**Роевой искусственный интеллект**

**Задание 1:** Алгоритмы роя частиц (PSO) для оценки регулирующего эффекта нормативного акта.

Реализуйте алгоритм роя частиц на пайтоне для оценки воздействия нового нормативного акта на экономику. Предположим, что программа использует исторические данные для оценки различных экономических показателей и находить оптимальные параметры для анализа влияния акта на экономику.

Тестовые данные:

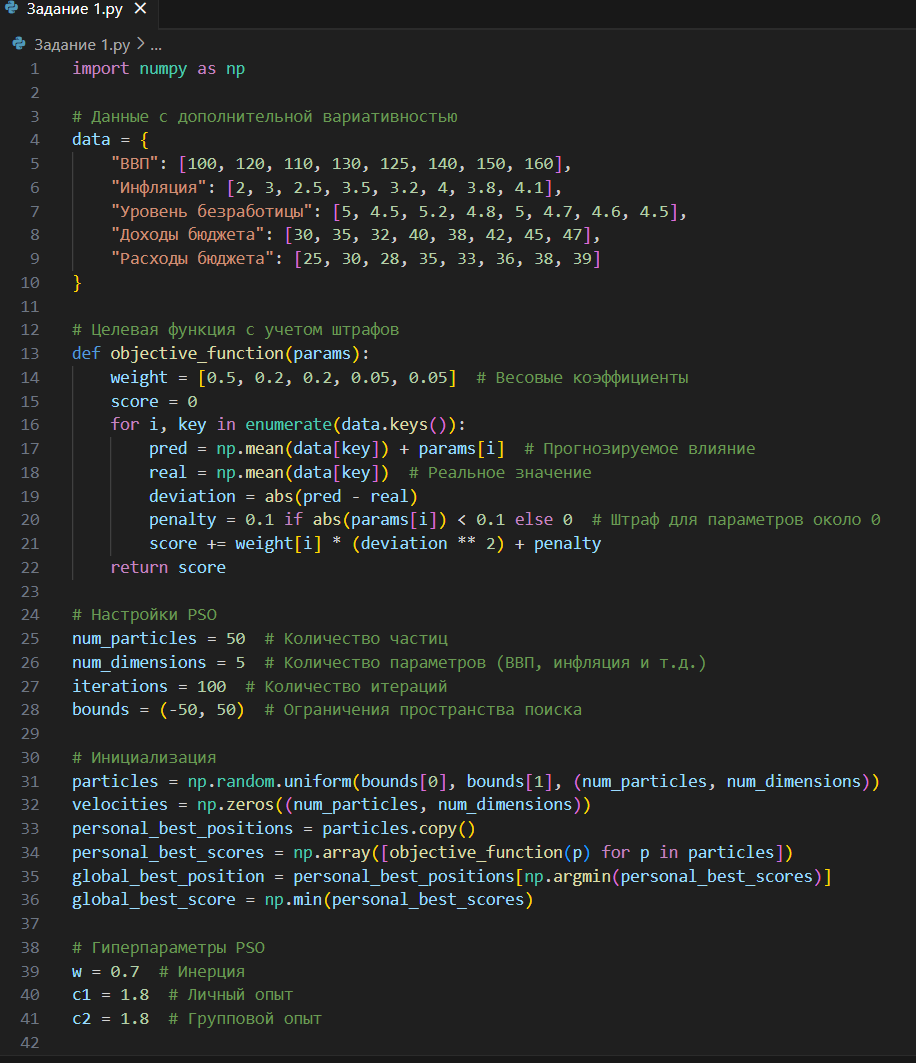
Экономический показатель 1 (ВВП): [100, 120, 110, 130, 125];

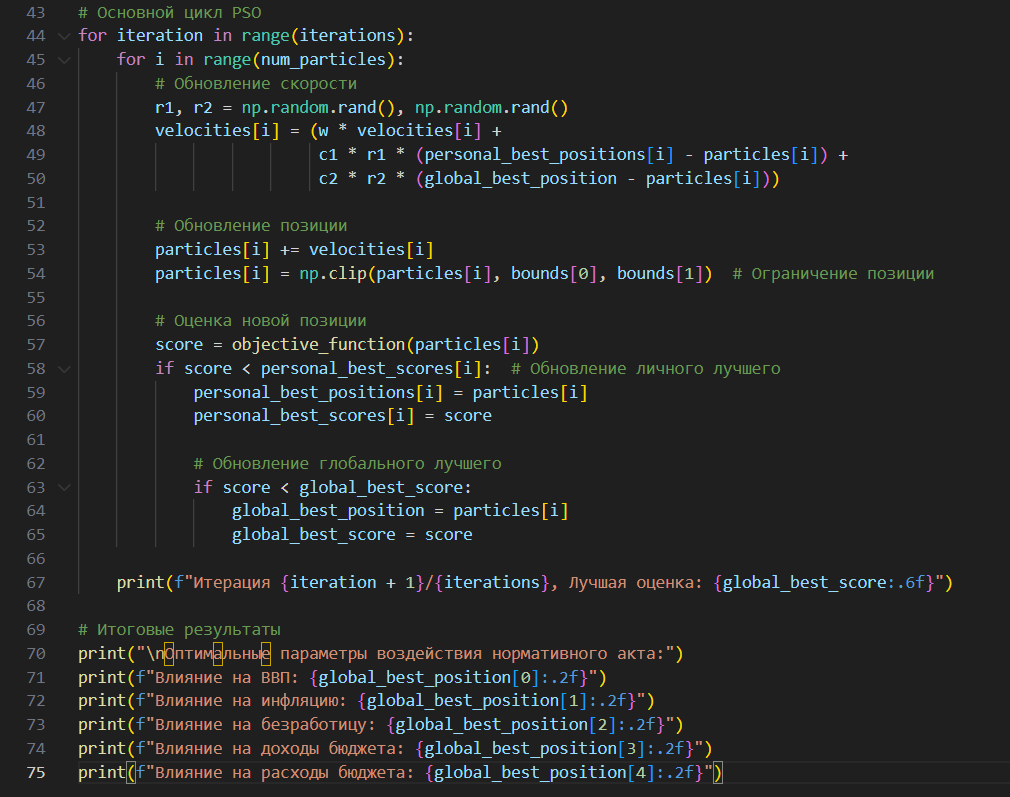
Экономический показатель 2 (Инфляция): [2, 3, 2.5, 3.5, 3.2];

Экономический показатель 3 (Уровень безработицы): [5, 4.5, 5.2, 4.8, 5];

Экономический показатель 4 (Доходы бюджета): [30, 35, 32, 40, 38];

Экономический показатель 5 (Расходы бюджета): [25, 30, 28, 35, 33].





**Результат**:

PS D:\Tin\Nam 3\Ki 1\Информационные системы в экономике\Zadanhie7> python -u "d:\Tin\Nam 3\Ki 1\Информационные системы в экономике\Zadanhie7\Задание 1.py"

Итерация 1/100, Лучшая оценка: 9.558105

Итерация 2/100, Лучшая оценка: 9.558105

Итерация 3/100, Лучшая оценка: 9.558105

Итерация 4/100, Лучшая оценка: 9.558105

Итерация 5/100, Лучшая оценка: 9.558105

Итерация 6/100, Лучшая оценка: 9.558105

Итерация 7/100, Лучшая оценка: 9.558105

Итерация 8/100, Лучшая оценка: 9.558105

Итерация 9/100, Лучшая оценка: 9.558105

Итерация 10/100, Лучшая оценка: 4.933667

Итерация 11/100, Лучшая оценка: 4.933667

Итерация 12/100, Лучшая оценка: 4.933667

Итерация 13/100, Лучшая оценка: 4.933667

Итерация 14/100, Лучшая оценка: 4.933667

Итерация 15/100, Лучшая оценка: 4.933667

Итерация 16/100, Лучшая оценка: 4.933667

Итерация 17/100, Лучшая оценка: 4.933667

Итерация 18/100, Лучшая оценка: 1.436777

Итерация 19/100, Лучшая оценка: 1.436777

Итерация 20/100, Лучшая оценка: 1.436777

Итерация 21/100, Лучшая оценка: 1.436777

Итерация 22/100, Лучшая оценка: 1.436777

Итерация 23/100, Лучшая оценка: 1.436777

Итерация 24/100, Лучшая оценка: 1.436777

Итерация 25/100, Лучшая оценка: 1.436777

Итерация 26/100, Лучшая оценка: 1.436777

Итерация 27/100, Лучшая оценка: 1.436777

Итерация 28/100, Лучшая оценка: 1.436777

Итерация 29/100, Лучшая оценка: 1.436777

Итерация 30/100, Лучшая оценка: 1.436777

Итерация 31/100, Лучшая оценка: 1.436777

Итерация 32/100, Лучшая оценка: 1.436777

Итерация 33/100, Лучшая оценка: 1.436777

Итерация 34/100, Лучшая оценка: 1.436777

Итерация 35/100, Лучшая оценка: 1.382520

Итерация 36/100, Лучшая оценка: 1.281613

Итерация 37/100, Лучшая оценка: 1.281613

Итерация 38/100, Лучшая оценка: 1.281613

Итерация 39/100, Лучшая оценка: 1.281613

Итерация 40/100, Лучшая оценка: 1.281613

Итерация 41/100, Лучшая оценка: 1.261939

Итерация 42/100, Лучшая оценка: 1.261939

Итерация 43/100, Лучшая оценка: 1.186875

Итерация 44/100, Лучшая оценка: 1.186875

Итерация 45/100, Лучшая оценка: 1.105734

Итерация 46/100, Лучшая оценка: 1.105734

Итерация 47/100, Лучшая оценка: 1.105734

Итерация 48/100, Лучшая оценка: 1.105734

Итерация 49/100, Лучшая оценка: 1.043816

Итерация 50/100, Лучшая оценка: 0.968026

Итерация 51/100, Лучшая оценка: 0.875662

Итерация 52/100, Лучшая оценка: 0.873996

Итерация 53/100, Лучшая оценка: 0.660859

Итерация 54/100, Лучшая оценка: 0.615212

Итерация 55/100, Лучшая оценка: 0.615212

Итерация 56/100, Лучшая оценка: 0.586269

Итерация 57/100, Лучшая оценка: 0.350287

Итерация 58/100, Лучшая оценка: 0.106796

Итерация 59/100, Лучшая оценка: 0.106796

Итерация 60/100, Лучшая оценка: 0.106796

Итерация 61/100, Лучшая оценка: 0.106796

Итерация 62/100, Лучшая оценка: 0.106796

Итерация 63/100, Лучшая оценка: 0.106796

Итерация 64/100, Лучшая оценка: 0.106796

Итерация 65/100, Лучшая оценка: 0.106796

Итерация 66/100, Лучшая оценка: 0.106796

Итерация 67/100, Лучшая оценка: 0.106796

Итерация 68/100, Лучшая оценка: 0.106796

Итерация 69/100, Лучшая оценка: 0.106796

Итерация 70/100, Лучшая оценка: 0.093033

Итерация 71/100, Лучшая оценка: 0.093033

Итерация 72/100, Лучшая оценка: 0.093033

Итерация 73/100, Лучшая оценка: 0.093033

Итерация 74/100, Лучшая оценка: 0.093033

Итерация 75/100, Лучшая оценка: 0.093033

Итерация 76/100, Лучшая оценка: 0.093033

Итерация 77/100, Лучшая оценка: 0.093033

Итерация 78/100, Лучшая оценка: 0.093033

Итерация 79/100, Лучшая оценка: 0.093033

Итерация 80/100, Лучшая оценка: 0.093033

Итерация 81/100, Лучшая оценка: 0.076216

Итерация 82/100, Лучшая оценка: 0.076216

Итерация 83/100, Лучшая оценка: 0.056135

Итерация 84/100, Лучшая оценка: 0.056135

Итерация 85/100, Лучшая оценка: 0.056135

Итерация 86/100, Лучшая оценка: 0.050085

Итерация 87/100, Лучшая оценка: 0.050085

Итерация 88/100, Лучшая оценка: 0.050085

Итерация 89/100, Лучшая оценка: 0.050085

Итерация 90/100, Лучшая оценка: 0.050085

Итерация 91/100, Лучшая оценка: 0.050085

Итерация 92/100, Лучшая оценка: 0.050085

Итерация 93/100, Лучшая оценка: 0.050085

Итерация 94/100, Лучшая оценка: 0.050085

Итерация 95/100, Лучшая оценка: 0.050085

Итерация 96/100, Лучшая оценка: 0.050085

Итерация 97/100, Лучшая оценка: 0.050085

Итерация 98/100, Лучшая оценка: 0.050085

Итерация 99/100, Лучшая оценка: 0.050085

Итерация 100/100, Лучшая оценка: 0.050085

Оптимальные параметры воздействия нормативного акта:

Влияние на ВВП: -0.09

Влияние на инфляцию: 0.12

Влияние на безработицу: 0.32

Влияние на доходы бюджета: -0.33

Влияние на расходы бюджета: 0.60

**Задание 2:** Алгоритмы муравьиной колонии (ACO) для управления программами проектов

Необходимо реализовать на пайтоне алгоритм муравьиной колонии для оптимизации распределения ресурсов в проекте. Программа должна моделировать несколько проектов, каждый из которых имеет свои задачи и необходимые ресурсы. Целью является нахождение оптимального распределения ресурсов, чтобы минимизировать общее время выполнения всех задач.

Тестовые данные:

Проект 1:

Задача 1: 5 ресурсов, 10 дней;

Задача 2: 3 ресурса, 5 дней.

Проект 2:

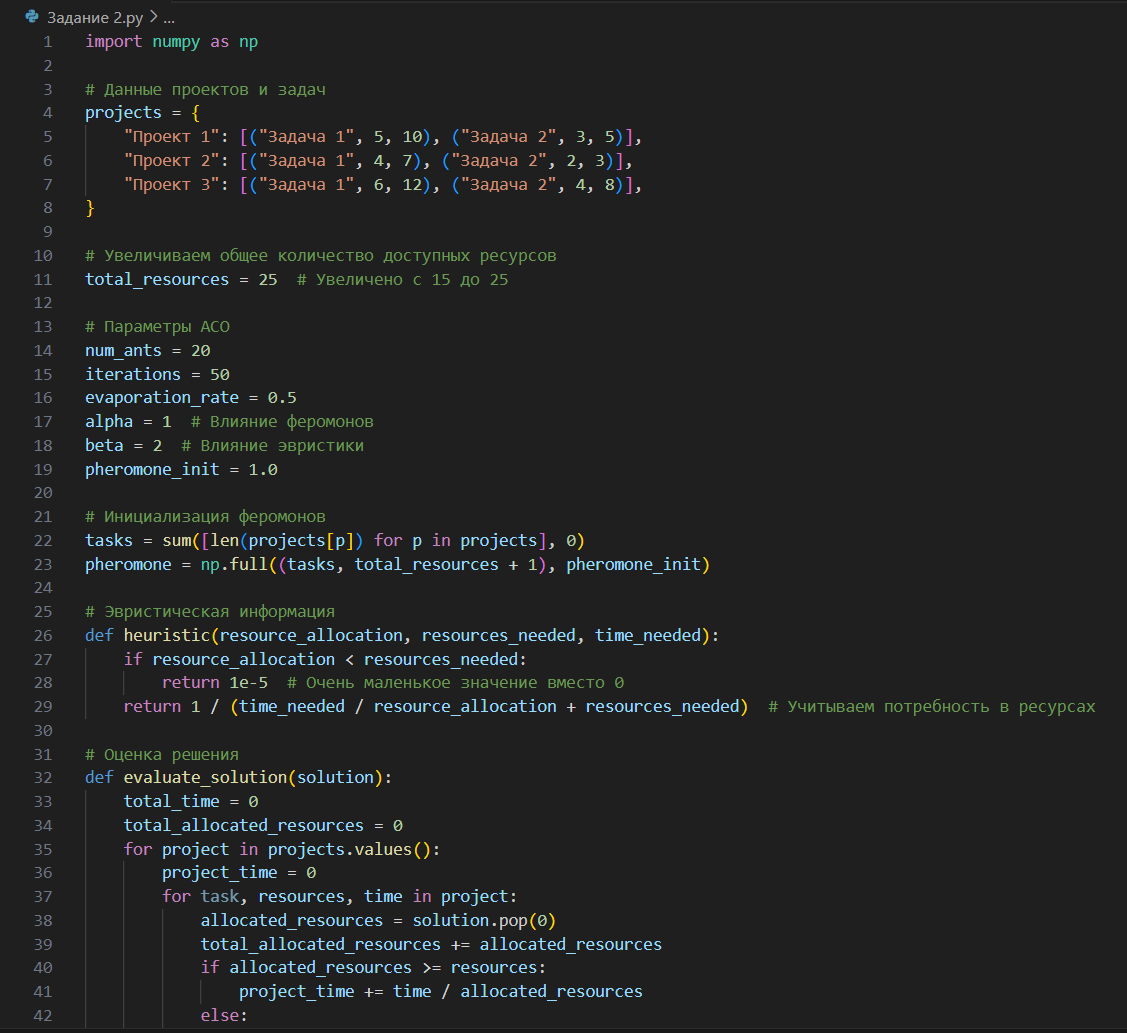
Задача 1: 4 ресурса, 7 дней;

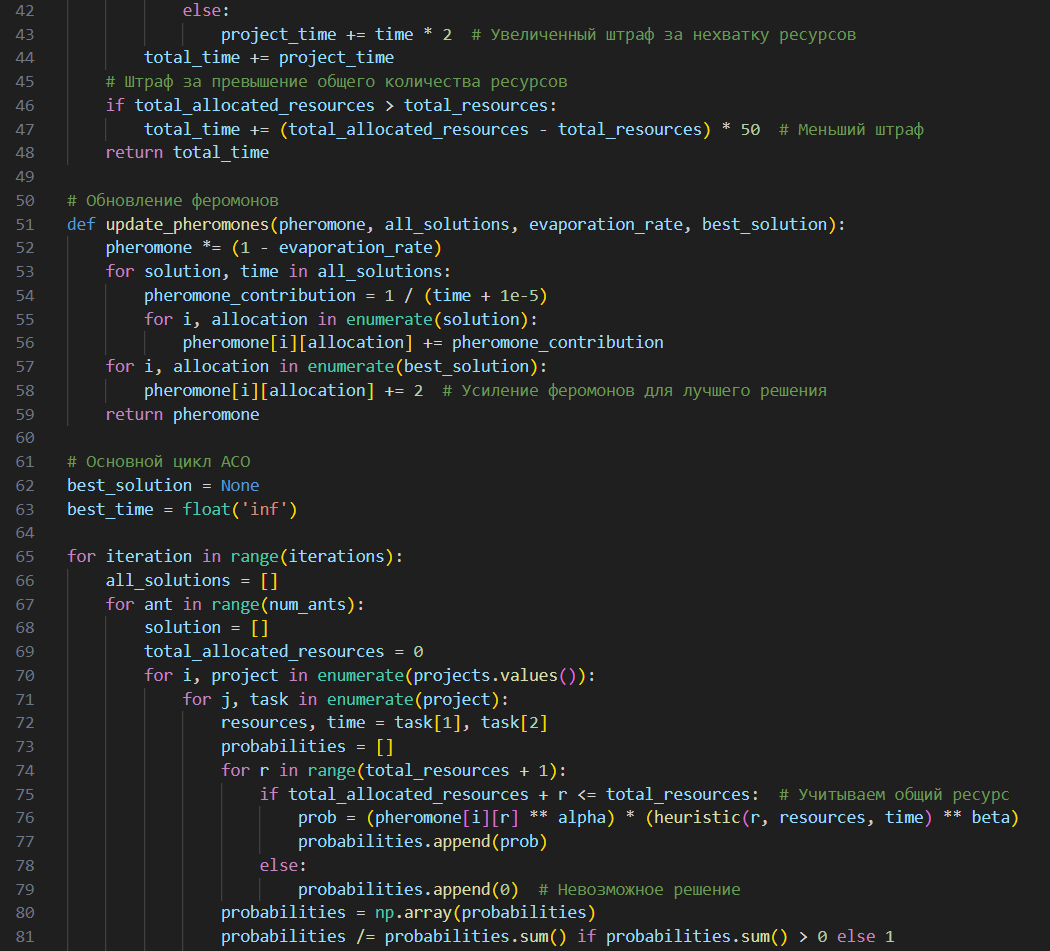
Задача 2: 2 ресурса, 3 дня.

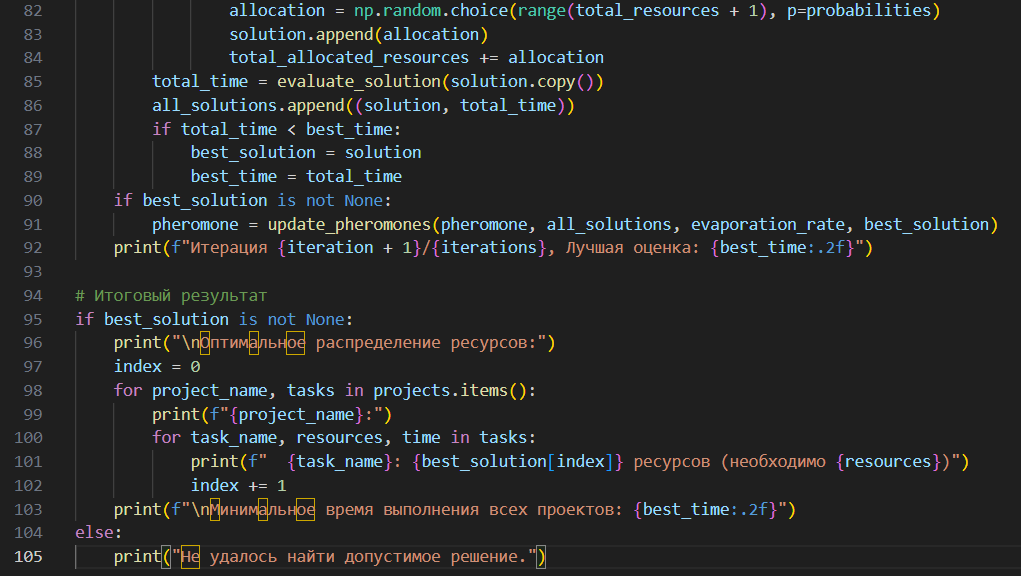
Проект 3:

Задача 1: 6 ресурсов, 12 дней;

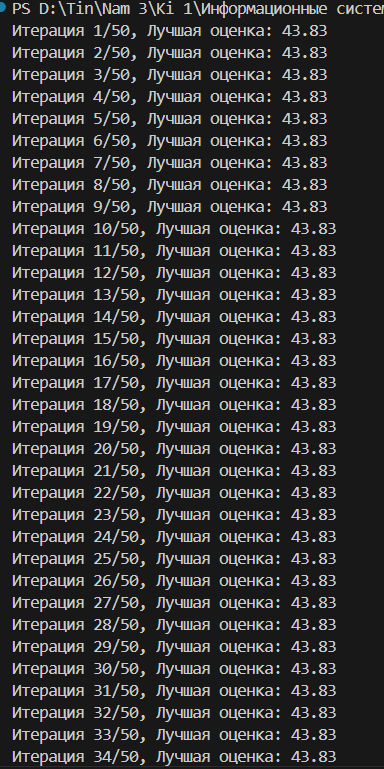
Задача 2: 4 ресурса, 8 дней.

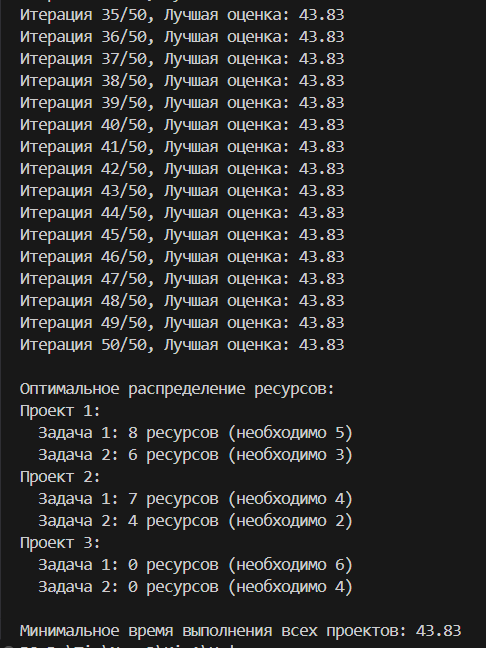






**Результат:**





**Задание 3:** Алгоритмы пчелиной колонии (Bee Algorithm) для управления рисками в процессах бюджетирования.

Разработайте программу на пайтоне, которая использует алгоритм пчелиной колонии для оценки рисков в процессе бюджетирования. Программа должна моделировать различные сценарии бюджета, определять риски и находить оптимальные стратегии минимизации этих рисков.

Тестовые данные:

Сценарий 1:

Риск 1: вероятность 20%, стоимость 1000;

Риск 2: вероятность 30%, стоимость 2000.

Сценарий 2:

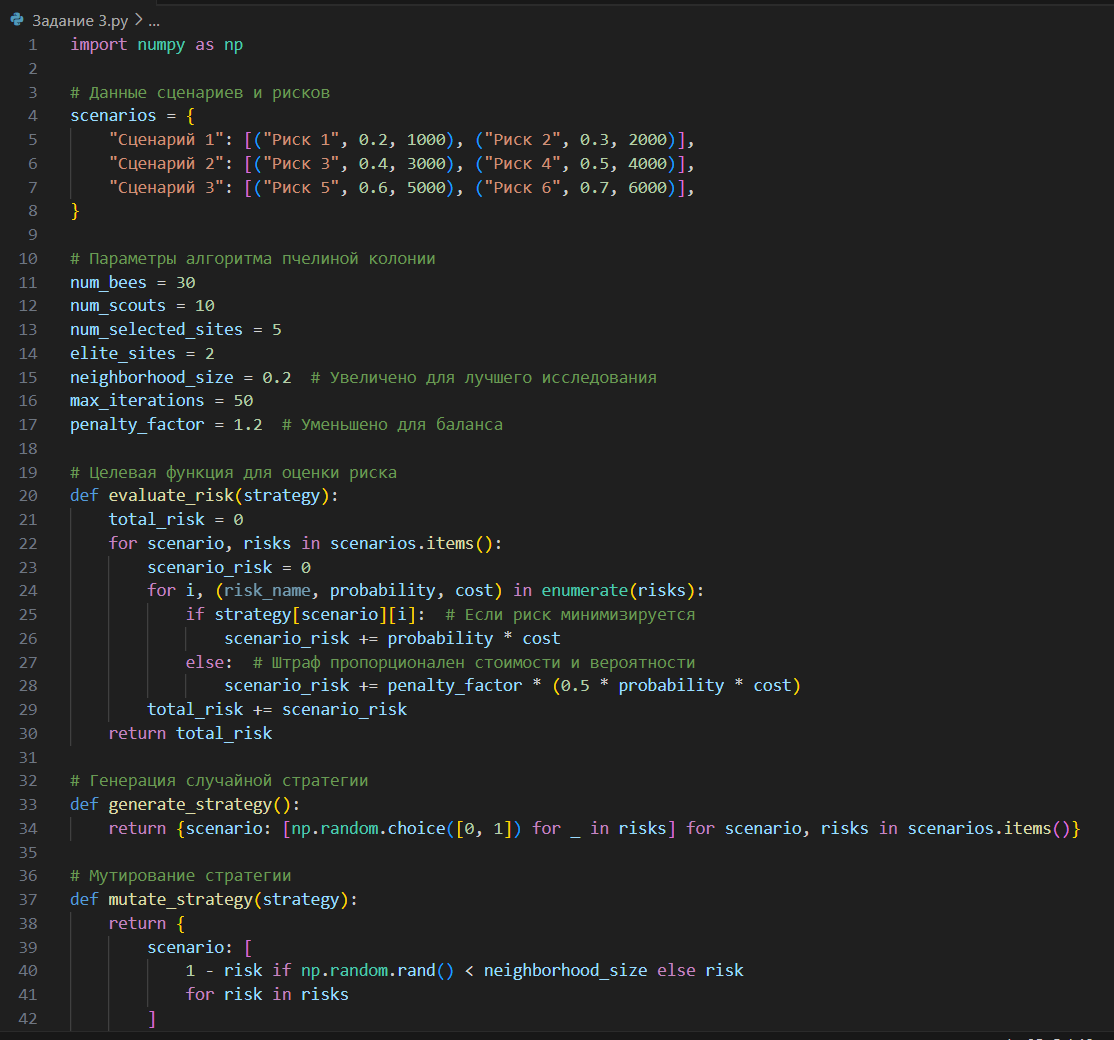
Риск 3: вероятность 40%, стоимость 3000;

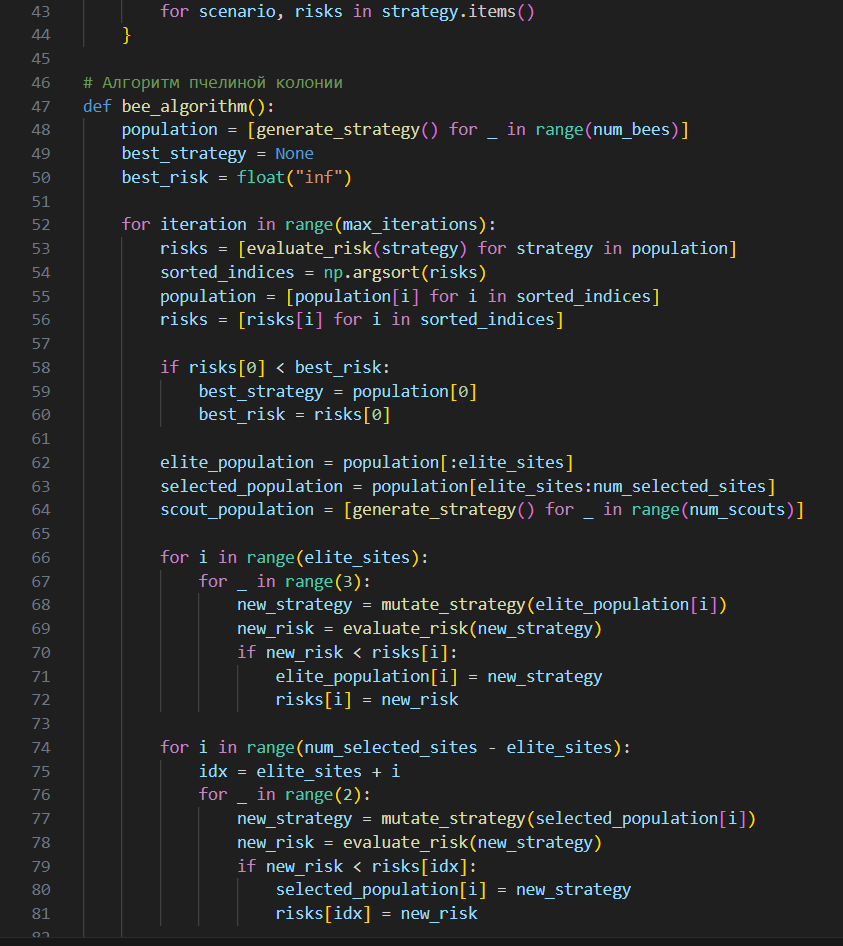
Риск 4: вероятность 50%, стоимость 4000.

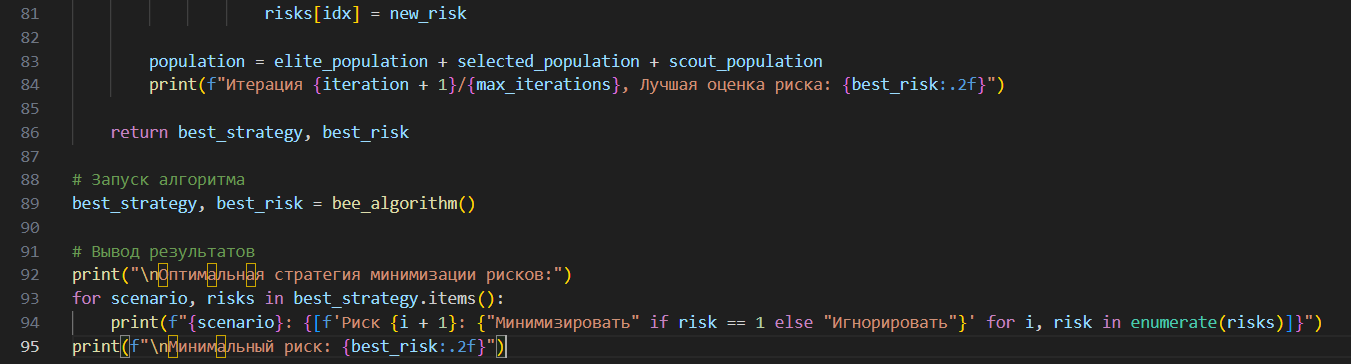
Сценарий 3:

Риск 5: вероятность 60%, стоимость 5000;

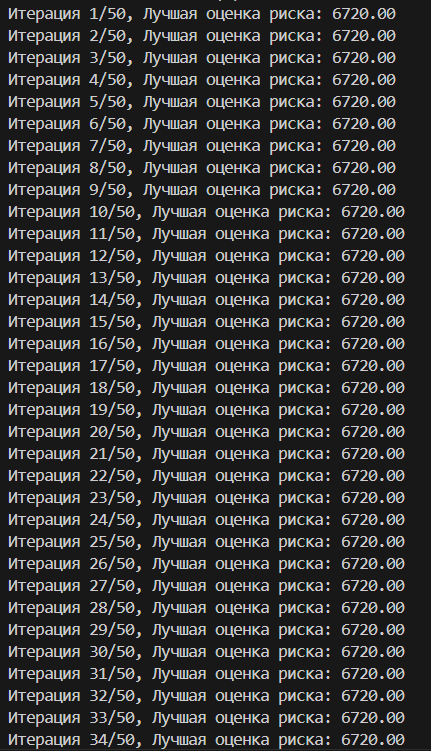
Риск 6: вероятность 70%, стоимость 6000.

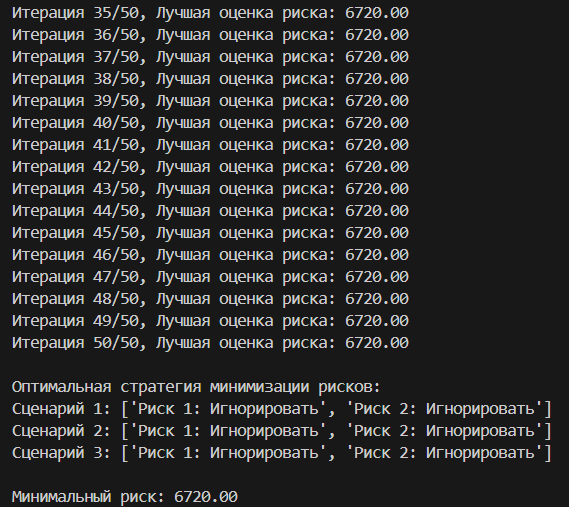






**Результат:**





**Задание 4:** Алгоритмы муравьиной колонии для многокритериальной оптимизации в управлении проектами

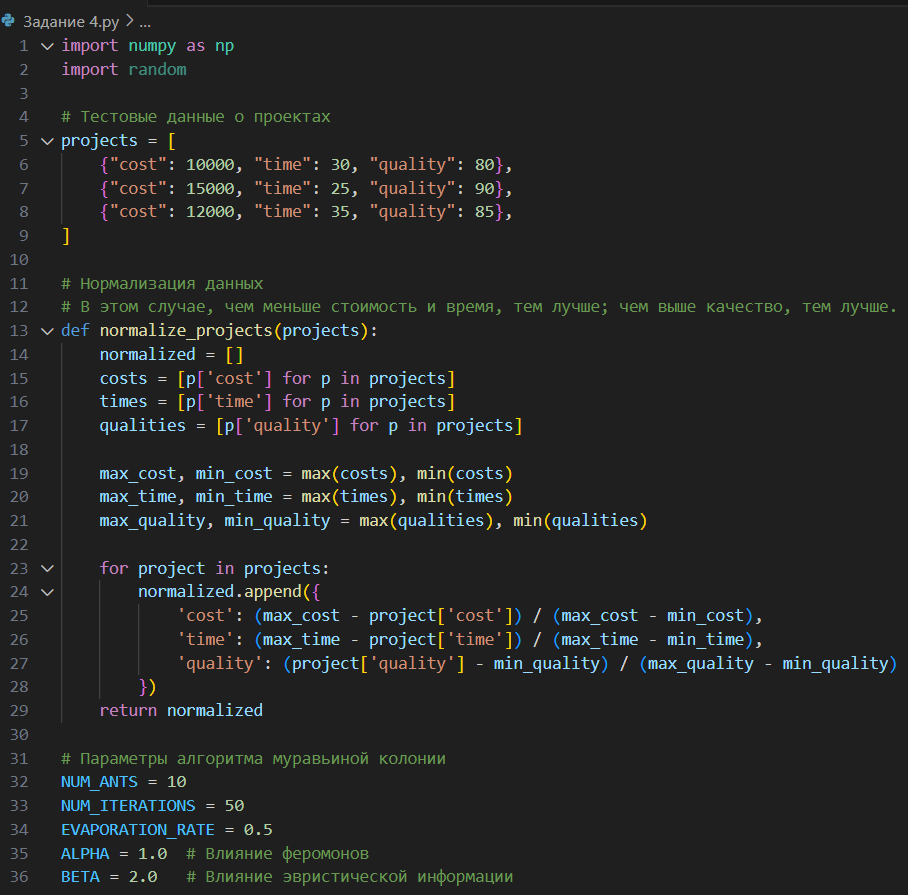
Разработайте программу на пайтоне, которая использует многокритериальную оптимизацию на основе муравьиной колонии для оценки нескольких проектов с различными критериями (стоимость, время, качество). Программа должна находить компромиссные решения для выбора наилучшего проекта.

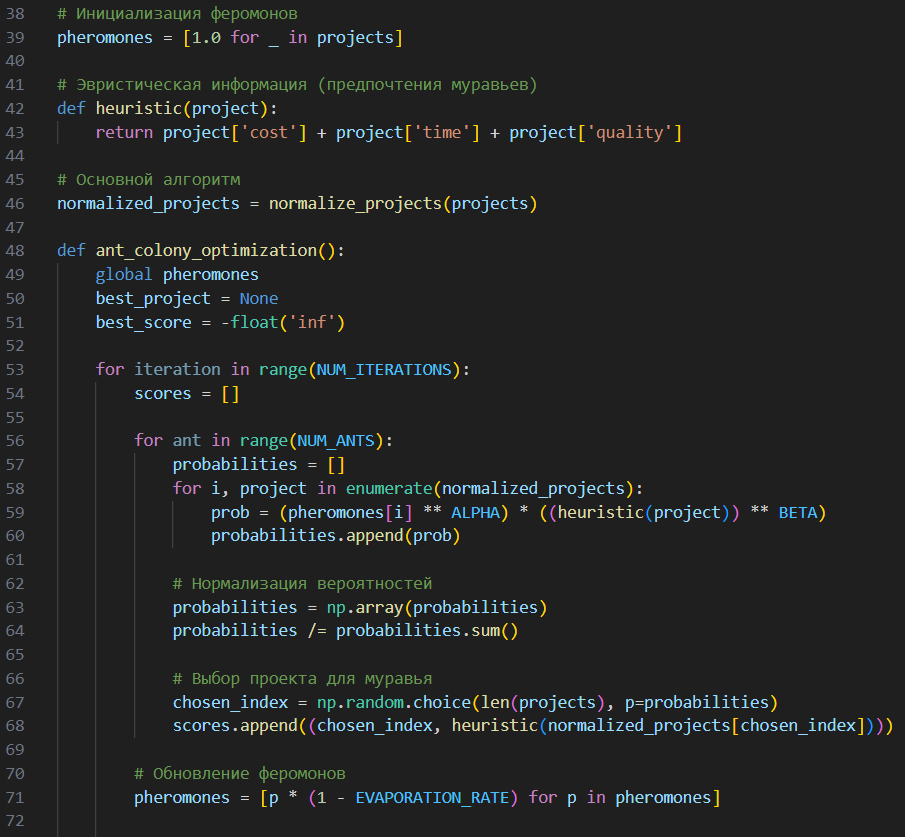
Тестовые данные:

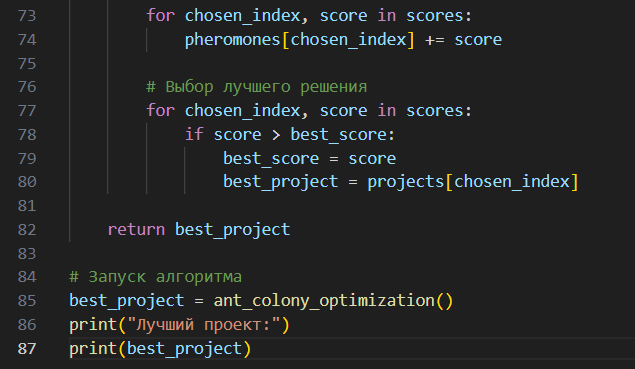
Проект 1: стоимость 10000, время 30 дней, качество 80%;

Проект 2: стоимость 15000, время 25 дней, качество 90%;.

Проект 3: стоимость 12000, время 35 дней, качество 85%







**Результат:**

Лучший проект:

{'cost': 15000, 'time': 25, 'quality': 90}

**Задание 5:** Алгоритмы флокинга для управления командами в проектах

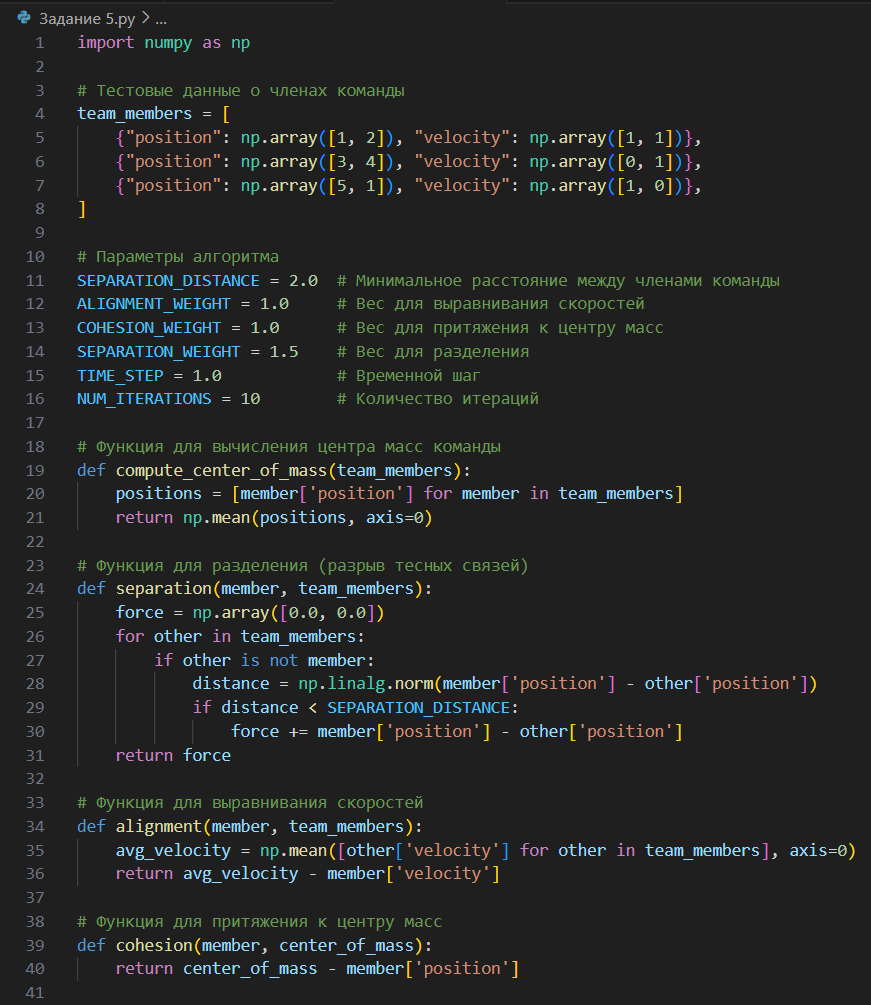
Реализуйте алгоритм на пайтоне флокинга для оптимизации работы команд в проекте. Программа должна моделировать взаимодействие членов команды, используя принципы разделения, выравнивания и притяжения, чтобы улучшить коммуникацию и эффективность работы.

Тестовые данные:

Член команды 1: положение (1, 2), скорость (1, 1);

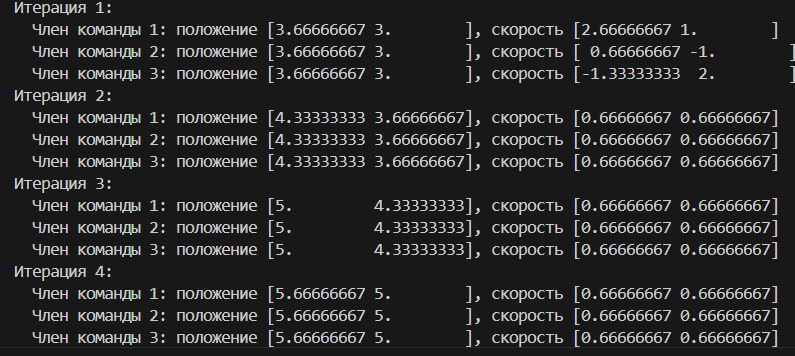
Член команды 2: положение (3, 4), скорость (0, 1);

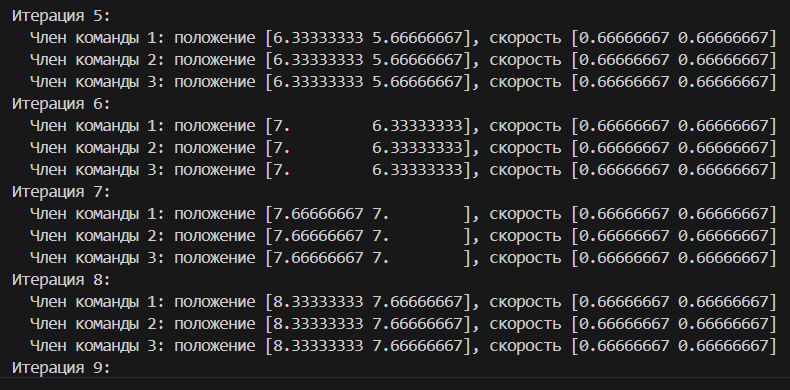
Член команды 3: положение (5, 1), скорость (1, 0).

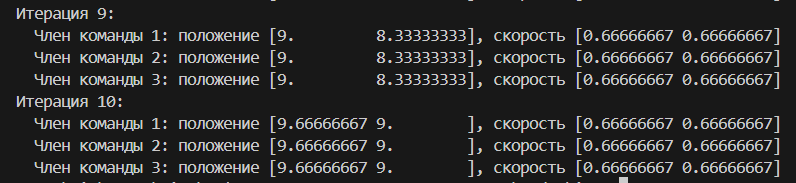




**Результат:**







**Задание 6:** Использование нескольких методов роевого интеллекта

Реализуйте алгоритм на пайтоне с использованием нескольких методов роевого интеллекта, который будет использоваться для оценки и прогнозирования показателей социально-экономического развития региона. Алгоритм должен учитывать исторические данные и различные факторы, влияющие на развитие региона, находить оптимальные параметры для анализа и прогнозирования. Целевая функция может включать: максимизацию ВВП, минимизацию уровня безработицы, оптимизацию расходов на социальные нужды, увеличение уровня инвестиций.

Тестовые данные (исторические):

ВВП: [100, 120, 110, 130, 125];

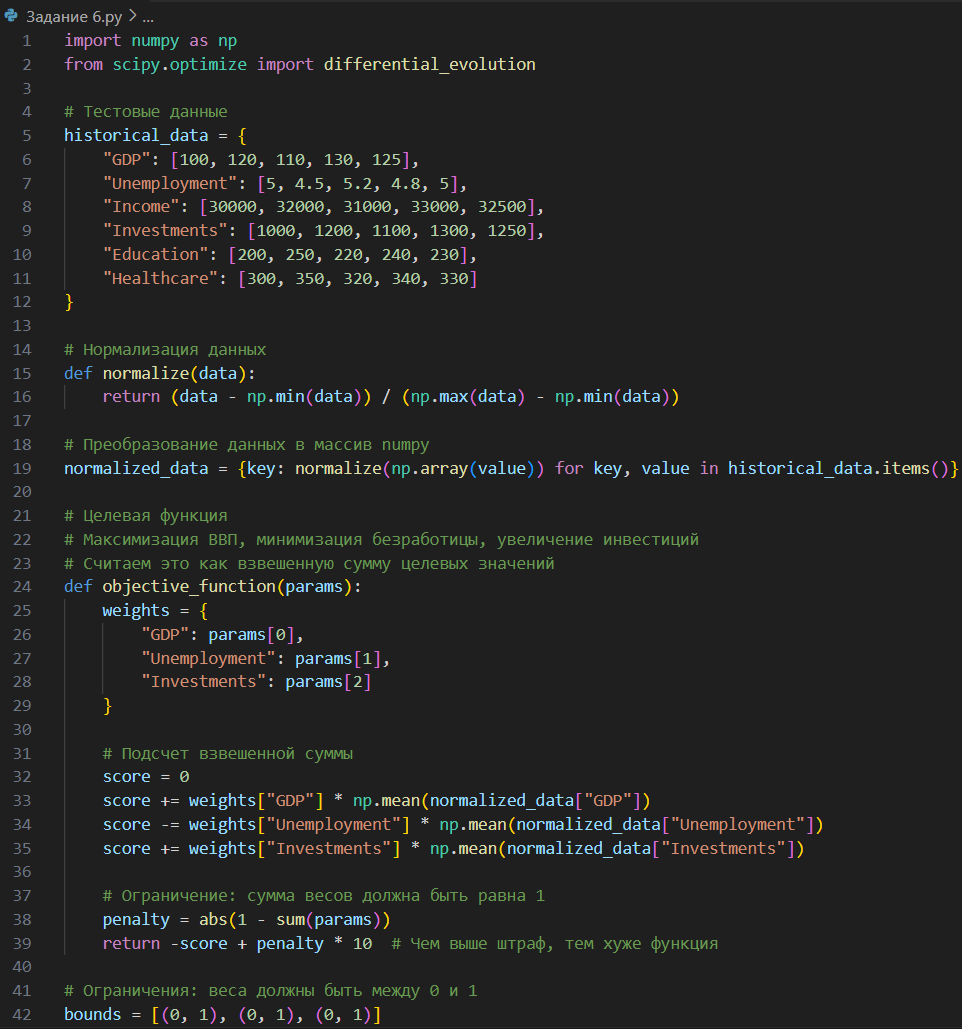
Уровень безработицы: [5, 4.5, 5.2, 4.8, 5] ;

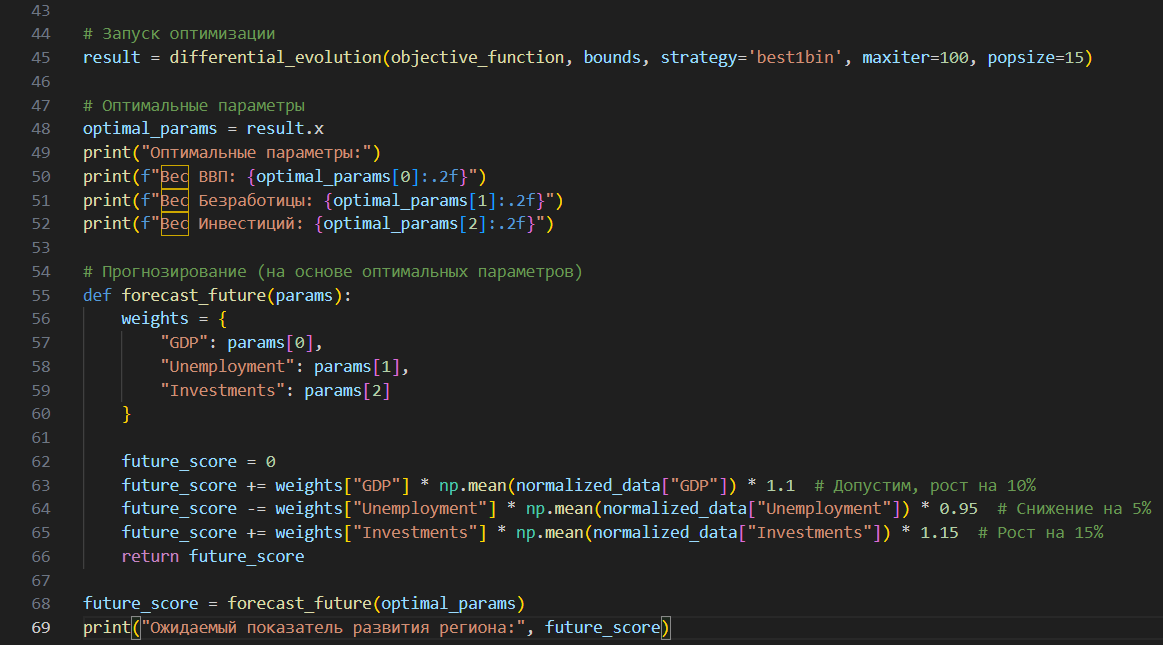
Средний доход на душу населения: [30000, 32000, 31000, 33000, 32500] ;

Уровень инвестиций: [1000, 1200, 1100, 1300, 1250] ;

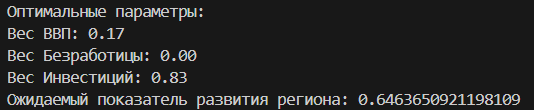
Расходы на образование: [200, 250, 220, 240, 230] ;

Расходы на здравоохранение: [300, 350, 320, 340, 330].





**Результат:**



**Для каждого задания обосновать использование методов роевого ИИ, предложить наиболее походящие методы решения.**