Задание №1

**1. Анализ предметной области:**

Транспортная задача - оптимизация грузоперевозок между поставщиками и потребителями с минимальными затратами.

**Предполагаемое приложение:**

Позволяет вводить данные о поставщиках, потребителях, запасах, и издержках, находит оптимальный план перевозок.

**2. Анализ требований:**

**Функциональные:**

* **Ввод:** Добавление/удаление/редактирование поставщиков, потребителей, запасов, потребностей, издержек.
* **Расчет:** Оптимальное решение, план перевозок, общие затраты.
* **Проверка:** Корректность ввода, соответствие запасов и потребностей.
* **Данные:** Сохранение и загрузка.

**Нефункциональные:** Интуитивный интерфейс, быстрая работа, надежность.

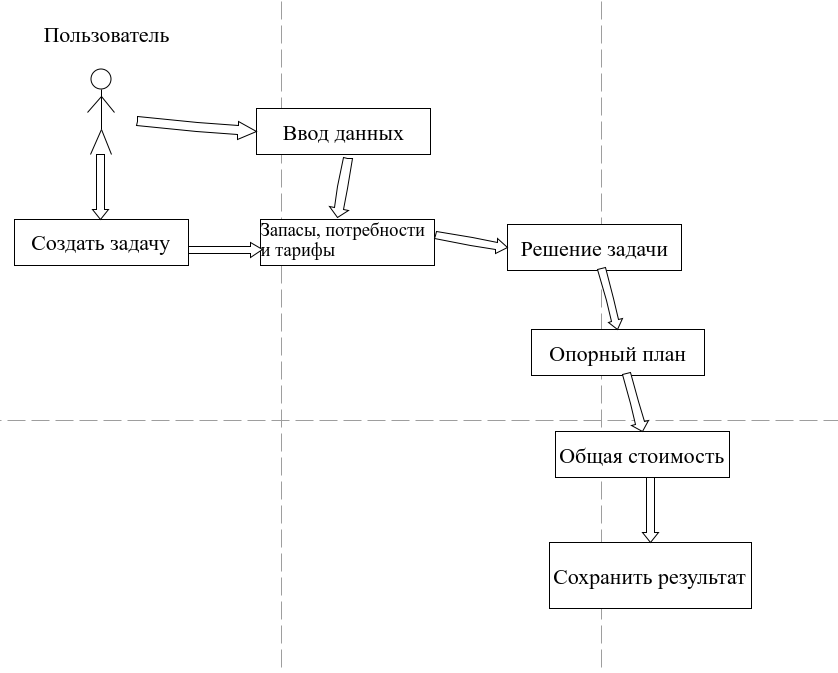
**3. Диаграмма вариантов использования:**

**Действующее лицо:**

* **Пользователь:** Работает с приложением.

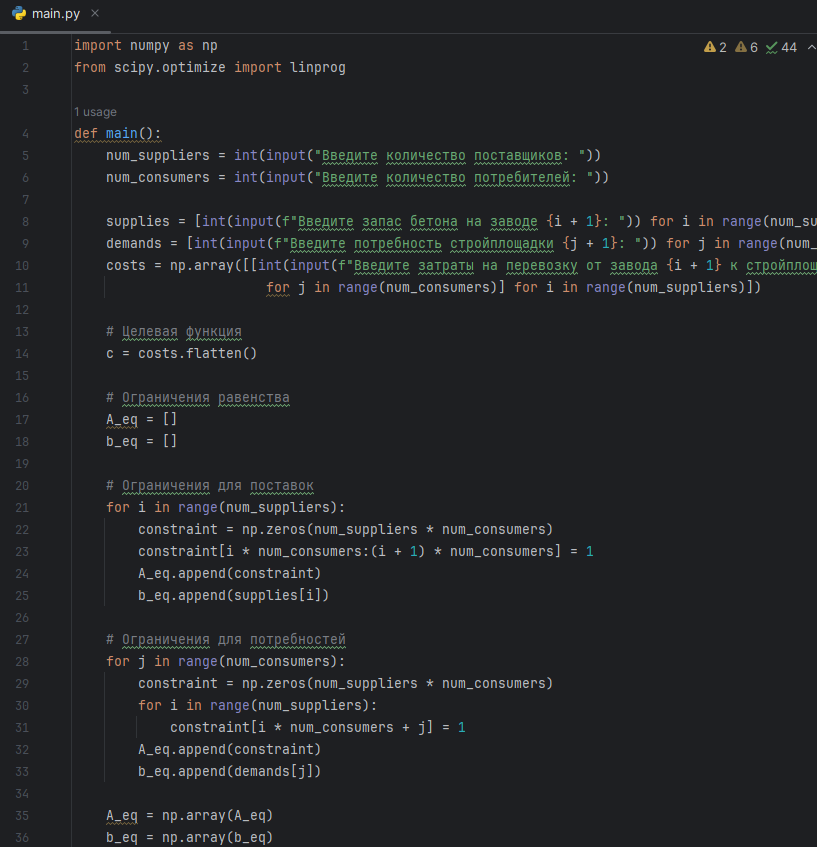
**Варианты использования:**

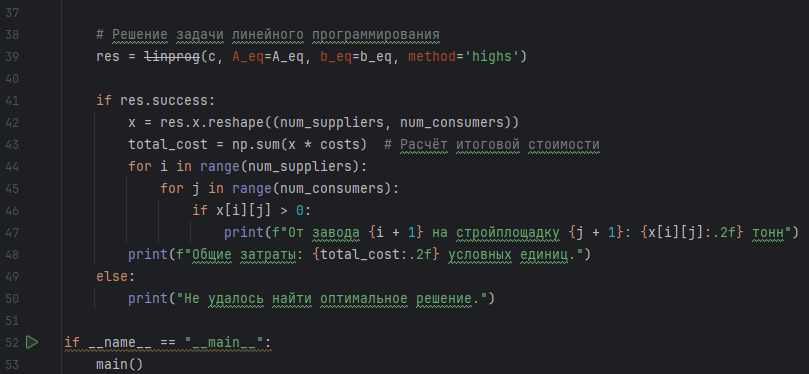
* **Управление поставщиками:** Добавление, удаление, изменение.
* **Управление потребителями:** Добавление, удаление, изменение.
* **Управление издержками:** Ввод и изменение.
* **Решение задачи:** Расчет, просмотр плана, просмотр затрат.
* **Управление данными:** Сохранение и загрузка.

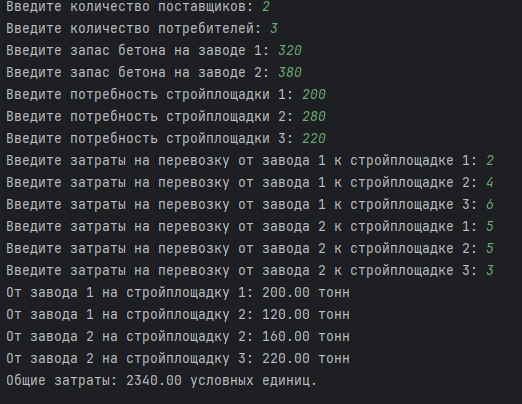
  
Рисунок №1. UML-диаграмма вариантов использования

Задание №2

Листинг программы:







import numpy as np

from scipy.optimize import linprog

def main():

num\_suppliers = int(input("Введите количество поставщиков: "))

num\_consumers = int(input("Введите количество потребителей: "))

supplies = [int(input(f"Введите запас бетона на заводе {i + 1}: ")) for i in range(num\_suppliers)]

demands = [int(input(f"Введите потребность стройплощадки {j + 1}: ")) for j in range(num\_consumers)]

costs = np.array([[int(input(f"Введите затраты на перевозку от завода {i + 1} к стройплощадке {j + 1}: "))

for j in range(num\_consumers)] for i in range(num\_suppliers)])

# Целевая функция

c = costs.flatten()

# Ограничения равенства

A\_eq = []

b\_eq = []

# Ограничения для поставок

for i in range(num\_suppliers):

constraint = np.zeros(num\_suppliers \* num\_consumers)

constraint[i \* num\_consumers:(i + 1) \* num\_consumers] = 1

A\_eq.append(constraint)

b\_eq.append(supplies[i])

# Ограничения для потребностей

for j in range(num\_consumers):

constraint = np.zeros(num\_suppliers \* num\_consumers)

for i in range(num\_suppliers):

constraint[i \* num\_consumers + j] = 1

A\_eq.append(constraint)

b\_eq.append(demands[j])

A\_eq = np.array(A\_eq)

b\_eq = np.array(b\_eq)

# Решение задачи линейного программирования

res = linprog(c, A\_eq=A\_eq, b\_eq=b\_eq, method='highs')

if res.success:

x = res.x.reshape((num\_suppliers, num\_consumers))

total\_cost = np.sum(x \* costs) # Расчёт итоговой стоимости

for i in range(num\_suppliers):

for j in range(num\_consumers):

if x[i][j] > 0:

print(f"От завода {i + 1} на стройплощадку {j + 1}: {x[i][j]:.2f} тонн")

print(f"Общие затраты: {total\_cost:.2f} условных единиц.")

else:

print("Не удалось найти оптимальное решение.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Задание №3

| **ID Тестового кейса** | **Описание** | **Входные данные** | **Ожидаемый результат** |
| --- | --- | --- | --- |
| TC-001 | Проверка, что приложение принимает корректные данные | Корректные значения | Приложение принимает данные и обрабатывает их без ошибок |  |  |
| TC-002 | Проверка обработки некорректных данных | Некорректные значения | Приложение выдает сообщение об ошибке |  |  |
| TC-003 | Проверка времени отклика приложения | Большой набор данных | Время отклика не превышает 2 секунд |  |  |
| TC-004 | Проверка корректности отображения UI элементов | - | Все элементы отображаются корректно |  |  |
| TC-005 | Проверка, что все кнопки доступны для взаимодействия | - | Все кнопки активны и кликабельны |  |  |
| TC-006 | Проверка реакции приложения на пустые поля | Пустые значения | Приложение выдает сообщение об ошибке |  |  |
| TC-007 | Проверка, что данные сохраняются при нажатии кнопки | Корректные данные | Данные успешно сохраняются |  |  |
| TC-008 | Проверка, что данные удаляются при нажатии кнопки | Выбранные данные | Данные успешно удаляются |  |  |