class RedeSocial:

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

self.usuarios = {}

self.amizades = {}

self.publicacoes = []

def adicionar\_usuario(self, nome):

if nome not in self.usuarios:

self.usuarios[nome] = []

self.amizades[nome] = []

print(f"Usuário {nome} adicionado.")

else:

print(f"Usuário {nome} já existe.")

def adicionar\_amigo(self, usuario1, usuario2):

if usuario1 in self.usuarios and usuario2 in self.usuarios:

if usuario2 not in self.amizades[usuario1]:

self.amizades[usuario1].append(usuario2)

self.amizades[usuario2].append(usuario1)

print(f"{usuario1} e {usuario2} agora são amigos.")

else:

print(f"{usuario1} e {usuario2} já são amigos.")

else:

print("Um ou ambos os usuários não existem.")

def publicar\_mensagem(self, usuario, mensagem):

if usuario in self.usuarios:

self.publicacoes.append((usuario, mensagem))

print(f"{usuario} publicou: {mensagem}")

else:

print("Usuário não encontrado.")

def comentar\_post(self, usuario, mensagem, index):

if index < len(self.publicacoes):

print(f"{usuario} comentou: {mensagem} no post de {self.publicacoes[index][0]}")

else:

print("Post não encontrado.")

def buscar\_usuario(self, nome):

if nome in self.usuarios:

print(f"Usuário {nome} encontrado.")

else:

print(f"Usuário {nome} não encontrado.")

class Biblioteca:

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

self.livros = {}

self.emprestimos = {}

def cadastrar\_livro(self, titulo):

if titulo not in self.livros:

self.livros[titulo] = True # True significa disponível

print(f"Livro '{titulo}' cadastrado.")

else:

print(f"Livro '{titulo}' já cadastrado.")

def emprestar\_livro(self, titulo, usuario):

if titulo in self.livros and self.livros[titulo]:

self.livros[titulo] = False # Livro não disponível

self.emprestimos[titulo] = usuario

print(f"Livro '{titulo}' emprestado para {usuario}.")

else:

print(f"Livro '{titulo}' não disponível.")

def devolver\_livro(self, titulo):

if titulo in self.emprestimos:

self.livros[titulo] = True # Livro disponível novamente

del self.emprestimos[titulo]

print(f"Livro '{titulo}' devolvido.")

else:

print(f"Livro '{titulo}' não foi emprestado.")

def verificar\_disponibilidade(self, titulo):

if titulo in self.livros:

return "Disponível" if self.livros[titulo] else "Indisponível"

else:

return "Livro não encontrado."

import calendar

from datetime import datetime

class Calendario:

def exibir\_calendario(self, ano, mes):

print(calendar.month(ano, mes))

def verificar\_feriado(self, data):

feriados = {

"01-01": "Ano Novo",

"25-12": "Natal",

# Adicione mais feriados conforme necessário

}

data\_formatada = data.strftime("%d-%m")

return feriados.get(data\_formatada, "Não é feriado.")

def calcular\_diferenca\_dias(self, data1, data2):

delta = abs((data2 - data1).days)

return delta

import random

class JogoAdivinhacao:

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

self.numero\_secreto = random.randint(1, 100)

def palpite(self, palpite):

if palpite < self.numero\_secreto:

return "Seu palpite é menor que o número secreto."

elif palpite > self.numero\_secreto:

return "Seu palpite é maior que o número secreto."

else:

return "Parabéns! Você acertou!"

class Peca:

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

self.cor = cor

class Peao(Peca):

def mover(self):

return