    self._modelo = modelo
    self._ano = ano
    self._cambio = cambio

  def andar(self):
    print(f"A moto {self._modelo} ano {self._ano} anda")

  def trocarMarcha(self):
    print(f"A moto {self._modelo} troca marcha de modo {self._cambio}")

  def acelerar(self, valor):
    self.velocidade += valor
    print(f"Velocidade atual: {self.velocidade} km/h")
class Veiculo():
  def __init__(self, tipo, velocidade):
    self._tipo = tipo
    self._velocidade = velocidade
  def descricao(self):
    print(f'Tipo: {self._tipo}')
    print(f'Velocidade: {self._velocidade}')

  def descricao(self):
      print(f"Este é um {self.tipo} e está na velocidade {self.velocidade}.")


veiculos = [Carro("Toyota", "Corolla", 2022),Moto("Honda", "CBR500R", 500),
Carro("Ford", "Mustang", 2022),Moto("Yamaha", "YZF-R1", 1000)]

for veiculo in veiculos:
  veiculo.descricao()
from Veiculo import Veiculo
class Carro (Veiculo):
  def __init__(self, modelo, ano, cambio):
    super().__init__(tipo = "Carro", velocidade = "100 km/h"  )
    self._modelo = modelo
    self._ano = ano
    self._cambio = cambio

  def andar(self):
    print(f" O {self._tipo}  {self._modelo} ano {self._ano} anda a {self._velocidade} ")

  def trocarMarcha(self):
    print(f"O {self._tipo}  {self._modelo} troca marcha de modo {self._cambio}")
```

```
  def __init__(self, marca, modelo, ano):
    super().__init__(marca, modelo)
    self.ano = ano

  def descricao(self):
    print(f"Este é um carro da marca {self.marca}, modelo {self.modelo} e ano
{self.ano}.")

  def acelerar(self, valor):
    self.velocidade += valor
    print(f"Velocidade atual: {self.velocidade} km/h")from Veiculo import Veiculo
from Carro import Carro
from Moto import Moto

carro1 = Carro("BMW", 2023, "automatico")
carro1.andar()
carro1.trocarMarcha()

def acelerar(veiculo, valor):
  veiculo.acelerar(valor)
```

7-

```
class Forma:
  def calcular_area(self):
      pass

class Retangulo(Forma):
  def __init__(self, comprimento, largura):
      self.comprimento = comprimento
      self.largura = largura

  def calcular_area(self):
      return self.comprimento * self.largura

class Circulo(Forma):
  def __init__(self, raio):
      self.raio = raio

  def calcular_area(self):
      import math
      return math.pi * self.raio ** 2

# Exemplo de uso:
retangulo = Retangulo(10, 20)
circulo = Circulo(6)

print(f"Área do retângulo: {retangulo.calcular_area()}")
print(f"Área do círculo: {circulo.calcular_area()}")
```

8-
```
class Imprimivel:
  def imprimir(self):
      pass

class Carro(Imprimivel):
  def __init__(self, marca, modelo, ano):
      self.marca = marca
      self.modelo = modelo
      self.ano = ano

  def imprimir(self):
      print(f"Carro: Marca - {self.marca}, Modelo - {self.modelo}, Ano -
{self.ano}")

class Moto(Imprimivel):
  def __init__(self, marca, modelo, cilindradas):
      self.marca = marca
      self.modelo = modelo
      self.cilindradas = cilindradas

  def imprimir(self):
      print(f"Moto: Marca - {self.marca}, Modelo - {self.modelo}, Cilindradas -
{self.cilindradas}")
```

9-

```
class Conta:
  def __init__(self, titular, saldo=0):
      self.titular = titular
      self.saldo = saldo

  def depositar(self, valor):
      if valor > 0:
          self.saldo += valor
          print(f'Depósito de R${valor} realizado. Novo saldo: R${self.saldo}')
      else:
          print("Valor inválido para depósito.")

  def sacar(self, valor):
      if 0 < valor <= self.saldo:
          self.saldo -= valor
          print(f'Saque de R${valor} realizado. Novo saldo: R${self.saldo}')
      else:
          print("Saldo insuficiente ou valor inválido para saque.")

  def consultar_saldo(self):
      return self.saldo

class ContaCorrente(Conta):
  def __init__(self, titular, saldo=0, limite=500):
```

```
        super().__init__(titular, saldo)
        self.limite = limite

    def sacar(self, valor):
        if valor <= self.saldo + self.limite:
            super().sacar(valor)
        else:
            print("Saldo e limite insuficientes para o saque.")

conta1 = Conta("ana", 1000)
conta2 = ContaCorrente("luisa", 5000, 1000)

print(f"Saldo de {conta1.titular}: R${conta1.consultar_saldo()}")
conta1.depositar(500)
conta1.sacar(300)
print(f"Saldo de {conta1.titular}: R${conta1.consultar_saldo()}")

print(f"Saldo de {conta2.titular}: R${conta2.consultar_saldo()}")
conta2.depositar(1000)
conta2.sacar(3000)
print(f"Saldo de {conta2.titular}: R${conta2.consultar_saldo()}")
```

10-

```
class Personagem:
    def __init__(self, nome, pontos_vida):
        self.nome = nome
        self.pontos_vida = pontos_vida

    def atacar(self, alvo):
        pass

class Guerreiro(Personagem):
    def __init__(self, nome):
        super().__init__(nome, pontos_vida=100)

    def atacar(self, alvo):
        dano = 20
        print(f"{self.nome} ataca {alvo.nome} com sua espada, causando {dano}
de dano.")
        alvo.pontos_vida -= dano

# Exemplo de uso:
heroi = Guerreiro("Sonic")
monstro = Personagem("Shadow the Hedgehog", pontos_vida=50)

heroi.atacar(monstro)
print(f"{monstro.nome} possui {monstro.pontos_vida} pontos de vida
restantes.")
```