class AgenteRobo:

def \_\_init\_\_(self):

self.energia = 100

self.bolsa = 0

self.localizacao = [0, 0]

self.objetivo\_alcancado = False

self.direcoes = {'Norte': (-1, 0), 'Sul': (1, 0), 'Leste': (0, 1), 'Oeste': (0, -1)}

self.ambiente = [['A', 'B', 'C', 'D'],

['E', 'F', 'G', 'H'],

['I', 'J', 'K', 'L'],

['M', 'N', 'O', 'P']]

def aspirar\_sujeira(self):

linha, coluna = self.localizacao

local\_atual = self.ambiente[linha][coluna]

self.ambiente[linha][coluna] = ' '

self.bolsa += 1

self.energia -= 1

print("Limpando sujeira")

def mover(self, direcao):

linha, coluna = self.localizacao

deslocamento = self.direcoes[direcao]

nova\_linha = linha + deslocamento[0]

nova\_coluna = coluna + deslocamento[1]

if self.validar\_localizacao(nova\_linha, nova\_coluna):

self.localizacao = [nova\_linha, nova\_coluna]

self.energia -= 1

def validar\_localizacao(self, linha, coluna):

return 0 <= linha < 4 and 0 <= coluna < 4

def voltar\_para\_casa(self):

while self.localizacao != [0, 0]:

self.mover(self.determinar\_direcao\_de\_volta())

def verificar\_objetivo(self):

if self.bolsa >= 10:

self.bolsa = 0

self.objetivo\_alcancado = True

def determinar\_direcao\_de\_volta(self):

linha, coluna = self.localizacao

if linha > 0:

return 'Norte'

elif coluna > 0:

return 'Oeste'

else:

return 'Sul'

def limpar\_quarto(self):

while not self.objetivo\_alcancado:

if self.energia <= 0:

print("Sem energia")

break

self.aspirar\_sujeira()

self.verificar\_objetivo()

if self.objetivo\_alcancado:

print("\nA Bolsa ficou cheia")

self.voltar\_para\_casa()

self.bolsa = 0

print("\nBolsa esvaziada")

print("\nLimpando novamente")

if not self.objetivo\_alcancado:

self.mover(self.determinar\_proxima\_acao())

print("\nO Quarto esta limpo!")

def determinar\_proxima\_acao(self):

import random

direcoes\_possiveis = list(self.direcoes.keys())

return random.choice(direcoes\_possiveis)

agente = AgenteRobo()

agente.limpar\_quarto()