МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №1**

по дисциплине: Системное моделирование

тема: «*Модель по экономике: Простая модель совокупного спроса и предложения*»

Выполнил: ст. группы ПВ-233

Ситников Алексей Павлович

Проверил:

Поляков Владимир Михайлович

Белгород 2025 г.

Начальный код:

import numpy as np  
import matplotlib.pyplot as plt  
  
  
def simulate\_ad\_as(C, I, G, X, M, Y\_star, k, P\_e, P, dt, T):  
 t = 0 # Начальное время  
 results = [] # Список для хранения результатов  
 print("Time\tPrice Level\tAD\tAS")  
  
 while t < T:  
 # Вычисление совокупного спроса  
 AD = C + I + G + (X - M)  
 # Вычисление совокупного предложения  
 AS = Y\_star + k \* (P - P\_e)  
 # Корректировка уровня цен  
 if AD > AS:  
 P += 0.01 \* (AD - AS) # Рост уровня цен  
 else:  
 P -= 0.01 \* (AS - AD) # Снижение уровня цен  
  
  
  
 # Сохранение результатов  
 results.append((t, P, AD, AS))  
 print(f"{t:.2f}\t{P:.2f}\t{AD:.2f}\t{AS:.2f}")  
 # Увеличение времени  
 t += dt  
 return results  
  
  
# Инициализация параметров  
C = 100 # Потребительские расходы  
I = 50 # Инвестиции  
G = 70 # Государственные расходы  
X = 30 # Экспорт  
M = 20 # Импорт  
Y\_star = 200 # Потенциальный выпуск  
k = 0.5 # Чувствительность предложения  
P\_e = 100 # Ожидаемый уровень цен  
P = 100 # Начальный уровень цен  
dt = 0.1 # Шаг времени  
T = 100 # Общее время моделирования  
# Запуск симуляции  
  
res = simulate\_ad\_as(C, I, G, X, M, Y\_star, k, P\_e, P, dt, T)  
  
aD = []  
aS = []  
time = []  
sum\_ = []  
t = 0  
for i in res:  
 time.append(t)  
 aD.append(i[2])  
 aS.append(i[3])  
 sum\_.append(i[1])  
 t+=dt  
  
plt.plot(time, aS, label='Кривая AS') # Предложение  
plt.plot(time, aD, label='Кривая AD') # Спрос  
plt.plot(time, sum\_, label='уровень цен') # Суммарная кривая  
plt.legend(loc='best')  
plt.show()

Вывод программы:

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Увеличили государственные расходы до 100, получили такой график:  
Изображение выглядит как текст, снимок экрана, линия, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Цена увеличилась.

Увеличили ожидаемое повышение цен до 120, получили:  
Изображение выглядит как текст, снимок экрана, линия, График

Контент, сгенерированный ИИ, может содержать ошибки.

Цена увеличилась ещё больше

Уменьшим инвестиции до 20, получим:  
Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Цена значительно уменьшилась.

Увеличим коэффициент чувствительности предложения к изменению цен до 1, получим:  
Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Цена упала, а графики AD и AS сходятся раньше