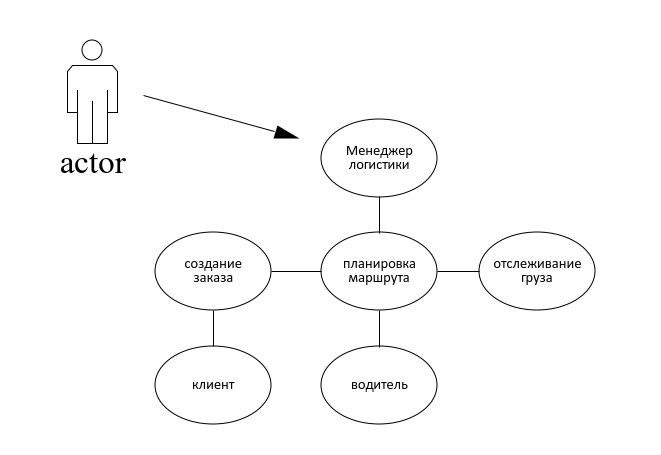
1. 

2. Листинг программы

import pandas as pd

import numpy as np

class TransportationProblem:

def \_\_init\_\_(self, supply, demand, costs):

self.supply = np.array(supply)

self.demand = np.array(demand)

self.costs = np.array(costs)

self.allocation = np.zeros\_like(self.costs, dtype=int)

self.m = len(supply)

self.n = len(demand)

def min\_element(self):

supply\_copy = self.supply.copy()

demand\_copy = self.demand.copy()

while True:

min\_cost = np.inf

min\_i, min\_j = -1, -1

for i in range(self.m):

for j in range(self.n):

if supply\_copy[i] > 0 and demand\_copy[j] > 0 and self.costs[i, j] < min\_cost:

min\_cost = self.costs[i, j]

min\_i, min\_j = i, j

if min\_i == -1:

break

amount = min(supply\_copy[min\_i], demand\_copy[min\_j])

self.allocation[min\_i, min\_j] = amount

supply\_copy[min\_i] -= amount

demand\_copy[min\_j] -= amount

def calculate\_total\_cost(self):

total\_cost = np.sum(self.allocation \* self.costs)

return total\_cost

def to\_dataframe(self):

return pd.DataFrame(self.allocation, columns=[f'Потр{i+1}' for i in range(self.n)], index=[f'Пост{i+1}' for i in range(self.m)])

# Пример использования:

supply = [35, 48, 66]

demand = [390, 490, 480, 480]

costs = [[33, 31, 32, 31],

[30, 29, 29, 28],

[32, 33, 31, 29]]

problem = TransportationProblem(supply, demand, costs)

problem.min\_element()

print(problem.to\_dataframe())

print("Общая стоимость:", problem.calculate\_total\_cost())

Тест

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 |
| P1 | 0 | 13 | 0 | 2 |
| P2 | 6 | 0 | 5 | 2 |
| P3 | 1 | 0 | 3 | 4 |
| Общая стоимость: 876 | | | | |
|  | P1 | P2 | P3 | P4 |
| P1 | 0 | 150 | 50 | 0 |
| P2 | 100 | 0 | 0 | 0 |
| P3 | 0 | 0 | 50 | 100 |
| Общая стоимость: 3750 | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 |
| P1 | 0 | 150 | 50 | 0 |
| P2 | 100 | 0 | 0 | 0 |
| P3 | 0 | 0 | 50 | 100 |
| Общая стоимость: 3750 | | | | |