**Федеральное государственное образовательное бюджетное**

**учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

Факультет информационных технологий и анализа больших данных

**Департамент анализа данных и машинного**

**обучения**

Домашняя работа №1

По дисциплине «Математические методы принятия

решений»

Выполнили:

Студенты группы БИ20-4

Иванова Ксения

Киракосян Виген

Крылов Никита

Мытарева Ангелина

Петрова Арина

Чайковская Анна

Проверил:

Аксенов Дмитрий Андреевич

**Москва – 2022**

СОДЕРЖАНИЕ

1. Физическая модель3
2. Математическая модель3
3. Архитектурные требования5
4. Заключение5

**Физическая модель**

Рассмотрим конкретный кейс из жизни: Фабрика «Московский шоколад» производит 2 вида кондитерских изделий: шоколад горький и ириски. Для их производства необходимо затратить определенное количество ингредиентов: какао бобы, карамель, сахар (в гр.). В таблице №1 представлены затраты ингредиентов на производство одного изделия.

Таблица №1 – «Затраты ингредиентов на производство»

|  |  |  |
| --- | --- | --- |
|  | Шоколад горький | Ириски |
| Какао бобы | 5 | 0 |
| Карамель | 15 | 17 |
| Сахар | 10 | 8 |

На складе завода хранится 10 000 гр. шоколада, 15 000 гр. сахара и 8 000 гр. карамели.

Необходимо определить сколько штук каждого из видов кондитерских изделий нужно произвести заводу, чтобы получить максимальную прибыль, если от продажи горького шоколада завод получает 90 рублей за штуку, от продажи ириски – 35 рублей.

**Математическая модель**

Проанализировав физическую модель задачи, можно сделать вывод, что её решение сводится к решению производственной задаче. Её можно рассматривать как задачу об определении плана производства продукции, обеспечивающий максимальную прибыль. Рассмотрим математическую модель классической производственной задачи в общем виде.

Пусть имеется предприятие, которое производит n видов продукции Пр. 1, Пр. 2, …, Пр. n. Для её производства оно располагает m видами ресурсов Рес. 1, Рес. 2, …, Рес. m, объём которых составляет соответственно: b1, b2, …, bm. Известна прибыль, которую можно получить от реализации единицы каждого вида продукции: С1, C2, …, Cn. Расход i-ого вида ресурсов на выпуск единицы j-ого вида продукции aij представлены в матрице:

Исходные данные:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ресурсы | Выпускаемая продукция | | | | Объём ресурсов |
| Пр. 1 | Пр. 2 | … | Пр. n |
| Рес. 1 | a11 | a12 | … | a1n | b1 |
| Рес. 2 | a21 | a22 | … | a2n | b2 |
| … | … | … | … | … | … |
| Рес. m | am1 | am2 | … | amn | bm |
| Цена ед. продукции | C1 | C2 | … | Cn |  |
| Цена выпуска продукции | X1 | X2 | … | Xn |  |

План-искомые данные: Xпл = (X1, X2, …, Xn)

Целевая функция:

Суммарная прибыль от продажи должна быть максимальной.

Ограничения:

Переменные не могут быть отрицательными:

**Общая форма записи:**

*; Xi – произвольные, I =*

**Архитектурные требования**

Проанализировав физическую модель задачи и формализацию данных, можно сделать вывод, что её решение сводится к решению производственной задачи. Её можно рассматривать как задачу об определении плана производства продукции, обеспечивающий максимальную прибыль. Для решения задачи был разработан следующий код:

* Импортируем 3 библиотеки (импортируем библиотеку по созданию графического интерфейса, импортируем библиотеку для математических вычислений, импортируем библиотеку для визуализации данных):

*from tkinter import \**

*import numpy as np*

*import matplotlib.pyplot as plt*

* Задаём функцию для изображения графика:

*def Draw():*

*# Создаём рисунок с координатной плоскостью*

*from matplotlib import ticker*

*fig = plt.subplots()*

*# Создаём область, в которой будет*

*# - отображаться график*

*x = np.arange(-1,170, 50)*

*#x = np.linspace(-4, 16, 100)*

*# Устанавливаем интервал основных делений:*

*ax = plt.gca()*

*ax.spines['top'].set\_color('none')*

*ax.spines['bottom'].set\_position('zero')*

*ax.spines['left'].set\_position('zero')*

*ax.spines['right'].set\_color('none')*

*ax.xaxis.set\_major\_locator(ticker.MultipleLocator(50))*

*# Устанавливаем интервал вспомогательных делений:*

*ax.xaxis.set\_minor\_locator(ticker.MultipleLocator(10))*

*# Добавляем подписи к осям:*

*ax.set\_xlabel('Товар X1', labelpad=-10)*

*ax.set\_ylabel('Товар X2', labelpad=-10)*

*#Задаем ограничительные функции*

*m1 = lambda x: int(t\_mat1.get()) / int( t\_prod21.get()) - int(t\_prod11.get()) \*x / int(t\_prod21.get())*

*m1\_1 = 'X2=' + str(int(t\_mat1.get()) / int( t\_prod21.get())) + "-" + str(int(t\_prod11.get()) / int(t\_prod21.get())) + "\*X1"*

*m2 = lambda x: int(t\_mat2.get()) / int( t\_prod22.get()) - int(t\_prod12.get()) \*x / int(t\_prod22.get())*

*m2\_2 ='X2=' + str(int(t\_mat2.get()) / int( t\_prod22.get())) + "-" + str(int(t\_prod12.get()) / int(t\_prod22.get())) + "\*X1"*

*m3 = lambda x: int(t\_mat3.get()) / int( t\_prod23.get()) - int(t\_prod13.get()) \*x / int(t\_prod23.get())*

*m3\_3 ='X2=' + str(int(t\_mat3.get()) / int( t\_prod23.get())) + "-" + str(int(t\_prod13.get()) / int(t\_prod23.get())) + "\*X1"*

*f0 = lambda x: 0 / int(t\_price2.get()) -int(t\_price1.get()) \* x / int(t\_price2.get())*

*#Выводим на форму функции m1, m2, m3*

*l\_RRR4 = Label(root, text =m1\_1 + '\n' + m2\_2 + '\n' + m3\_3)*

*l\_RRR4.grid(column=5, row=10, columnspan=2)*

*#Вычисление координат точки с высокой эффективностью, пересечение графика Материал 1 и Материал 2*

*t\_x1\_max = m1\_1[3:] + '=' + m2\_2[3:]*

*l\_RRR5 = Label(root, text= t\_x1\_max)*

*l\_RRR5.grid(column=0, row=12, columnspan=2)*

*#Рассчитаем координаты точки максимума эффективности*

*l\_RRR6 = Label(root, text= t\_x1\_max)*

*X1\_c = int(t\_mat2.get()) / int( t\_prod22.get()) - int(t\_mat1.get()) / int( t\_prod21.get())*

*X1\_k = -int(t\_prod11.get()) / int(t\_prod21.get()) + int(t\_prod12.get()) / int(t\_prod22.get())*

*X1\_max = str(X1\_c) + '=' +str(X1\_k) + '\*X1'*

*X2\_max = str(int(t\_mat2.get()) / int( t\_prod22.get())) + '-' + str (int(t\_prod12.get()) / int(t\_prod22.get())) + '\*' + str(X1\_c/X1\_k)*

*X2\_max\_c = int(t\_mat2.get()) / int( t\_prod22.get()) - int(t\_prod12.get()) / int(t\_prod22.get())\*(X1\_c/X1\_k)*

*l\_RRR6 = Label(root, text= t\_x1\_max +'\n' + X1\_max + '\n x1 = ' + str(X1\_c/X1\_k) + '\n x2 = ' + X2\_max + '\n X2 = ' + str(X2\_max\_c))*

*l\_RRR6.grid(column=0, row=12, columnspan=2)*

*#Перенесем значения точки максимума на график и форму*

*f\_max = t\_price1.get() + '\*' + str(X1\_c/X1\_k) + '+' + t\_price2.get() + '\*' + str(X2\_max\_c)*

*f\_max\_c = int(t\_price1.get()) \* (X1\_c/X1\_k) + int(t\_price2.get()) \* (X2\_max\_c)*

*l\_RRR7 = Label(root, text= 'Максимум эффективности f1=' + f\_max +'='+ str(f\_max\_c))*

*l\_RRR7.grid(column=0, row=13, columnspan=6)*

*#Построим линии уровня по максимальной точке*

*f1 = lambda x: (60\*4+110\*5) / int(t\_price2.get()) -int(t\_price1.get()) \* x / int(t\_price2.get())*

*plt.title('График производственной задачи', fontsize=12) #Нанесем заголовок*

*#Отрисуем линейные графики на координатной плоскости*

*plt.plot(x, m1(x), color='red')*

*plt.plot(x, m2(x), color='blue')*

*plt.plot(x, m3(x), color='green')*

*plt.plot(x, f0(x), '--', color='grey')*

*plt.plot(x, f1(x), '--', color='grey')*

*plt.plot(60, 110, '.', color='black')*

*plt.text(5, 90, l\_mat1.cget("text") +': ' + m1\_1, fontsize=8, color='red')*

*plt.text(5, 170, l\_mat2.cget("text") +': ' + m2\_2, fontsize=8, color='blue')*

*plt.text(5, 300, l\_mat3.cget("text") +': ' + m3\_3, fontsize=8, color='green')*

*plt.text(5, 10, 'f0 - минимум эффективности', fontsize=8, color='black')*

*plt.text(60, 120, 'f1 - максимум эффективности', fontsize=8, color='black')*

*plt.text(80, 150, 'f1=' + f\_max +'='+ str(f\_max\_c), fontsize=8, color='black')*

*#Сохраненим изображение графика в файл*

*plt.savefig('График.png')*

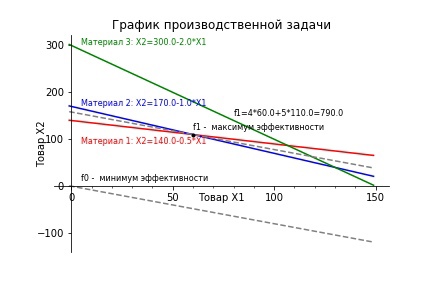
*img = Image.open('C://График.png')*

*plt.show()*

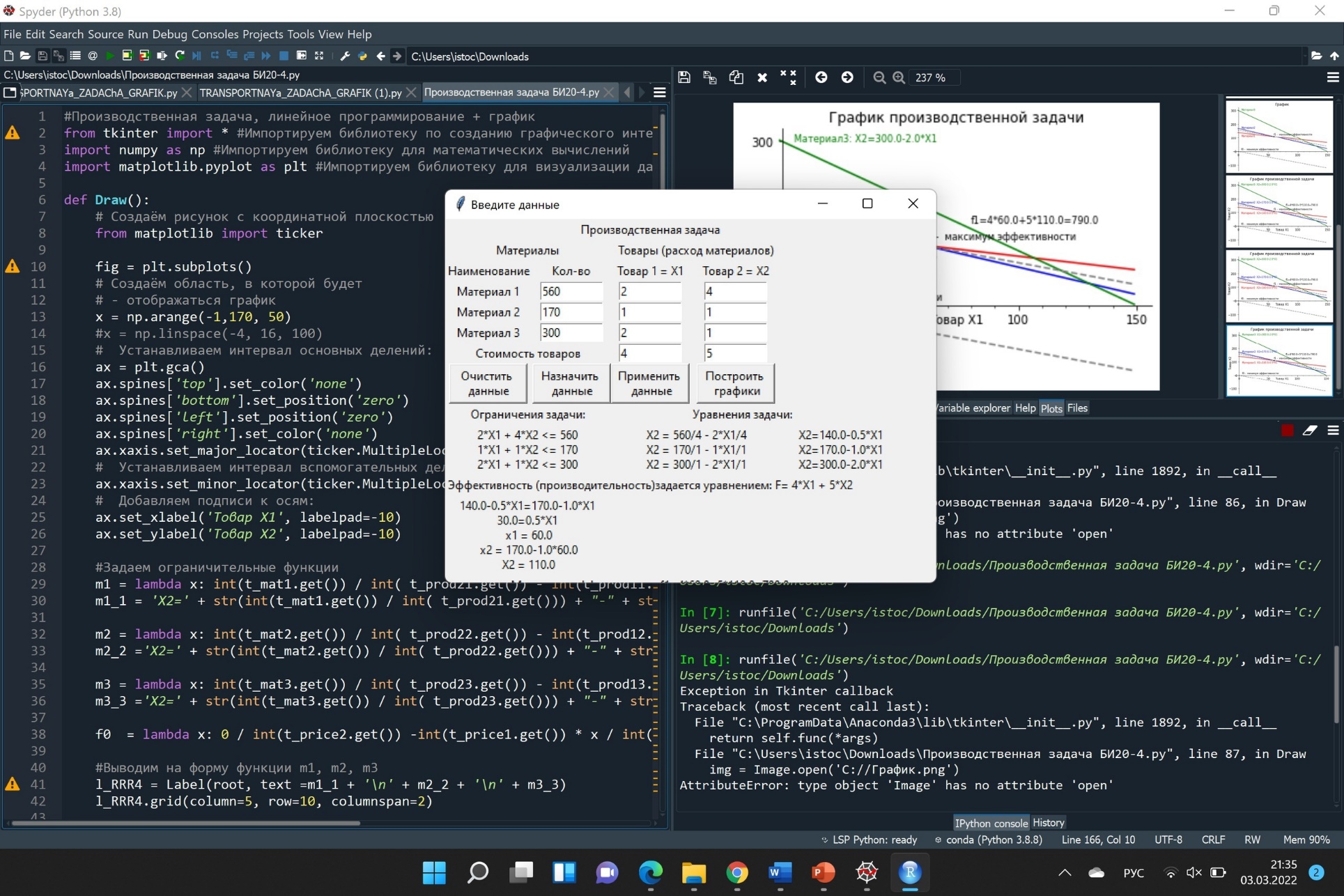
*#Добавим изображение*

*#imm()*

Пример ответа:



1. При запуске программы на экране появляется пользовательский интерфейс, в который вводятся значения матрицы:



* Задаём функцию для очищения всех полей формы (если в поле ввода чисел в матрицу было вписано не то значение, то можно его удалить):

*def Clean():*

*t\_mat1.delete(0,END)*

*t\_mat2.delete(0,END)*

*t\_mat3.delete(0,END)*

*t\_prod11.delete(0,END)*

*t\_prod12.delete(0,END)*

*t\_prod13.delete(0,END)*

*t\_prod21.delete(0,END)*

*t\_prod22.delete(0,END)*

*t\_prod23.delete(0,END)*

*t\_price1.delete(0,END)*

*t\_price2.delete(0,END)*

* Функция для задания значений по умолчанию:

*def Default():*

*Clean()*

*t\_mat1.insert(0, '560')*

*t\_mat2.insert(0, '170')*

*t\_mat3.insert(0, '300')*

*t\_prod11.insert(0, '2')*

*t\_prod12.insert(0, '1')*

*t\_prod13.insert(0, '2')*

*t\_prod21.insert(0, '4')*

*t\_prod22.insert(0, '1')*

*t\_prod23.insert(0, '1')*

*t\_price1.insert(0, '4')*

*t\_price2.insert(0, '5')*

* *Задаём функцию для оформления пользовательского интерфейса, задаём ограничения на графике*

*def RRR(): #Обработка входных данных в виде пояснений на форме*

*l\_RRR = Label(root, text="Ограничения задачи:")*

*l\_RRR.grid(column=0, row=9, columnspan=2)*

*txt\_usl1 = t\_prod11.get() + '\*X1 + ' + t\_prod21.get() + '\*X2 <= ' + t\_mat1.get()*

*txt\_usl2 = t\_prod12.get() +'\*X1 + ' + t\_prod22.get() + '\*X2 <= ' + t\_mat2.get()*

*txt\_usl3 = t\_prod13.get() +'\*X1 + ' + t\_prod23.get() + '\*X2 <= ' + t\_mat3.get()*

*l\_RRR1 = Label(root, text = txt\_usl1 + "\n" + txt\_usl2 + "\n" + txt\_usl3)*

*l\_RRR1.grid(column=0, row=10, columnspan=2)*

*l\_RRR3 = Label(root, text="Уравнения задачи:")*

*l\_RRR3.grid(column=3, row=9, columnspan=2)*

*u1 = 'X2 = ' + t\_mat1.get() + '/' + t\_prod21.get() + " - " + t\_prod11.get() + '\*X1/' + t\_prod21.get()*

*u2 = 'X2 = ' + t\_mat2.get() + '/' + t\_prod22.get() + " - " + t\_prod12.get() + '\*X1/' + t\_prod22.get()*

*u3 = 'X2 = ' + t\_mat3.get() + '/' + t\_prod23.get() + " - " + t\_prod13.get() + '\*X1/' + t\_prod23.get()*

*l\_RRR2 = Label(root, text =u1 + '\n' + u2 + '\n' + u3)*

*l\_RRR2.grid(column=2, row=10, columnspan=2)*

*F = 'F= ' + t\_price1.get() + '\*X1 + ' + t\_price2.get() + '\*X2'*

*l\_RRR3 = Label(root, text="Эффективность (производительность)задается уравнением: " + F)*

*l\_RRR3.grid(column=0, row=11, columnspan=6)*

*root = Tk() #Текущая форма пользовательского интерфейса*

*root.title("Введите данные")*

*root.geometry('500x400') #Размер формы*

*label\_title = Label(root, text="Производственная задача")*

*label\_title.grid(column=0, row=1, columnspan=6)*

*l\_mat0 = Label(root, text="Материалы")*

*l\_mat0.grid(column=0, row=2, columnspan=2)*

*l\_mat01 = Label(root, text="Наименование")*

*l\_mat01.grid(column=0, row=3)*

*l\_mat02 = Label(root, text="Кол-во")*

*l\_mat02.grid(column=1, row=3)*

*l\_mat1 = Label(root, text="Материал 1")*

*l\_mat1.grid(column=0, row=4)*

*l\_mat2 = Label(root, text="Материал 2")*

*l\_mat2.grid(column=0, row=5)*

*l\_mat3 = Label(root, text="Материал 3")*

*l\_mat3.grid(column=0, row=6)*

*t\_mat1 = Entry(root, width=10)*

*t\_mat1.grid(column=1, row=4)*

*t\_mat2 = Entry(root, width=10)*

*t\_mat2.grid(column=1, row=5)*

*t\_mat3 = Entry(root, width=10)*

*t\_mat3.grid(column=1, row=6)*

*l\_prod0 = Label(root, text="Товары (расход материалов)")*

*l\_prod0.grid(column=2, row=2, columnspan=2)*

*l\_prod1 = Label(root, text="Товар 1 = X1")*

*l\_prod1.grid(column=2, row=3)*

*l\_prod2 = Label(root, text="Товар 2 = X2")*

*l\_prod2.grid(column=3, row=3)*

*t\_prod11 = Entry(root, width=10)*

*t\_prod11.grid(column=2, row=4)*

*t\_prod12 = Entry(root, width=10)*

*t\_prod12.grid(column=2, row=5)*

*t\_prod13 = Entry(root, width=10)*

*t\_prod13.grid(column=2, row=6)*

*t\_prod21 = Entry(root, width=10)*

*t\_prod21.grid(column=3, row=4)*

*t\_prod22 = Entry(root, width=10)*

*t\_prod22.grid(column=3, row=5)*

*t\_prod23 = Entry(root, width=10)*

*t\_prod23.grid(column=3, row=6)*

*l\_price = Label(root, text="Стоимость товаров")*

*l\_price.grid(column=0, row=7, columnspan=2)*

*t\_price1 = Entry(root, width=10)*

*t\_price1.grid(column=2, row=7)*

*t\_price2 = Entry(root, width=10)*

*t\_price2.grid(column=3, row=7)*

*#Добавление кнопок на форму*

*button = Button(root, text="Применить \n данные", width=10, command=RRR)*

*button.grid(column=2, row=8)*

*button0 = Button(root, text="Назначить \n данные", width=10, command=Default)*

*button0.grid(column=1, row=8)*

*button1 = Button(root, text="Очистить \n данные", width=10, command=Clean)*

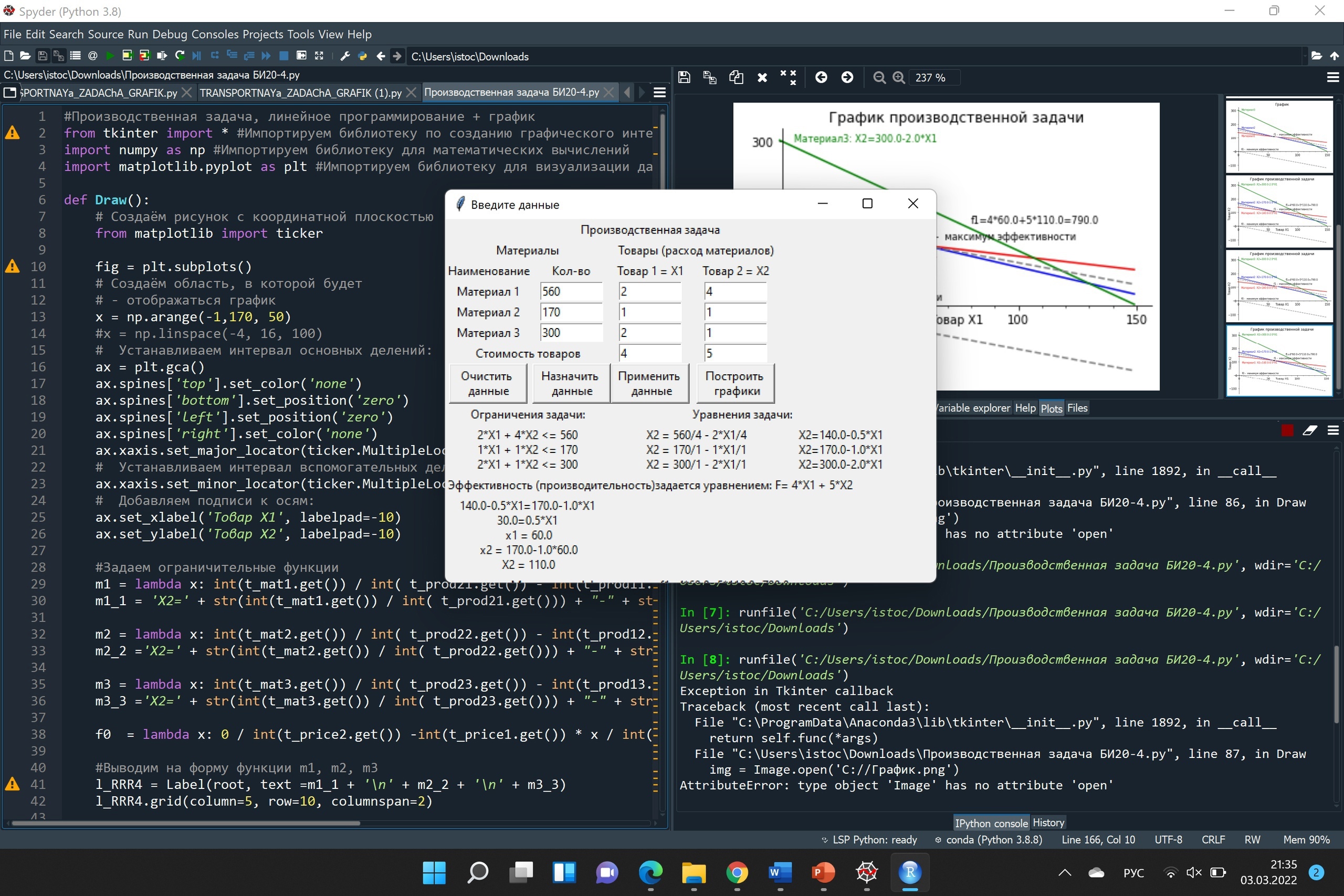
*button1.grid(column=0, row=8)*

*button1 = Button(root, text="Построить \n графики", width=10, command=Draw)*

*button1.grid(column=3, row=8)*

*root.mainloop() #Запуск формы на просмотр пользователю*

Пример ответа:



**Заключение**

С помощью представленного алгоритма удалось решить поставленную перед нами задачу. Для конкретной задачи ответом будет служить максимальная прибыль = 790.

Можно предложить следующий вариант развития данного кода:

1. В нашем коде нет возможности ввода размерности матрицы, что можно предусмотреть в дальнейшем;
2. Обозначить на графике цветом область допустимых значений;
3. Визуализация графика на пользовательском интерфейсе;
4. Отображение ответа - максимальной прибыли не только на графике, но и на пользовательском интерфейсе.

В процессе решения мы поняли, что этот же алгоритм можно применить и для решения таких задач как:

1. Нахождения максимальной прибыли.

\**Издательство выпускает 2 вида книг: детективы и бизнес-литература. Для изготовления книг используется различное количество материалов. Сколько и каких изданий нужно выпустить издательству, чтобы его прибыль была максимальной?*

1. Нахождение максимального дневного потребления продуктов для улучшения состояния.

*\*Гражданин К. сдал кровь и выяснилось, что у него понижен уровень йода, железа и кальция. Врач предоставил таблицу, в которой расписаны овощи и содержащиеся в них полезные вещества (мкг) на кг. Минимальное месячное потребление йода составляет 1500 мкг, железа - 450 мкг, кальция - 2500 мкг. Какой из продуктов будет лучше употреблять гражданину К. для стабильного состояния.*

1. Нахождение максимальное калорийности продуктов

*\*Граждан С. пришел в ресторан быстрого питания "Макдоналдс", чтобы набрать вес попросил у сотрудников лист кбжу для подсчета калорийности продуктов. Лист кбжу содержит все блюда ресторана с указыванием белков, жиров и углеводов. Гражданин С. взял из меню два продукта. Какой продукт имеет больше калорийности для повышения веса.*