

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Математичні методи дослідження операцій»

на тему:

«Розв’язання задачі математичного програмування»

студента III курсу групи ІПЗ-21-
спеціальності 121 «Інженерія програмного
забезпечення»

Цалка Ярослава Володимировича
(прізвище, ім'я та по-батькові)

Керівник ст. викладач кафедри ІПЗ
Локтікова Т.М.

Дата захисту: "" червня 2024 р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____	<u>Локтікова Т.М</u>
(підпис)	(прізвище та ініціали)
_____	<u>Кушнір Н.О.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Полторак В.П</u>
(підпис)	(прізвище та ініціали)

Житомир – 2024

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
В.о. зав. каф. ІПЗ, к.т.н., доц.
_____ Андрій МОРОЗОВ
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ

до курсової роботи

студенту 3-го курсу групи ІПЗ-21-2 Ярославу ЦАЛКУ

- Тема роботи:** Розв'язання задачі математичного програмування (МП).
- Термін здачі** студентом закінченої роботи: червня 2024 р.
- Вихідні дані** до роботи: нехай можливо купити хліб, сало, маргарин, картоплю, яйця та шоколад за ціною $c_1 = 5$ у.о., $c_2 = 30$ у.о., $c_3 = 40$ у.о., $c_4 = 5$ у.о., $c_5 = 180$ у.о., і $c_6 = 400$ у.о. відповідно. Дієта повинна задовольняти обмеженням по білках, жирах і вуглеводах, які містяться в продуктах у таких кількостях (табл.1). Біологічна норма є такою: $b_1 = 6$ у.о., $b_2 = 4$ у.о., $b_3 = 20$ у.о.

Таблиця 1

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад
Білки	0,1	0,2	0	0	0,6	0,6
Жири	0	0,5	0,8	0	0	0,2
Вуглеводи	0,5	0,1	0	0,5	0	0,2

Потрібно визначити, скільки купити хліба, сала, маргарину, картоплі, яєць і шоколаду, щоби виконувалась біологічна норма мінімальної вартості.

- Перелік питань, які мають бути розроблені:** 1) Вступ, 2) Побудова математичної моделі задачі, 3) Вибір й обґрунтування методу розв'язання задачі МП, 4) Розв'язання задачі МП вручну за допомогою вибраного методу, 5) Розробка алгоритму розв'язання задачі МП, 6) Вибір й обґрунтування інструментальних засобів для реалізації програми розв'язання задачі МП, 7) Написання програми розв'язання задачі МП, 8) Тестування програмного продукту, 9) Висновки, 10) Джерела інформації, 11) Додатки.

- Перелік графічного матеріалу:** алгоритм розв'язання задачі МП, скріншоти роботи програми.

- Дата видачі завдання:** « » лютого 2024 р.

Завдання прийняв до виконання студент групи ІПЗ-21-2

Ярослав ЦАЛКО

Керівник курсової роботи, ст. викл. каф. ІПЗ

Тамара ЛОКТИКОВА

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «Розв’язання задачі математичного програмування» складається зі вступу, трьох розділів, висновків, джерел інформації та додатків.

Текстова частина викладена на 34 сторінках друкованого тексту.

Джерела інформації містять 7 найменувань і займають 1 сторінку.

Пояснювальна записка має 7 сторінок додатків. У роботі наведено 8 рисунків.

Загальний обсяг роботи – 41 сторінка.

У першому розділі було проаналізовано завдання та складено математичну модель.

У другому розділі обрано метод розв’язання задачі математичного програмування та розв’язано задачу вручну.

У третьому розділі наведено метод реалізації програми, вибір засобів реалізації, проектування схеми роботи алгоритмів та проведено тестування програмного забезпечення.

Висновки містять у собі результати виконаної роботи з розв’язання задачі математичного програмування та створення програми розв’язання задачі за даними, введеними користувачем.

У додатках представлений лістинг створеної програми.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ВСТУП.....	5
1. ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ ЗАДАЧІ	6
2. РОЗВ'ЯЗАННЯ ЗАДАЧІ МП	9
2.1 Вибір й обґрунтування методу розв'язання задачі	9
2.2 Розв'язання задачі МП	9
3. РОЗРОБКА ПРОГРАМИ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ МП	19
3.1 Аналіз методів реалізації та опис алгоритмів	19
3.2 Вибір й обґрунтування інструментальних засобів для реалізації програмного додатку.....	20
3.3 Розробка програмного додатку	21
3.4 Тестування програмного додатку	28
ВИСНОВКИ.....	32
СПИСОК ДЖЕРЕЛ	33
ДОДАТКИ.....	34

ВСТУП

Математичний аналіз відіграє ключову роль у сучасному науковому дослідженні та прийнятті рішень у різних галузях. Використовуючи математичні та статистичні методи, вчені та фахівці можуть створювати моделі та аналізувати складні системи з метою знаходження оптимальних рішень та оптимізації процесів. Операційний аналіз, зокрема, широко використовується у таких сферах як логістика, транспорт, виробництво та фінанси для оптимізації маршрутів, управління запасами, планування процесів та моделювання портфелів.

Мета цього курсу - дослідження методів вирішення задач математичного програмування на прикладах з практики. Це охоплює розробку математичних моделей, вибір програмних інструментів та аналіз алгоритмів з метою кращого розуміння основних принципів операційного аналізу.

У цьому контексті об'єктом дослідження є самі задачі математичного програмування, а предметом - програмне забезпечення, яке дозволяє їх розв'язувати з інтуїтивно зрозумілим інтерфейсом. Операційний аналіз відіграє важливу роль у розробці ефективних рішень у сучасних умовах.

Незважаючи на складність, операційний аналіз допомагає розуміти та вдосконалювати процеси в суспільстві та бізнесі, стаючи невід'ємною складовою наукового підходу до оптимізації різноманітних аспектів людської діяльності.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

1. ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ ЗАДАЧІ

Нехай можливо купити хліб, сало, маргарин, картоплю, яйця та шоколад за ціною $c_1 = 5$ у.о., $c_2 = 30$ у.о., $c_3 = 40$ у.о., $c_4 = 5$ у.о., $c_5 = 180$ у.о., і $c_6 = 400$ у.о. відповідно. Дієта повинна задовольняти обмеженням по білках, жирах і вуглеводах, які містяться в продуктах у таких кількостях (табл.1).

Біологічна норма є такою: $b_1 = 6$ у.о., $b_2 = 4$ у.о., $b_3 = 20$ у.о.

Таблиця 1

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад
Білки	0,1	0,2	0	0	0,6	0,6
Жири	0	0,5	0,8	0	0	0,2
Вуглеводи	0,5	0,1	0	0,5	0	0,2

Необхідно розрахувати, скільки потрібно купити різних продуктів, таких як хліб, сало, маргарин, картопля, яйця та шоколад, щоби забезпечити мінімальну вартість дієти з погляду біологічної норми.

У цій задачі важливо врахувати кілька ключових факторів, що впливають на результат обчислень:

- Поживна цінність продуктів, виражена у вмісті білків, жирів та вуглеводів;
- Ціна кожного продукту, такого як хліб, сало, маргарин, картопля, яйця та шоколад;
- Біологічна норма, яка визначає необхідну кількість кожного продукту у дієті.

Позначимо x_1 як кількість хліба, x_2 - кількість сала, x_3 – кількість маргарину, x_4 – кількість картоплі, x_5 – кількість яєць та x_6 – кількість шоколаду.

Оскільки дієта повинна забезпечити необхідну кількість білків згідно з біологічною нормою, потрібно встановити обмеження, яке контролюватиме вміст цієї хімічної сполуки. Наприклад, якщо хліб містить 0.1 одиниць білка, сало – 0.2, а маргарин – 0, то нам необхідно врахувати ці значення при формулюванні обмежень.

$$0.1x_1 + 0.2x_2 + 0x_3 + 0x_4 + 0.6x_5 + 0.6x_6 \geq 6$$

Додамо нерівність, яка буде контролювати кількість жирів у продуктах порівняно з бажаною нормою:

$$0x_1 + 0.5x_2 + 0.8x_3 + 0x_4 + 0x_5 + 0.2x_6 \geq 4,$$

Додамо нерівність, яка буде контролювати кількість вуглеводів у продуктах порівняно з бажаною нормою:

$$0.5x_1 + 0.1x_2 + 0x_3 + 0.5x_4 + 0x_5 + 0.2x_6 \geq 20$$

Також важливо враховувати, що вміст будь-якої поживної речовини не може бути від'ємним, що впливає зі смислу умови. Тому наступне обмеження встановить їхню невід'ємність:

$$x_1, x_2 \dots x_6 \geq 0$$

$$F(x_1, x_2 \dots x_6) = 5x_1 + 30x_2 + 40x_3 + 5x_4 + 180x_5 + 400x_6 \rightarrow \min$$

Система рівнянь:

$$\begin{cases} 0.1x_1 + 0.2x_2 + 0x_3 + 0x_4 + 0.6x_5 + 0.6x_6 \geq 6, \\ 0x_1 + 0.5x_2 + 0.8x_3 + 0x_4 + 0x_5 + 0.2x_6 \geq 4, \\ 0.5x_1 + 0.1x_2 + 0x_3 + 0.5x_4 + 0x_5 + 0.2x_6 \geq 20, \\ x_1, x_2 \dots x_6 \geq 0 \end{cases}$$

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

У першому розділі було проаналізовано завдання та отримано математичну модель. Поставлена задача – це задача лінійного програмування, оскільки і цільова функція, і функції-обмеження є лінійними.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

2. РОЗВ'ЯЗАННЯ ЗАДАЧІ МП

2.1 Вибір й обґрунтування методу розв'язання задачі

Після ретельного аналізу різних методів пошуку оптимального рішення для нашої задачі, ми вирішили використати метод штучного базису як найефективніший і найшвидший з усіх варіантів. Цей метод особливо ефективний у випадках, коли маємо справу з лінійними задачами математичного програмування та існують обмеження на ресурси зі знаком більше або дорівнює.

Метод штучного базису засновується на системі лінійних алгебраїчних рівнянь, яка враховує взаємозв'язки між різними змінними та обмеженнями задачі. Цей метод дозволяє знаходити оптимальні розв'язки швидше та ефективніше, порівняно з іншими підходами. Він часто використовується у виробничій сфері, логістиці, транспорті та інших галузях для оптимізації ресурсів та вирішення складних завдань планування.

Таким чином, обраний метод є ідеальним інструментом для нашої задачі і допоможе нам ефективно знайти оптимальні рішення, враховуючи всі обмеження та вимоги, що стоять перед нами.

2.2 Розв'язання задачі МП

$$F(x_1, x_2 \dots x_6) = 5x_1 + 30x_2 + 40x_3 + 5x_4 + 180x_5 + 400x_6 \rightarrow \min$$

$$\begin{cases} 0.1x_1 + 0.2x_2 + 0x_3 + 0x_4 + 0.6x_5 + 0.6x_6 \geq 6, \\ 0x_1 + 0.5x_2 + 0.8x_3 + 0x_4 + 0x_5 + 0.2x_6 \geq 4, \\ 0.5x_1 + 0.1x_2 + 0x_3 + 0.5x_4 + 0x_5 + 0.2x_6 \geq 20, \\ x_1, x_2 \dots x_6 \geq 0 \end{cases}$$

Нерівності обертаються в рівності шляхом введення вільних змінних – x_7, x_8, x_9 відповідно:

$$F(x_1, x_2 \dots x_9) =$$

$$5x_1 + 30x_2 + 40x_3 + 5x_4 + 180x_5 + 400x_6 + 0x_7 + 0x_8 + 0x_9 \rightarrow \min$$

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\begin{cases} 0.1x_1 + 0.2x_2 + 0x_3 + 0x_4 + 0.6x_5 + 0.6x_6 - 1x_7 = 6, \\ 0x_1 + 0.5x_2 + 0.8x_3 + 0x_4 + 0x_5 + 0.2x_6 - 1x_8 = 4, \\ 0.5x_1 + 0.1x_2 + 0x_3 + 0.5x_4 + 0x_5 + 0.2x_6 - 1x_9 = 20, \\ x_1, x_2 \dots x_9 \geq 0 \end{cases}$$

Введемо в задачу 3 штучні змінні:

$$F(x_1, x_2 \dots x_9) =$$

$$5x_1 + 30x_2 + 40x_3 + 5x_4 + 180x_5 + 400x_6 + 0x_7 + 0x_8 + 0x_9 + Mx_{10} + Mx_{11} + Mx_{12} \rightarrow \min$$

$$\begin{cases} 0.1x_1 + 0.2x_2 + 0x_3 + 0x_4 + 0.6x_5 + 0.6x_6 - 1x_7 + 1x_{10} = 6, \\ 0x_1 + 0.5x_2 + 0.8x_3 + 0x_4 + 0x_5 + 0.2x_6 - 1x_8 + 1x_{11} = 4, \\ 0.5x_1 + 0.1x_2 + 0x_3 + 0.5x_4 + 0x_5 + 0.2x_6 - 1x_9 + 1x_{12} = 20, \\ x_1, x_2 \dots x_9 \geq 0 \end{cases}$$

Отримана таблиця:

	C	–	5	30	40	5	180	400	0	0	0	M	M	M
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
M	x_{10}	6	0.1	0.2	0	0	0.6	0.6	-1	0	0	1	0	0
M	x_{11}	4	0	0.5	0.8	0	0	0.2	0	-1	0	0	1	0
M	x_{12}	20	0.5	0.1	0	0.5	0	0.2	0	0	-1	0	0	1
	Δ	30 M	0.6 M- 5	0.8 M- 30	0.8 M- 40	0.5 M- 5	0.6 M- 180	M- 400	-M	-M	-M	0	0	0

Розв'язуємо задачу ЛП на \min , маємо в індексному рядку одну з найдодатніших оцінок $\Delta_6 = M - 400$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_6 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв’язку задачі – x_6 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{6}{0.6}; \frac{4}{0.2}; \frac{20}{0.2} \right\} = \frac{6}{0.6}$$

Отже, напрямний рядок – x_{10} . Змінна x_{10} виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{106} .

Отримана таблиця:

	С	–	5	30	40	5	180	400	0	0	0	М	М
	В	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{11}	A_{12}
400	x_6	10	0.17	0.33	0	0	1	1	-1.67	0	0	0	0
М	x_{11}	2	-0.03	0.43	0.8	0	-0.2	0	0.33	-1	0	1	0
М	x_{12}	18	0.47	0.03	0	0.5	-0.2	0	0.33	0	-1	0	1
	Δ	20М +400 0	0.43 М+6 1.67	0.47 М+1 03.3 3	0.8М -40	0.5М -5	- 0.4М +220	0	0.67 М- 666. 67	-М	-М	0	0

Розв’язуємо задачу ЛП на \min , маємо в індексному рядку одну з найдодатніших оцінок $\Delta_3 = 0.8М - 40$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв’язку задачі. Визначаємо напрямний стовпець – A_3 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв’язку задачі – x_3 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{2}{0.8} \right\} = \frac{5}{2}$$

Отже, напрямний рядок – x_{11} . Змінна x_{11} виводитиметься з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{113} .

	С	–	5	30	40	5	180	400	0	0	0	М
	В	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{12}
400	x_6	10	0.17	0.33	0	0	1	1	-1.67	0	0	0
40	x_3	2.5	-0.04	0.54	1	0	-0.25	0	0.42	-1.25	0	0
М	x_{12}	18	0.47	0.03	0	0.5	-0.2	0	0.33	0	-1	1
	Δ	18М +410 0	0.47 М+6 0	0.03 М+1 25	0	0.5М -5	- 0.2М +210	0	0.33 М- 650	-50	-М	0

Розв'язуємо задачу ЛП на *min*, маємо в індексному рядку одну з найдодатніших оцінок $\Delta_4 = 0.5М - 5$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі. Визначаємо напрямний стовпець – A_4 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_4 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{18}{0.5} \right\} = \frac{18}{0.5}$$

Отже, напрямний рядок – x_{12} . Змінна x_{12} виводитиметься з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{124} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
400	x_6	10	0.17	0.33	0	0	1	1	-1.67	0	0
40	x_3	2.5	-0.04	0.54	1	0	-0.25	0	0.42	-1.25	0
5	x_4	36	0.93	0.07	0	1	-0.4	0	0.67	0	-2
	Δ	4280	64.67	125.3 3	0	0	208	0	- 646.6 7	-50	-10

Розв'язуємо задачу ЛП на *min*, маємо в індексному рядку одну з найдодатніших оцінок $\Delta_5 = 208$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_5 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_5 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{10}{1} \right\} = 10$$

Отже, напрямний рядок – x_6 . Змінна x_6 виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{65} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
180	x_5	10	0.17	0.33	0	0	1	1	-1.67	0	0
40	x_3	5	0	0.63	1	0	0	0.25	0	-1.25	0
5	x_4	40	1	0.2	0	1	0	0.4	0	0	-2
	Δ	2200	30	56	0	0	0	-208	-300	-50	-10

Розв'язуємо задачу ЛП на *min*, маємо в індексному рядку одну з найдодатніших оцінок $\Delta_2 = 56$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_2 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_2 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{10}{0.33}; \frac{5}{0.63}; \frac{40}{0.2} \right\} = \frac{5}{0.63} = 7.94$$

Отже, напрямний рядок – x_3 . Змінна x_3 виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{32} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
180	x_5	7.33	0.17	0	-0.53	0	1	0.87	-1.67	0.67	0
30	x_2	8	0	1	1.6	0	0	0.4	0	-2	0
5	x_4	38.4	1	0	-0.32	1	0	0.32	0	0.4	-2
	Δ	1752	30	0	-89.6	0	0	-230.4	-300	62	-10

Розв'язуємо задачу ЛП на *min*, маємо в індексному рядку одну з найдодатніших оцінок $\Delta_8 = 62$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_8 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_8 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{7,33}{0,67}; \frac{38,4}{0,4} \right\} = \frac{7,33}{0,67}$$

Отже, напрямний рядок – x_5 . Змінна x_5 виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{58} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
0	x_8	11	0.25	0	-0.8	0	1.5	1.3	-2.5	1	0
30	x_2	30	0.5	1	0	0	3	3	-5	0	0
5	x_4	34	0.9	0	0	1	-0.6	-0.2	1	0	-2
	Δ	1070	14.5	0	-40	0	-93	-311	-145	0	-10

Розв'язуємо задачу ЛП на *min*, маємо в індексному рядку одну з найдодатніших оцінок $\Delta_1 = 14.5$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_1 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_1 .

Визначимо напрямний рядок.

$$\text{Min} \left\{ \frac{11}{0.25}; \frac{30}{0.5}; \frac{34}{0.9} \right\} = \frac{34}{0.9}$$

Отже, напрямний рядок – x_4 . Змінна x_4 виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{41} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
0	x_8	1.56	0	0	-0.8	-0.28	1.67	1.36	-2.78	1	0.56
30	x_2	11.11	0	1	0	-0.56	3.33	3.11	-5.56	0	1.11
5	x_1	37.78	1	0	0	1.11	-0.67	-0.22	1.11	0	-2.22
	Δ	522.2 2	0	0	-40	-16.11	-83.33	- 307.7 8	- 161.1 1	0	22.22

Розв'язуємо задачу ЛП на \min , маємо в індексному рядку одну з найдодатніших оцінок $\Delta_9 = 22.22$, а у відповідному стовпці додатні елементи, тому можливий перехід до іншого, більш кращого розв'язку задачі.

Визначаємо напрямний стовпець – A_9 (за найдодатнішою оцінкою), який вказує на змінну, що вводиться в базис для покращення розв'язку задачі – x_9 .

Визначимо напрямний рядок.

$$\min \left\{ \frac{1.56}{0.56}; \frac{11.11}{1.11} \right\} = \frac{1.56}{0.56}$$

Отже, напрямний рядок – x_8 . Змінна x_8 виводиться з базису. На перетині напрямного стовпця та напрямного рядка знаходиться напрямний елемент – x_{89} .

	C	–	5	30	40	5	180	400	0	0	0
	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
0	x_9	2.8	0	0	-1.44	-0.5	3	2.44	-5	1.8	1
30	x_2	8	0	1	1.6	0	0	0.4	0	-2	0
5	x_1	44	1	0	-3.2	0	6	5.2	-10	4	0
	Δ	460	0	0	-8	-5	-150	-362	-50	-40	0

Отриманий план пройшов перевірку на оптимальність, так як в індексному рядку відсутні додатні елементи, задачу розв'язано.

Оптимальний план:

$$x_1 = 44, x_2 = 8, F_{min} = 460.$$

Висновки до розділу 2

У другому розділі було проаналізовано наявні методи та визначено найдоцільніший метод розв'язання задачі МП. Було розв'язано задачу та знайдено розв'язок, що задовольняє поставлені умови.

3. РОЗРОБКА ПРОГРАМИ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ МП

3.1 Аналіз методів реалізації та опис алгоритмів

Схема алгоритму:



Рис. 3.1. Алгоритм

На основі інформації, наведеної на схемі, можна сказати, що програма продовжує виконувати пошук оптимального розв'язку, поки не будуть вилучені всі додатні елементи у найнижчому рядку. Після цього отримані дані збираються у єдиному масиві та виводяться на екран для подальшого перегляду з боку користувача.

3.2 Вибір й обґрунтування інструментальних засобів для реалізації програмного додатку

Для реалізації програмного забезпечення ми вирішили скористатися мовою програмування C#. Обираючи цю мову, ми керувалися кількома ключовими перевагами, які вона пропонує.

По-перше, C# має велике та активне ком'юніті, а також обширну базу ресурсів, уроків та документації. Це робить процес розробки більш доступним та ефективним, оскільки ми можемо легко знайти відповіді на свої питання та корисні поради від досвідчених розробників.

По-друге, C# є частиною платформи .NET, що відкриває доступ до широкого спектру бібліотек та інструментів для розробки. Це дозволяє нам швидко та ефективно реалізовувати різноманітні функції та можливості у наших програмах.

Крім того, мова програмування C# має сучасні можливості, такі як автоматизоване управління пам'яттю та високий рівень безпеки, що робить процес розробки надійним та безпечним.

Отже, обираючи C#, ми отримуємо мову програмування, яка ідеально підходить для створення багатоплатформових застосунків, оскільки платформа .NET підтримує різні операційні системи та пристрої. Це робить її оптимальним вибором для нашого проекту.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

3.3 Розробка програмного додатку

Алгоритмічні розрахунки ініціюються в класі Simplex:

```
public Simplex(Function function, Constraint[] constraints)
{
    if (function.isExtrMax)
    {
        this.function = function;
    }
    else
    {
        this.function = Canonize(function);
    }

    getMatrix(constraints);
    getFunctionArray();
    CalculateBottomRow();

    for (int i = 0; i < F.Length; i++)
    {
        F[i] = -functionVariables[i];
    }
}
```

Розставляються елементи нижнього рядка, прораховуються елементи індексного рядка.

Ініціалізація таблиці також відбувається при старті, формується матриця з відповідними значеннями у кожній комірці:

```
public void getMatrix(Constraint[] constraints)
{
    for (int i = 0; i < constraints.Length; i++)
    {
        if (constraints[i].right < 0)
        {
            double[] negatedLeftVars = new double[constraints[i].left.Length];

            for (int j = 0; j < constraints[i].left.Length; j++)
            {
                negatedLeftVars[j] = -constraints[i].left[j];
            }

            string sign = constraints[i].sign;

            if (sign == ">=")
            {
                sign = "<=";
            }
            else if (sign == "<=")
            {
                sign = ">=";
            }
        }
    }
}
```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Constraint negatedConstraint = new Constraint(negatedLeftVars, -
constraints[i].right, sign);
        constraints[i] = negatedConstraint;
    }
}

double[][] originalMatrix = new double[constraints.First().left.Length][];

for (int i = 0; i < constraints.First().left.Length; i++)
{
    originalMatrix[i] = new double[constraints.Length];
    for (int j = 0; j < constraints.Length; j++)
    {
        originalMatrix[i][j] = constraints[j].left[i];
    }
}

double[][] appendixMatrix = new double[0][];
double[] Bs = new double[constraints.Length];

for (int i = 0; i < constraints.Length; i++)
{
    Constraint current = constraints[i];

    Bs[i] = current.right;

    if (current.sign == ">=")
    {
        appendixMatrix = appendColumn(appendixMatrix, getColumn(-1, i,
constraints.Length));
    }
    else if (current.sign == "<=")
    {
        appendixMatrix = appendColumn(appendixMatrix, getColumn(1, i,
constraints.Length));
    }
}

double[][] newMatrix = new double[constraints.First().left.Length +
appendixMatrix.Length][];

for (int i = 0; i < constraints.First().left.Length; i++)
{
    newMatrix[i] = originalMatrix[i];
}

for (int i = constraints.First().left.Length; i <
constraints.First().left.Length + appendixMatrix.Length; i++)
{
    newMatrix[i] = appendixMatrix[i - constraints.First().left.Length];
}

bool[] hasBasicVar = new bool[constraints.Length];

for (int i = 0; i < constraints.Length; i++)
{
    hasBasicVar[i] = false;
}

C = new int[constraints.Length];

int ci = 0;
for (int i = 0; i < newMatrix.Length; i++)
{

```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

```

bool hasOnlyNulls = true;
bool hasOne = false;
Tuple<int, int> onePosition = new Tuple<int, int>(0, 0);
for (int j = 0; j < constraints.Length; j++)
{
    if (newMatrix[i][j] == 1)
    {
        if (hasOne)
        {
            hasOnlyNulls = false;
            break;
        }
        else
        {
            hasOne = true;
            onePosition = new Tuple<int, int>(i, j);
        }
    }
    else if (newMatrix[i][j] != 0)
    {
        hasOnlyNulls = false;
        break;
    }
}

if (hasOnlyNulls && hasOne)
{
    hasBasicVar[onePosition.Item2] = true;
    C[ci] = onePosition.Item1;
    ci++;
}

m = new bool[newMatrix.Length];

for (int i = 0; i < newMatrix.Length; i++)
{
    m[i] = false;
}

for (int i = 0; i < constraints.Length; i++)
{
    if (!hasBasicVar[i])
    {
        double[] basicColumn = new double[constraints.Length];

        for (int j = 0; j < constraints.Length; j++)
        {
            basicColumn[j] = j == i ? 1 : 0;
        }

        newMatrix = appendColumn(newMatrix, basicColumn);
        m = append(m, true);
        C[ci] = newMatrix.Length - 1;
        ci++;
    }
}

this.b = Bs;
this.matrix = newMatrix;
}
}

```

Далі стартує цикл, що займається пошуком оптимального розв'язку:

```
public Tuple<List<Iteration>, TableAnswerType> GetIterations()
{
    List<Iteration> list = new List<Iteration>();

    var initTable = new Iteration(b, matrix, M, F, C, functionVariables, mBool, m);
    list.Add(initTable);

    MainElement pivot = FindMainElement();
    int i = 0;

    while (pivot.result == TableAnswerType.NoSolve && i < 100)
    {
        CalculateIteration(pivot.index);
        Iteration table = new Iteration(b, matrix, M, F, C, functionVariables,
mBool, m);
        pivot = FindMainElement();

        list.Add(table);
        i++;
    }

    return new Tuple<List<Iteration>, TableAnswerType> (list, pivot.result);
}
```

Якщо поточний план не оптимальний, знаходиться напрямний елемент:

```
MainElement FindMainElement()
{
    int columnMIndex = FindCol(M);

    if (mBool || columnMIndex == -1)
    {
        mBool = true;
        int columnFIndex = FindCol(F);

        if (columnFIndex != -1)
        {
            int rowIndex = FindRow(matrix[columnFIndex], b);

            if (rowIndex != -1)
                return new MainElement(new Tuple<int, int>(columnFIndex, rowIndex),
TableAnswerType.NoSolve);
            else
                return new MainElement(null, TableAnswerType.Unbounded);
        }
        else
            return new MainElement(null, TableAnswerType.Found);
    }
    else
    {
        int rowIndex = FindRow(matrix[columnMIndex], b);

        if (rowIndex != -1)
            return new MainElement(new Tuple<int, int>(columnMIndex, rowIndex),
TableAnswerType.NoSolve);
        else
            return new MainElement(null, TableAnswerType.Unbounded);
    }
}
```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Після цього – йдуть розрахунки наступної таблиці:

```
void CalculateIteration(Tuple<int, int> pivotElementIndices)
{
    double[][] updatedMatrix = new double[matrix.Length][];

    C[pivotElementIndices.Item2] = pivotElementIndices.Item1;

    double[] newRowForJ = new double[matrix.Length];

    for (int i = 0; i < matrix.Length; i++)
    {
        newRowForJ[i] = matrix[i][pivotElementIndices.Item2] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2];
    }

    double[] updatedB = new double[b.Length];

    for (int i = 0; i < b.Length; i++)
    {
        if (i == pivotElementIndices.Item2)
        {
            updatedB[i] = b[i] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2];
        }
        else
        {
            updatedB[i] = b[i] - b[pivotElementIndices.Item2] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2] *
matrix[pivotElementIndices.Item1][i];
        }
    }

    b = updatedB;

    for (int i = 0; i < matrix.Length; i++)
    {
        updatedMatrix[i] = new double[C.Length];
        for (int j = 0; j < C.Length; j++)
        {
            if (j == pivotElementIndices.Item2)
            {
                updatedMatrix[i][j] = newRowForJ[i];
            }
            else
            {
                updatedMatrix[i][j] = matrix[i][j] - newRowForJ[i] *
matrix[pivotElementIndices.Item1][j];
            }
        }
    }

    matrix = updatedMatrix;
    CalculateBottomRow();
}
```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

Всі таблиці збираються в один масив. Далі подана функція виводить їх по одній:

```
void DrawIteration(int id)
{
    ClearTable();

    int addCols = 3;
    grid.ColumnCount = snaps.First().matrix.Length + addCols;
    grid.RowHeadersVisible = false;
    grid.ColumnHeadersVisible = false;

    Iteration snap = snaps[id];

    for (int i = 0; i < snaps.First().matrix.Length + addCols; i++)
    {
        grid.Columns[i].Width = 50;
        grid.Columns[i].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
    }

    string[] firstRow = new string[snaps.First().matrix.Length + addCols];

    firstRow[0] = "C";
    firstRow[1] = "B";
    firstRow[2] = "X0";

    for (int i = addCols; i < snaps.First().matrix.Length + addCols; i++)
    {
        firstRow[i] = $"X{i - (addCols - 1)}";
    }

    grid.Rows.Add(firstRow);

    for (int i = 0; i < snaps.First().C.Length; i++)
    {
        string[] row = new string[snaps.First().matrix.Length + addCols];
        for (int j = 0; j < snaps.First().matrix.Length + addCols; j++)
        {
            if (j == 1)
            {
                row[j] = $"X{snap.C[i] + 1}";
            }
            else if (j == 0)
            {
                row[j] = snap.m[snap.C[i]] ? "-M" :
                $"{Math.Abs(snap.fVars[snap.C[i]])}";
            }
            else if (j == 2)
            {
                row[j] = Round(snap.b[i]).ToString();
            }
            else
            {
                row[j] = Round(snap.matrix[j - addCols][i]).ToString();
            }
        }
        grid.Rows.Add(row);
    }

    string[] fRow = new string[snaps.First().matrix.Length + addCols];

    fRow[1] = "F";
    fRow[2] = Math.Abs(Round(snap.fValue)).ToString();
}
```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

for (int i = addCols; i < snaps.First().matrix.Length + addCols; i++)
{
    fRow[i] = Round(snap.F[i - addCols]).ToString();
}

grid.Rows.Add(fRow);
}

```

Також задля зручності виводиться кінцевий результат, або ж повідомлення, що розв'язків немає за відповідного сценарію:

```

void ShowAnswer(Tuple<List<Iteration>, TableAnswerType> result)
{
    var x1 = Round(GetValueOfX(result.Item1.Last(), 0));
    var x2 = Round(GetValueOfX(result.Item1.Last(), 1));
    var x3 = Round(GetValueOfX(result.Item1.Last(), 2));
    var x4 = Round(GetValueOfX(result.Item1.Last(), 3));
    var x5 = Round(GetValueOfX(result.Item1.Last(), 4));
    var x6 = Round(GetValueOfX(result.Item1.Last(), 5));

    var f = Math.Abs(Round(result.Item1.Last().fValue));

    switch (result.Item2)
    {
        case TableAnswerType.Found:
        {
            resBread.Text = x1.ToString();
            resSalo.Text = x2.ToString();
            resMargarine.Text = x3.ToString();
            resPotato.Text = x4.ToString();
            resEggs.Text = x5.ToString();
            resChoco.Text = x6.ToString();
            resSum.Text = f.ToString();

            labelError.Visible = false;
            break;
        }
        case TableAnswerType.Unbounded:
        case TableAnswerType.NoSolve:
        {
            resBread.Text = "";
            resSalo.Text = "";
            resMargarine.Text = "";
            resPotato.Text = "";
            resEggs.Text = "";
            resChoco.Text = "";
            resSum.Text = "";

            labelError.Visible = true;
            break;
        }
    }
}

```

3.4 Тестування програмного додатку

Меню додатку:

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.0	0.5	0.8	0.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	0.0	0.2	20.0
Ціна	5.0	30.0	40.0	5.0	180.0	400.0	
Результат							

0

Рис. 3.2. Меню

Для запуску обчислень необхідно натиснути на кнопку у верхньому лівому куті:

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.0	0.5	0.8	0.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	0.0	0.2	20.0
Ціна	5.0	30.0	40.0	5.0	180.0	400.0	
Результат	44	8	0	0	0	0	460

1

С	В	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
-М	X10	6	0.1	0.2	0	0	0.6	0.6	-1	0	0	1	0	0
-М	X11	4	0	0.5	0.8	0	0	0.2	0	-1	0	0	1	0
-М	X12	20	0.5	0.2	0	0.5	0	0.2	0	0	-1	0	0	1
	F	0	5	30	40	5	180	400	0	0	0	0	0	0

Рис. 3.3. Розв'язана задача

Також можливо перемикатися між таблицями:

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.0	0.5	0.8	0.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	0.0	0.2	20.0
Ціна	5.0	30.0	40.0	5.0	180.0	400.0	
Результат	44	8	0	0	0	0	460

2

С	В	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
400	X6	10	0.17	0.33	0	0	1	1	-1.67	0	0	1.67	0	0
-M	X11	2	-0.03	0.43	0.8	0	-0.2	0	0.33	-1	0	-0.33	1	0
-M	X12	18	0.47	0.13	0	0.5	-0.2	0	0.33	0	-1	-0.33	0	1
	F	4000	-61.67	-103.33	40	5	-220	0	666.67	0	0	-666.67	0	0

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.0	0.5	0.8	0.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	0.0	0.2	20.0
Ціна	5.0	30.0	40.0	5.0	180.0	400.0	
Результат	44	8	0	0	0	0	460

3

С	В	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
400	X6	10	0.17	0.33	0	0	1	1	-1.67	0	0	1.67	0	0
40	X3	2.5	-0.04	0.54	1	0	-0.25	0	0.42	-1.25	0	-0.42	1.25	0
-M	X12	18	0.47	0.13	0	0.5	-0.2	0	0.33	0	-1	-0.33	0	1
	F	4100	-60	-125	0	5	-210	0	650	50	0	-650	-50	0

Рис. 3.4 – 3.5. Наступні слайди

Відповідь до задачі відповідає отриманій із ручних обрахунків. Отже, складений алгоритм розв'язання працює належним чином. Для повної впевненості було вирішено і аналогічні задачі:

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.3	0.5	0.8	4.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	1.0	0.2	20.0
Ціна	5.0	30.0	40.0	5.0	180.0	400.0	
Результат	60	0	0	0	0	0	300

7

С	В	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
0	X9	10	0	0.8	0	-0.5	2	2.8	-5	0	1	5	0	-1
5	X1	60	1	2	0	0	6	6	-10	0	0	10	0	0
0	X8	14	0	0.1	-0.8	-4	1.8	1.6	-3	1	0	3	-1	0
	F	300	0	20	40	5	150	370	50	0	0	-50	0	0

Калькулятор калорій

Обрахувати

	Хліб	Сало	Маргарин	Картопля	Яйця	Шоколад	Норма
Білки	0.1	0.2	0.0	0.0	0.6	0.6	6.0
Жири	0.3	0.5	0.8	4.0	0.0	0.2	4.0
Вуглеводи	0.5	0.2	0.0	0.5	1.0	0.2	20.0
Ціна	59.0	30.0	40.0	50.0	180.0	400.0	
Результат	0	30	0	28	0	0	2300

8

С	В	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
30	X2	30	0.5	1	0	0	3	3	-5	0	0	5	0	0
50	X4	28	0.8	0	0	1	0.8	-0.8	2	0	-2	-2	0	2
0	X8	123	3.15	0	-0.8	0	4.7	-1.9	5.5	1	-8	-5.5	-1	8
	F	2300	4	0	40	0	50	350	50	0	100	-50	0	-100

Рис. 3.6 – 3.7. Аналогічні задачі

Отже, поставлене завдання було виконане згідно зарання запланованих вимог до ПЗ та із усіма врахуваннями щодо доступного функціоналу для користувача. Отриманий додаток відповідає всім нормам та готвий до використання.

Висновки до розділу 3:

Було спроектовано програмний додаток та побудовано схеми алгоритмів. Обрано C# Windows Forms для реалізації. Програма була протестована, відповідь збігається із відповіддю при ручних підрахунках. Додаток відповідає усім нормам та вимогам, спланованим на початку.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі написання курсового проекту з математичних методів дослідження операцій було використано метод штучного базису. Під час дослідження було досліджено принципи вирішення оптимізаційних завдань, спеціально у контексті виробництва, використовуючи інструменти математичного програмування. Основною частиною роботи було побудування математичної моделі, встановлення обмежень та формулювання цільової функції, після чого був застосований метод штучного базису для вирішення задачі оптимізації. Використання цього методу дозволило досягти оптимального розв'язку та продемонструвати його в рамках курсового проекту.

На етапі проектування програмного забезпечення були визначені основні цілі та вимоги, розроблена загальна схема роботи програми, включаючи метод штучного базису. Для реалізації проекту була використана мова програмування C# та платформа .NET, що забезпечило необхідні інструменти. Ключові методи для роботи з інструментами оптимізації були ретельно вивчені та використані в процесі розробки програмного продукту.

Після розробки програмного забезпечення було проведено тестування для перевірки його працездатності та відповідності змодельованим результатам. У результаті реалізації методу штучного базису та розробки програмного продукту було успішно вирішено поставлену задачу оптимізації, що підкреслює важливість та практичну цінність математичних методів дослідження операцій у розв'язанні складних завдань у сучасних комерційних реаліях та наукових дослідженнях.

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

СПИСОК ДЖЕРЕЛ

1. Локтікова Т., Кушнір Н. Методичні рекомендації
2. Документація по .NET – <https://learn.microsoft.com/uk-ua/dotnet/>

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

Git репозиторій проєкту: <https://github.com/Tsalko-Yaroslav/Coursework-MMDO-tsalko>

```
public Simplex(Function function, Constraint[] constraints)
{
    if (function.isExtrMax)
    {
        this.function = function;
    }
    else
    {
        this.function = Canonize(function);
    }

    getMatrix(constraints);
    getFunctionArray();
    CalculateBottomRow();

    for (int i = 0; i < F.Length; i++)
    {
        F[i] = -functionVariables[i];
    }
}

public void getMatrix(Constraint[] constraints)
{
    for (int i = 0; i < constraints.Length; i++)
    {
        if (constraints[i].right < 0)
        {
            double[] negatedLeftVars = new double[constraints[i].left.Length];

            for (int j = 0; j < constraints[i].left.Length; j++)
            {
                negatedLeftVars[j] = -constraints[i].left[j];
            }

            string sign = constraints[i].sign;

            if (sign == ">=")
            {
                sign = "<=";
            }
            else if (sign == "<=")
            {
                sign = ">=";
            }

            Constraint negatedConstraint = new Constraint(negatedLeftVars, -
constraints[i].right, sign);
            constraints[i] = negatedConstraint;
        }
    }

    double[][] originalMatrix = new double[constraints.First().left.Length][];

    for (int i = 0; i < constraints.First().left.Length; i++)
    {
        originalMatrix[i] = new double[constraints.Length];
        for (int j = 0; j < constraints.Length; j++)
```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            originalMatrix[i][j] = constraints[j].left[i];
        }
    }

    double[][] appendixMatrix = new double[0][];
    double[] Bs = new double[constraints.Length];

    for (int i = 0; i < constraints.Length; i++)
    {
        Constraint current = constraints[i];

        Bs[i] = current.right;

        if (current.sign == ">=")
        {
            appendixMatrix = appendColumn(appendixMatrix, getColumn(-1, i,
constraints.Length));
        }
        else if (current.sign == "<=")
        {
            appendixMatrix = appendColumn(appendixMatrix, getColumn(1, i,
constraints.Length));
        }
    }

    double[][] newMatrix = new double[constraints.First().left.Length +
appendixMatrix.Length][];

    for (int i = 0; i < constraints.First().left.Length; i++)
    {
        newMatrix[i] = originalMatrix[i];
    }

    for (int i = constraints.First().left.Length; i <
constraints.First().left.Length + appendixMatrix.Length; i++)
    {
        newMatrix[i] = appendixMatrix[i - constraints.First().left.Length];
    }

    bool[] hasBasicVar = new bool[constraints.Length];

    for (int i = 0; i < constraints.Length; i++)
    {
        hasBasicVar[i] = false;
    }

    C = new int[constraints.Length];

    int ci = 0;
    for (int i = 0; i < newMatrix.Length; i++)
    {
        bool hasOnlyNulls = true;
        bool hasOne = false;
        Tuple<int, int> onePosition = new Tuple<int, int>(0, 0);
        for (int j = 0; j < constraints.Length; j++)
        {
            if (newMatrix[i][j] == 1)
            {
                if (hasOne)
                {
                    hasOnlyNulls = false;
                    break;
                }
            }
            else

```

```

        {
            hasOne = true;
            onePosition = new Tuple<int, int>(i, j);
        }
    }
    else if (newMatrix[i][j] != 0)
    {
        hasOnlyNulls = false;
        break;
    }
}

if (hasOnlyNulls && hasOne)
{
    hasBasicVar[onePosition.Item2] = true;
    C[ci] = onePosition.Item1;
    ci++;
}
}

m = new bool[newMatrix.Length];

for (int i = 0; i < newMatrix.Length; i++)
{
    m[i] = false;
}

for (int i = 0; i < constraints.Length; i++)
{
    if (!hasBasicVar[i])
    {
        double[] basicColumn = new double[constraints.Length];

        for (int j = 0; j < constraints.Length; j++)
        {
            basicColumn[j] = j == i ? 1 : 0;
        }

        newMatrix = appendColumn(newMatrix, basicColumn);
        m = append(m, true);
        C[ci] = newMatrix.Length - 1;
        ci++;
    }
}

this.b = Bs;
this.matrix = newMatrix;
}

}

public Tuple<List<Iteration>, TableAnswerType> GetIterations()
{
    List<Iteration> list = new List<Iteration>();

    var initTable = new Iteration(b, matrix, M, F, C, functionVariables, mBool, m);
    list.Add(initTable);

    MainElement pivot = FindMainElement();
    int i = 0;

    while (pivot.result == TableAnswerType.NoSolve && i < 100)
    {
        CalculateIteration(pivot.index);
    }
}

```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Iteration table = new Iteration(b, matrix, M, F, C, functionVariables,
mBool, m);
        pivot = FindMainElement();

        list.Add(table);
        i++;
    }

    return new Tuple<List<Iteration>, TableAnswerType> (list, pivot.result);
}

MainElement FindMainElement()
{
    int columnMIndex = FindCol(M);

    if (mBool || columnMIndex == -1)
    {
        mBool = true;
        int columnFIndex = FindCol(F);

        if (columnFIndex != -1)
        {
            int rowIndex = FindRow(matrix[columnFIndex], b);

            if (rowIndex != -1)
                return new MainElement(new Tuple<int, int>(columnFIndex, rowIndex),
TableAnswerType.NoSolve);
            else
                return new MainElement(null, TableAnswerType.Unbounded);
        }
        else
            return new MainElement(null, TableAnswerType.Found);
    }
    else
    {
        int rowIndex = FindRow(matrix[columnMIndex], b);

        if (rowIndex != -1)
            return new MainElement(new Tuple<int, int>(columnMIndex, rowIndex),
TableAnswerType.NoSolve);
        else
            return new MainElement(null, TableAnswerType.Unbounded);
    }
}

void CalculateIteration(Tuple<int, int> pivotElementIndices)
{
    double[][] updatedMatrix = new double[matrix.Length][];

    C[pivotElementIndices.Item2] = pivotElementIndices.Item1;

    double[] newRowForJ = new double[matrix.Length];

    for (int i = 0; i < matrix.Length; i++)
    {
        newRowForJ[i] = matrix[i][pivotElementIndices.Item2] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2];
    }

    double[] updatedB = new double[b.Length];

```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        for (int i = 0; i < b.Length; i++)
        {
            if (i == pivotElementIndices.Item2)
            {
                updatedB[i] = b[i] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2];
            }
            else
            {
                updatedB[i] = b[i] - b[pivotElementIndices.Item2] /
matrix[pivotElementIndices.Item1][pivotElementIndices.Item2] *
matrix[pivotElementIndices.Item1][i];
            }
        }

        b = updatedB;

        for (int i = 0; i < matrix.Length; i++)
        {
            updatedMatrix[i] = new double[C.Length];
            for (int j = 0; j < C.Length; j++)
            {
                if (j == pivotElementIndices.Item2)
                {
                    updatedMatrix[i][j] = newRowForJ[i];
                }
                else
                {
                    updatedMatrix[i][j] = matrix[i][j] - newRowForJ[i] *
matrix[pivotElementIndices.Item1][j];
                }
            }

            matrix = updatedMatrix;
            CalculateBottomRow();
        }

void DrawIteration(int id)
{
    ClearTable();

    int addCols = 3;
    grid.ColumnCount = snaps.First().matrix.Length + addCols;
    grid.RowHeadersVisible = false;
    grid.ColumnHeadersVisible = false;

    Iteration snap = snaps[id];

    for (int i = 0; i < snaps.First().matrix.Length + addCols; i++)
    {
        grid.Columns[i].Width = 50;
        grid.Columns[i].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
    }

    string[] firstRow = new string[snaps.First().matrix.Length + addCols];

    firstRow[0] = "C";
    firstRow[1] = "B";
    firstRow[2] = "X0";

    for (int i = addCols; i < snaps.First().matrix.Length + addCols; i++)

```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        firstRow[i] = $"X{i - (addCols - 1)}";
    }

    grid.Rows.Add(firstRow);

    for (int i = 0; i < snaps.First().C.Length; i++)
    {
        string[] row = new string[snaps.First().matrix.Length + addCols];
        for (int j = 0; j < snaps.First().matrix.Length + addCols; j++)
        {
            if (j == 1)
            {
                row[j] = $"X{snap.C[i] + 1}";
            }
            else if (j == 0)
            {
                row[j] = snap.m[snap.C[i]] ? "-M" :
                $"{Math.Abs(snap.fVars[snap.C[i]])}";
            }
            else if (j == 2)
            {
                row[j] = Round(snap.b[i]).ToString();
            }
            else
            {
                row[j] = Round(snap.matrix[j - addCols][i]).ToString();
            }
        }
        grid.Rows.Add(row);
    }

    string[] fRow = new string[snaps.First().matrix.Length + addCols];

    fRow[1] = "F";
    fRow[2] = Math.Abs(Round(snap.fValue)).ToString();

    for (int i = addCols; i < snaps.First().matrix.Length + addCols; i++)
    {
        fRow[i] = Round(snap.F[i - addCols]).ToString();
    }

    grid.Rows.Add(fRow);
}

```

```

void ShowAnswer(Tuple<List<Iteration>, TableAnswerType> result)
{
    var x1 = Round(GetValueOfX(result.Item1.Last(), 0));
    var x2 = Round(GetValueOfX(result.Item1.Last(), 1));
    var x3 = Round(GetValueOfX(result.Item1.Last(), 2));
    var x4 = Round(GetValueOfX(result.Item1.Last(), 3));
    var x5 = Round(GetValueOfX(result.Item1.Last(), 4));
    var x6 = Round(GetValueOfX(result.Item1.Last(), 5));

    var f = Math.Abs(Round(result.Item1.Last().fValue));

    switch (result.Item2)
    {
        case TableAnswerType.Found:
            {
                resBread.Text = x1.ToString();
                resSalo.Text = x2.ToString();
                resMargarine.Text = x3.ToString();
            }
    }
}

```

		Цалко Я.В.			Житомирська політехніка. 24.121.7.000 – ПЗ	Арк.
		Локтікова Т.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        resPotato.Text = x4.ToString();
        resEggs.Text = x5.ToString();
        resChoco.Text = x6.ToString();
        resSum.Text = f.ToString();

        labelError.Visible = false;
        break;
    }
    case TableAnswerType.Unbounded:
    case TableAnswerType.NoSolve:
    {
        resBread.Text = "";
        resSalo.Text = "";
        resMargarine.Text = "";
        resPotato.Text = "";
        resEggs.Text = "";
        resChoco.Text = "";
        resSum.Text = "";

        labelError.Visible = true;
        break;
    }
}
}

```