

Полтавський університет економіки і торгівлі

Навчально-науковий інститут денної освіти

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

(підпис)

«___» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**«АЛГОРИТМІЧНА РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ІНТЕРАКТИВНОГО КОМПОНЕНТА ДИСТАНЦІЙНОГО КУРСУ
«МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ» НА
ПРИКЛАДІ МОДИФІКОВАНОГО СИМПЛЕКС-МЕТОДУ»»**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Литвиненко Олександр Сергійович

_____ «___» _____ 202_ р.

(підпис)

Науковий керівник к. ф.-м. н., доцент, Ольховська Олена Володимирівна

_____ «___» _____ 202_ р.

(підпис)

Рецензент _____

ПОЛТАВА 2026

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	2
ВСТУП	5
1 ПОСТАНОВКА ЗАДАЧІ	8
1.1 Модифікований симплекс-метод.....	8
1.1.1 Основні формули МСМ	8
1.1.2 Алгоритм модифікованого симплекс-методу.....	9
1.2 Основні вимоги до програмного продукту, який реалізує інтерактивний компонент з теми «Модифікований симплекс-метод» ...	13
2 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	15
2.1 Дистанційні технології навчання та їх складові	15
2.2 Огляд технологій розробки інтерактивних компонентів для Web-середовища.....	17
2.2.1 Flash-технології.....	17
2.2.2 Java-технології	18
2.3 Огляд відомих інтерактивних компонентів з математичних дисциплін	19
3 ТЕОРЕТИЧНА ЧАСТИНА	24
3.1 Алгоритмізація інтерактивного компонента.....	24
3.2 Блок-схема алгоритму роботи інтерактивного компонента.	44
3.3 Обґрунтування вибору програмного забезпечення для реалізації інтерактивного компонента	51
4 ПРАКТИЧНА ЧАСТИНА	54
4.1 Опис програмного продукту	54
4.2 Інструкція по використанню.....	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТОК А Вихідний код програмної реалізації інтерактивного компонента.....	61
ДОДАТОК Б Інструкція по використанню	80

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ
СКОРОЧЕНЬ І ТЕРМІНІВ**

Скорочення, позначення, терміни	Пояснення
ВНЗ	Вищий навчальний заклад
ЗЛП	Задача лінійного програмування
МОДО	Методи оптимізації і дослідження операцій
МП	Мале підприємство
МСМ	Модифікований симплекс-метод
ПК	Персональний комп'ютер, електронна обчислювальна машина, призначена для особистого використання, ціна, розміри та можливості якої задовольняють потреби багатьох людей.
Програмна платформа Java	Ряд програмних продуктів і специфікацій, котрі сумісно надають систему для розробки прикладного програмного забезпечення і вбудовування її в будь-яке крос-платформенне програмне забезпечення.
ПУЕТ	Полтавський університет економіки і торгівлі
ТОВ	Товариство з обмеженою відповідальністю, організація, суб'єкт господарювання, статутний капітал якої поділений на частки, розмір яких встановлюється статутом товариства.
AWT	Abstract Window Toolkit, абстрактний віконний інтерфейс – оригінальний пакет класів мови програмування Java, що слугує для створення графічного інтерфейсу користувача.
IDE	Integrated Development Environment, інтегроване середовище розробки – це комп'ютерна програма, що

Скорочення, позначення, терміни	Пояснення
	допомагає програмістові розробляти нове програмне забезпечення чи модифікувати (удосконалювати) вже існуюче.
Java	Об'єктно-орієнтована мова програмування, основний компонент програмної платформи Java.
Java-апплет	Програмний компонент в двійковому коді віртуальної машини Java, виконується у вікні браузера Web-сторінок
JIT-компіляція	Just-in-time compilation, компіляція «на льоту» – це технологія збільшення продуктивності програмних систем, що використовують байт-код, шляхом трансляції байт-коду в машинний код безпосередньо під час роботи програми.
JRE	Java Runtime Environment, мінімальна реалізація віртуальної машини, що необхідна для виконання Java-додатків, без компілятора чи інших засобів розробки.
JVC	Java Foundation Classes, бібліотека базових класів Java.
JVM	Java Virtual Machine, віртуальна машина Java – набір комп'ютерних програм та структур даних, що використовують модель віртуальної машини для виконання інших комп'ютерних програм чи скриптів.
LMS Moodle	Learning Management System «Moodle», система керування навчанням «Moodle» – модульне об'єктно-орієнтоване динамічне навчальне середовище, яке може використовуватися як платформа для

Скорочення, позначення, терміни	Пояснення
	електронного, в тому числі дистанційного навчання.
Off-line	Можливість спілкування з користувачем в мережі, шляхом відправки повідомлень (текст, графічні зображення або мультимедіа), з подальшим їх отриманням користувачем, як тільки той з'явиться в мережі.
On-line	Можливість спілкування з користувачем в мережі у режимі реального часу засобами аудіо-відео сервісів або сервісів відправки текстових повідомлень.
RMI	Remote Method Invocation, програмний інтерфейс виклику віддалених методів у мові Java.
Swing	Інструментарій для створення графічного інтерфейсу користувача мовою програмування Java.
TCP/IP	Transmission Control Protocol / Internet Protocol, протокол керування передачею / міжмережевий протокол
URL	Uniform Resource Locator, уніфікований локатор ресурсів – стандартизована адреса певного ресурсу (такого як документ, чи зображення) в Інтернеті (чи деінде).

ВСТУП

Одним із шляхів розв'язання проблеми підвищення рівня і якості навчання та зменшення аудиторного навантаження, а також широкого використання дистанційної форми навчання, є впровадження в освіту інноваційних технологій, які дозволяють істотно підвищувати ефективність навчального процесу [1].

Застосування розробленої методики інтерактивних компонентів у навчальних дисциплінах для дистанційного навчання дозволяє вирішити проблему впровадження інформаційних дистанційних технологій у навчально-виховний процес ВНЗ з математичних, гуманітарних, технічних і технологічних напрямів та спеціальностей підготовки.

Інтерактивні компоненти призначені для формування у студентів практичних навичок в результаті спеціальної вправи, вони виконують такі завдання, як перевірка роботи без втручання викладача на машині (механізмі), верстаті, електричному приладі, літаку, пароплаві і т.д., при вивченні структури. Тому у дистанційному навчанні інтерактивні компоненти є оптимальним способом реалізації навчального процесу, де студент не прив'язаний до місця й часу навчання, має постійний доступ до навчальних матеріалів та можливість спілкування із викладачами не тільки очно, а й online і off-line за допомогою Інтернет.

Для розробки інтерактивних компонентів широко використовується об'єктно-орієнтована мова програмування Java.

Програмна реалізація інтерактивного компонента для теми «Модифікований симплекс-метод» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» віртуального навчального середовища ПУЕТ забезпечить відповідність дистанційного навчального курсу згідно з пунктом 3.6, 4.5, 5.3 положення [2].

Актуальність теми. Відповідно до [3], на базі Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі», було створено електронне інформаційне навчально-наукове

середовище для дистанційного навчання [4]. У [2] визначено, що до складу системотехнічного забезпечення дистанційного навчання входять інтерактивні компоненти. Отже, алгоритмічна розробка та програмна реалізація інтерактивного компонента дистанційного курсу «Методи оптимізації та дослідження операцій (МОДО)» на прикладі модифікованого симплекс-методу» є актуальною, оскільки такого інтерактивного компонента не має.

Мета розробки. Розробити алгоритм, програму, перевірити її та впровадити інтерактивний компонент з теми «Модифікований симплекс-метод» для дистанційного навчального курсу «Методи оптимізації та дослідження операцій» з метою підвищити якість професійно-практичної підготовки студентів.

Завдання розробки. Розробити алгоритм, програму, перевірити її та впровадити інтерактивний компонент з теми «Модифікований симплекс-метод» для дистанційного навчального курсу «Методи оптимізації та дослідження операцій» для реалізації розв'язування ЗЛП модифікованим симплекс-методом у вигляді тренінгу.

Об'єкт розробки. Об'єктом розробки є технології Java при створенні інтерактивних компонентів.

Предмет розробки. Предмет розробки є технології Java при створенні інтерактивних компонентів з теми «Модифікований симплекс-метод» для електронного інформаційного навчально-наукового середовища дистанційного навчання.

Методи розробки. Засоби мови програмування Java. Методи розробки клієнтського Java-модуля в IDE Apache NetBeans. Методи оптимізації, зокрема модифікований симплекс-метод.

Практична новизна одержаних результатів. Практичною новизною одержаних результатів є використання Java-бібліотеки «JLaTeXMath» для відображення математичних формул та символів в інтерактивному компоненті, а також алгоритмічна розробка та програмна реалізація інтерактивного компонента дистанційного курсу

«Методи оптимізації та дослідження операцій» на прикладі модифікованого симплекс-методу»».

Практичне значення роботи, отриманих результатів. Алгоритмічна розробка та програмна реалізація інтерактивного компонента на прикладі «Модифікованого симплекс-методу» дистанційного навчального курсу «МОДО» дозволить студентам на практиці засвоювати навички розв'язування задач лінійного програмування модифікованим симплекс-методом. Як наслідок, це підвищить якість професійно-практичної підготовки студентів.

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Модифікований симплекс-метод

1.1.1 Основні формули МСМ

Для більш ефективної організації обчислювального процесу симплекс-метод використовують в іншій, так званій модифікованій формі [5, 6].

В алгоритмі симплекс-методу суттєвим є аналіз оцінок

$$\Delta_j = z_j - c_j = \sum_{k=1}^m c_{i_k} a_{i_k j} - c_j, \quad (0.1)$$

де i_1, i_2, \dots, i_m – номери базисних векторів,

c_j – коефіцієнт цільової функції біля змінної x_j ,

$a_{i_1 j}, \dots, a_{i_m j}$ – коефіцієнти розкладання векторів P_j по векторам базису $P_{i_1}, P_{i_2}, \dots, P_{i_m}$, тобто

$$\begin{pmatrix} a_{i_1 j} \\ \dots \\ a_{i_m j} \end{pmatrix} = B^{-1} \cdot P_j, \quad (0.2)$$

де B^{-1} – матриця обернена до матриці $B = (P_{i_1}, P_{i_2}, \dots, P_{i_m})$ – матриці, складеної з векторів базису.

Оцінки знаходять на кожній ітерації симплекс-методу. Для цього на кожній ітерації знаходять $a_{i_1 j}, \dots, a_{i_m j}$. В модифікованому симплекс-методі такої необхідності немає, тому, що на кожній ітерації обчислюють вектор

$$\Omega = C_\delta B^{-1}, \quad (0.3)$$

де $C_\delta = (c_{i_1}, \dots, c_{i_m})$.

З (0.1) та (0.2) маємо

$$\Delta_j = C_\delta B^{-1} P_j - c_j \quad (0.4)$$

або

$$\Delta_j = \Omega P_j - c_j \quad (0.5)$$

Таким чином, замість знаходження на кожній ітерації для кожного вектора P_j його координат в базисі P_i, \dots, P_m за (0.2), а потім для знаходження Δ_j за (0.4) множити C_δ на $B^{-1}P_j$, можна

- 1) спочатку знайти Ω (за (0.3)),
- 2) потім за (0.5) для кожного P_j знайти Δ_j .

В багатьох випадках модифікований таким чином алгоритм симплекс-методу є більш ефективним (економним з точки зору кількості операцій).

1.1.2 Алгоритм модифікованого симплекс-методу

Нехай є ЗЛП в канонічній формі:

$$\sum_{j=1}^n c_j x_j \rightarrow \max$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m;$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n,$$

тут $b_i > 0 \quad \forall i = 1, 2, \dots, m$.

Нехай знайдено базисний допустимий розв'язок, що визначається базисом P_i, \dots, P_m . Можемо утворити матрицю $B = (P_i, \dots, P_m)$ і знайти до неї обернену B^{-1} .

Результати обчислень зручно представляти у вигляді таблиць. Розглядають основні та допоміжну таблиці. Допоміжну таблицю використовують при переході від однієї до іншої основної таблиць.

Розглянемо побудову таблиць. Почнемо з допоміжної.

Допоміжна таблиця схожа на звичайну симплекс-таблицю. На відміну від неї має додаткові рядки (з номерами $m + 1, m + 2, \dots, m + k$), де записують значення $\Delta_1^{(1)}, \dots, \Delta_n^{(1)}; \Delta_1^{(2)}, \dots, \Delta_n^{(2)}; \dots; \Delta_1^{(k)}, \dots, \Delta_n^{(k)}$, що отримують в

процесі розв'язування задачі. Крім додаткових рядків справа до таблиці приписують додаткові k стовпців для координат векторів $\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(k)}$, що розраховують при розв'язуванні задачі. Таким чином, допоміжна таблиця має вигляд табл. 1.1.

Таблиця 1.1 Вигляд допоміжної таблиці в МСМ

i	Базис	C_δ	P_0	c_1	c_2	...	c_n	$\Omega^{(1)}$	$\Omega^{(2)}$...	$\Omega^{(k)}$
				P_1	P_2	...	P_n				
1	P_{i_1}	C_{i_1}						$\omega_1^{(1)}$	$\omega_1^{(2)}$...	$\omega_1^{(k)}$
2	P_{i_2}	C_{i_2}						$\omega_2^{(1)}$	$\omega_2^{(2)}$...	$\omega_2^{(k)}$
...
m	P_{i_m}	C_{i_m}						$\omega_m^{(1)}$	$\omega_m^{(2)}$		$\omega_m^{(k)}$
m+1	$\Delta_j^{(1)}$			$\Delta_1^{(1)}$	$\Delta_2^{(1)}$...	$\Delta_n^{(1)}$				
m+2	$\Delta_j^{(2)}$			$\Delta_1^{(2)}$	$\Delta_2^{(2)}$...	$\Delta_n^{(2)}$				
...				
m+k	$\Delta_j^{(k)}$			$\Delta_1^{(k)}$	$\Delta_2^{(k)}$...	$\Delta_n^{(k)}$				

Основна таблиця (табл. 1.2) відрізняється від звичайної симплекс-таблиці тим що:

- 1) в таблицю записують стовпці A_i – стовпці з матриці B^{-1} (а не вектори умов P_j);
- 2) в $m + 1$ рядку записують координати вектора Ω (а не оцінки Δ_j);
- 3) табл. 1.2 має додатковий стовпець P_s^* – координати вектора P_s в базисі $P_{i_1}, P_{i_2}, \dots, P_{i_m}$ – вектора, який потрібно ввести в базис на наступній ітерації.

Основна таблиця (табл. 1.2) схожа на звичайну симплекс-таблицю тим, що має $m + 1$ рядок, перші стовпці: i (номер рядка), базис та вектори C_δ, P_0 .

Визначення вектора P_s^* проводять так:

- 1) знаходять вектор $\Omega^{(1)} = C_\delta \cdot B^{-1}$ (за (0.3)), тобто i -та компонента $\Omega^{(1)}$ – це скалярний добуток $(C_\delta, A_i) = \omega_i^{(1)}$.

- 2) знайдені координати вектора $\Omega^{(1)}$ записують в останній рядок основної таблиці МСМ (табл. 1.2), а в допоміжній таблиці МСМ (табл. 1.1) – в поточний останній стовпець $\Omega^{(1)}$;
- 3) за формулою (0.5) ($\Delta_j = \Omega P_j - c_j$) знаходять $\Delta_j^{(1)}$, записують їх в поточний останній рядок допоміжної таблиці МСМ;
- 4) перевіряють розв'язок на оптимальність: якщо для всіх $j=1, 2, \dots, n$, $\Delta_j^{(1)} \geq 0$ – розв'язок оптимальний;
- 5) якщо існує $\Delta_j^{(1)} < 0$, то задача або нерозв'язна, або можна перейти до нового базисного розв'язку зі значенням цільової функції не меншим ніж ϵ .

Таблиця 1.2 Основна таблиця МСМ

i	Базис	C_δ	P_0	A_1	A_2	...	A_m	P_s^*
1	P_{i_1}	C_{i_1}	b_{i_1}	α_{11}	α_{12}	...	α_{1m}	β_{i_1s}
2	P_{i_2}	C_{i_2}	b_{i_2}	α_{21}	α_{22}	...	α_{2m}	β_{i_2s}
...
r	P_{i_r}	C_{i_r}	b_{i_r}	α_{r1}	α_{r2}	...	α_{rm}	β_{i_rs} ←
...
m	P_{i_m}	C_{i_m}	b_{i_m}	α_{m1}	α_{mr}	...	α_{mm}	β_{i_ms}
m+1	$\Omega^{(1)}$ →		F_0	$\omega_1^{(1)}$	$\omega_2^{(1)}$...	$\omega_m^{(1)}$	$\Delta_s^{(1)}$ ↑

$$\Omega^{(1)} = (\omega_1^{(1)}, \dots, \omega_m^{(1)}) \quad \Delta_s^{(1)} < 0$$

Для перевірки (аналізу) цієї ситуації вибирають

$$\max_{\substack{1 \leq j \leq n \\ \Delta_j^{(1)} < 0}} |\Delta_j^{(1)}| = |\Delta_s^{(1)}|,$$

якщо таких s декілька – вибирають довільне. Так вибирають номер s стовпця (вектора) P_s (P_s^*) для запису в останній стовпець основної таблиці МСМ (в перші m рядків);

б) Для обчислення координат цього вектор-стовпця в поточному базисі беруть матрицю B^{-1} (стовпці A_1, \dots, A_m з таблиці 1.2) і множать на вектор P_s (з таблиці 1.1). Так визначають вектор-стовпець P_s^* з основної таблиці МСМ.

Аналіз чи нерозв'язна задача (чи необхідний подальший перерахунок таблиць) роблять на основі аналізу координат $\beta_{i,s}$ вектора P_s^* з табл. 1.2. Якщо всі $\beta_{i,s} \leq 0$, $k=1, 2, \dots, m$, то задача **нерозв'язна**. Якщо існує $\beta_{i,s} > 0$, то переходять до нового базисного розв'язку. Визначають

$$\theta = \min_{\substack{\beta_{i,s} > 0 \\ 1 \leq k \leq m}} \frac{b_{i_k}}{\beta_{i_k s}} = \frac{b_{i_r}}{\beta_{i_r s}}.$$

Тобто рядок r – напрямний, вектор P_s^* (в табл. 1.2) – напрямний. В базисі P_i заміняють на P_s . Розв'язальним елементом є $\beta_{i,s}$. **Перераховують** основну таблицю МСМ, переходячи до нової основної таблиці. Для цього перші m рядків стовпців P_0, A_1, \dots, A_m та F_0 переобчислюють за звичайними правилами симплекс-метода.

Потім знаходять, як описано, Ω (позначимо його $\Omega^{(2)}$) потім $\Delta_j^{(2)}$, $j = 1, 2, \dots, n$, і так далі.

Продовжуючи ітераційний процес, за певну кількість кроків або встановлюємо нерозв'язність задачі, або знаходимо її розв'язок.

Коротко алгоритм МСМ вкладається в такі етапи:

1. знаходження базисного розв'язку ЗЛП, базису P_{i_1}, \dots, P_{i_m} ;
2. обчислення B^{-1} для $B = (P_{i_1}, \dots, P_{i_m})$;
3. обчислення $\Omega = C_s B^{-1}$;
4. обчислення $\Delta_j = \Omega P_j - c_j$;
5. перевірка: якщо всі $\Delta_j \geq 0$, то отримано оптимальний розв'язок.

Якщо існує $\Delta_j < 0$, то знаходять s з умови $|\Delta_s| = \max_{\substack{\Delta_j < 0 \\ 1 \leq j \leq n}} |\Delta_j|$;

6. обчислюємо в базисі P_{i_1}, \dots, P_{i_m} координати вектора

$$P^*_s = \begin{pmatrix} \beta_{i_1 s} \\ \dots \\ \beta_{i_m s} \end{pmatrix};$$

7. перевірка: якщо всі $\beta_{i_k s} \leq 0$, $k = 1, 2, \dots, m$, то цільова функція задачі необмежена. Якщо існує $\beta_{i_r s} > 0$, то знаходять

$$\theta = \min_{\substack{\beta_{i_k s} > 0 \\ 1 \leq k \leq m}} \frac{b_{i_k}}{\beta_{i_k s}} = \frac{b_{i_r}}{\beta_{i_r s}}.$$

8. за допомогою розв'язального елемента $\beta_{i_r s}$ (напрямний рядок r ; напрямний стовпець s) переобчислюють основну таблицю МСМ при цьому знаходяться:
- 1) компоненти нового базисного розв'язку;
 - 2) матриця B^{-1} , обернена до нової матриці B (що складена з нового базису)). Перехід на крок 3.

1.2 Основні вимоги до програмного продукту, який реалізує інтерактивний компонент з теми «Модифікований симплекс-метод»

Програмний продукт, який реалізує інтерактивний компонент з теми «Модифікований симплекс-метод» повинен відповідати наступним вимогам:

- 1) реалізовувати алгоритм роботи інтерактивного компонента;
- 2) бути реалізованим у відповідності до парадигми об'єктно-орієнтованого програмування;
- 3) забезпечити можливість впровадження в LMS Moodle віртуального навчального середовища ПУЕТ;
- 4) забезпечити неможливість нанесення шкоди системі та порушення системи безпеки LMS Moodle віртуального навчального середовища ПУЕТ;
- 5) забезпечити стабільність та високу продуктивність під час роботи;

- б) мати зручний та зрозумілий інтерфейс користувача;
- 7) забезпечити можливість завантажуватися і виконуватися на безлічі процесорів, операційних систем і браузерів, підключених до Інтернету;
- 8) супроводжуватися відповідною програмною документацією.

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Дистанційні технології навчання та їх складові

Дистанційні технології навчання – це комплекс освітніх технологій, включаючи психолого-педагогічні та інформаційно-комунікаційні, що надають можливість реалізувати процес дистанційного навчання у навчальних закладах та наукових установах [2].

Науково-методичне забезпечення дистанційного навчання включає:

- а) методичні (теоретичні та практичні) рекомендації щодо розроблення та використання педагогічно-психологічних та інформаційно-комунікаційних технологій дистанційного навчання;
- б) критерії, засоби і системи контролю якості дистанційного навчання;
- в) змістовне, дидактичне та методичне наповнення веб-ресурсів (дистанційних курсів) навчального плану/навчальної програми підготовки.

Системотехнічне забезпечення дистанційного навчання включає:

- а) апаратні засоби (персональні комп'ютери, мережеве обладнання, джерела безперебійного живлення, сервери, обладнання для відеоконференц-зв'язку тощо), що забезпечують розроблення і використання веб-ресурсів навчального призначення, управління навчальним процесом та необхідні види навчальної взаємодії між суб'єктами дистанційного навчання у синхронному і асинхронному режимах;
- б) інформаційно-комунікаційне забезпечення із пропускну здатністю каналів, що надає всім суб'єктам дистанційного навчання навчального закладу цілодобовий доступ до веб-

ресурсів і веб-сервісів для реалізації навчального процесу у синхронному та асинхронному режимах;

- в) програмне забезпечення загального та спеціального призначення (у тому числі для осіб з особливими потребами), яке має бути ліцензійним або побудованим на програмних продуктах з відкритими кодами;
- г) веб-ресурси навчальних дисциплін (програм), що необхідні для забезпечення дистанційного навчання, можуть містити:

- 1) методичні рекомендації щодо їх використання, послідовності виконання завдань, особливостей контролю тощо;
- 2) документи планування навчального процесу (навчальні програми, навчально-тематичні плани, розклади занять);
- 3) відео- та аудіозаписи лекцій, семінарів тощо;
- 4) мультимедійні лекційні матеріали;
- 5) термінологічні словники;
- 6) практичні завдання із методичними рекомендаціями щодо їх виконання;
- 7) віртуальні лабораторні роботи із методичними рекомендаціями щодо їх виконання;
- 8) інтерактивні компоненти із методичними рекомендаціями щодо їх використання;
- 9) пакети тестових завдань для проведення контрольних заходів, тестування із автоматизованою перевіркою результатів, тестування із перевіркою викладачем;
- 10) ділові ігри із методичними рекомендаціями щодо їх використання;
- 11) електронні бібліотеки чи посилання на них;
- 12) бібліографії;

13) дистанційний курс, що об'єднує зазначені вище веб-ресурси навчальної дисципліни (програми) єдиним педагогічним сценарієм;

14) інші ресурси навчального призначення.

Перелік веб-ресурсів навчальних дисциплін (програм), необхідних для забезпечення дистанційного навчання, визначається навчальним закладом залежно від профілю навчальної дисципліни.

2.2 Огляд технологій розробки інтерактивних компонентів для Web-середовища

Зробимо огляд провідних технологій розробки інтерактивних компонентів для Web-середовища.

2.2.1 Flash-технології

Для розробки інтерактивних компонентів широко використовується мультимедійна та програмна платформа Adobe Flash в поєднанні з мовою сценаріїв ActionScript [17].

Можливості Adobe Flash дозволяють розробляти насичений мультимедіа-інтерфейс інтерактивних компонентів і віртуальних лабораторних робіт, робити його оригінальним та цікавим для користувачів.

Мова сценаріїв ActionScript дозволяє додавати в додаток складні функції інтерактивності, керування відтворенням і відображення даних [18]. Мова має власний синтаксис і зарезервовані ключові слова, вона дозволяє використовувати змінні для зберігання і зчитування інформації. ActionScript має багату бібліотеку вбудованих класів, які дозволяють створювати об'єкти для виконання багатьох корисних задач. Гнучкість та функціонал мови ActionScript дозволяють в точності змоделювати виробничі процеси, обладнання, та багато іншого.

На інтерактивних компонентах, розроблених за допомогою Adobe Flash та ActionScript користувачі мають змогу в повноті ознайомитися з принципами роботи обладнання і його будовою та спробувати самостійно налаштувати його, перевірити рівень і якість знань, отриманих при вивченні певної гуманітарної, математичної чи технічної дисципліни.

Головний недолік Flash-додатків – надмірна вимогливість до ресурсів процесора [19]. Недостатня потужність комп'ютера може впливати на продуктивність операційної системи в цілому, або призвести до викривлення результатів роботи Flash-додатка, пов'язаних з відображенням анімації або підрахунком часу. Іноді це пов'язано з недоліками програмного забезпечення, відповідального за обробку Flash-компонентів, або низькою якістю самого Flash-додатка.

Інший важливий недолік полягає в тому, що не завжди є можливість запустити Flash-додаток, або вона пов'язана з деякими труднощами (наприклад, необхідно встановити плагін або оновити його до останньої версії). Деякі користувачі (або системні адміністратори в рамках цілої мережі) відключають у налаштуваннях браузера можливість завантажувати контент, оброблюваний плагінами або, що завантажується у фреймах з метою інформаційної безпеки (у зв'язку з можливою загрозою з боку контенту, наприклад, перехоплення буфера обміну), економії системних ресурсів, або для блокування рекламних банерів [20].

Це робить технологію в цілому ненадійною також для розробників, яким ніхто не гарантує, що веб-додаток на основі Flash буде взагалі відтворено.

2.2.2 Java-технології

Використання Java-технологій є найбільш поширеним способом розробки інтерактивних компонентів для Web-середовища [9]. Переваги даного способу розробки зумовлено особливостями об'єктно-орієнтованої мови програмування Java, які розглянуто в пункті 3.3.

Навчальний інтерактивний компонент, що реалізовано з використанням Java-технологій, являє собою аплет – веб-додаток, який дозволяє виконувати Java код безпосередньо в вікні браузера [9].

Аплети використовуються для надання інтерактивних можливостей веб-додаткам, котрі не можуть бути надані HTML [10]. Так як байт-код Java платформи незалежний, то Java-аплети можуть виконуватися за допомогою плагінів браузерами багатьох платформ, включаючи Microsoft Windows, UNIX, Apple Mac OS і GNU/Linux [11].

Можливості Java дозволяють розробляти інтерактивний інтерфейс, робити його зручним для користувачів [12].

2.3 Огляд відомих інтерактивних компонентів з математичних дисциплін

З метою підвищення якості розроблюваного інтерактивного компонента зробимо огляд інтерактивних компонентів з математичних дисциплін.

- 1) Інтерактивний компонент з теми «Визначення загального розв'язку лінійного рекурентного однорідного співвідношення» дистанційного навчального курсу «Дискретна математика» Головного центру дистанційного навчання ПУЕТ, розроблений в рамках кваліфікаційної роботи студентом групи СІ-52 Нефьодовим С. І. [21].

Після запуску інтерактивного компонента користувачу доступна сторінка на якій він може обрати один із запропонованих прикладів та ознайомитися із інформацією про розробника.

Обравши один з прикладів та натиснувши кнопку «Підтвердити» користувач починає проходити тренінг.

В процесі проходження тренінгу користувач відповідає на запропоновані йому запитання шляхом введення числових значень, вибору

варіантів відповідей селектором або «прапорцем». Якщо користувач дає неправильну відповідь на запитання тренінгу, то отримує інформацію про допущення помилки та підказку.

Інтерактивний компонент реалізовано мовою програмування Java.

Графічний інтерфейс користувача інтерактивного компонента реалізовано засобами інструментарію Swing – частини бібліотеки базових класів Java (JVC). Swing надає більш функціональний набір програмних компонентів для створення графічного інтерфейсу користувача, у порівнянні з більш ранішнім інструментарієм AWT [13]. Завдяки реалізації інтерактивного компонента з використанням інструментарію Swing забезпечена універсальність графічного інтерфейсу користувача на всіх платформах.

Логічні частини інтерактивного компонента розміщені на окремих вкладках (див. рисунок 2.1).

Це забезпечує зручність під час використання інтерактивного компонента. Користувач може повертатися на попередні кроки алгоритму, що підвищує рівень засвоєння практичних навичок з теми «Визначення загального розв'язку лінійного рекурентного однорідного співвідношення».

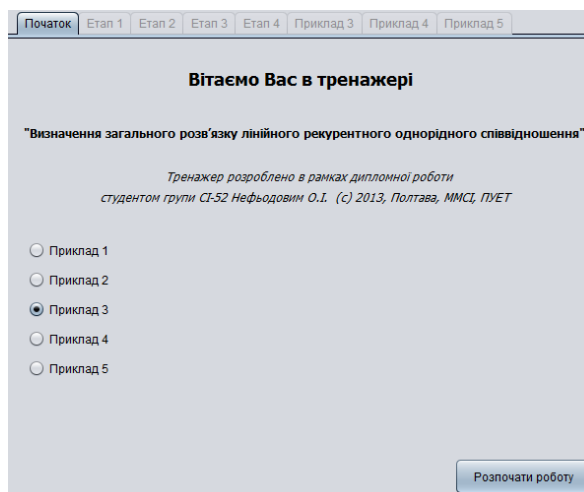


Рисунок 2.1 – Головна сторінка інтерактивного компонента

2) Інтерактивний компонент з теми «Алгоритм Крускала» Центру дистанційного навчання Сумського державного університету [22].

Інтерактивний компонент демонструє реалізацію алгоритму Крускала і фактично не потребує від користувача участі в алгоритмі роботи. Користувач лише натискає кнопку «Далі» для переходу на наступний крок алгоритму.

Інтерактивний компонент реалізовано засобами мови програмування ActionScript і мультимедійної та програмної платформи Adobe Flash.

Графічний інтерфейс користувача одновіконний, насичений анімацією та графічними об'єктами (див. рисунок 2.2).

Насичення графічного інтерфейсу інтерактивного компонента анімацією та графічними об'єктами дозволяє якнайкраще візуалізувати алгоритм Крускала. Це надає користувачу можливість краще зрозуміти та засвоїти навчальний матеріал.

На основі наведеного вище розгляду інтерактивних компонентів з математичних дисциплін, можна виокремити особливості кожного інтерактивного компонента.

Перший розглянутий інтерактивний компонент має стандартний, зручний і зрозумілий інтерфейс користувача. Під час роботи інтерактивний компонент постійно взаємодіє з користувачем через засоби елементів керування графічного інтерфейсу.

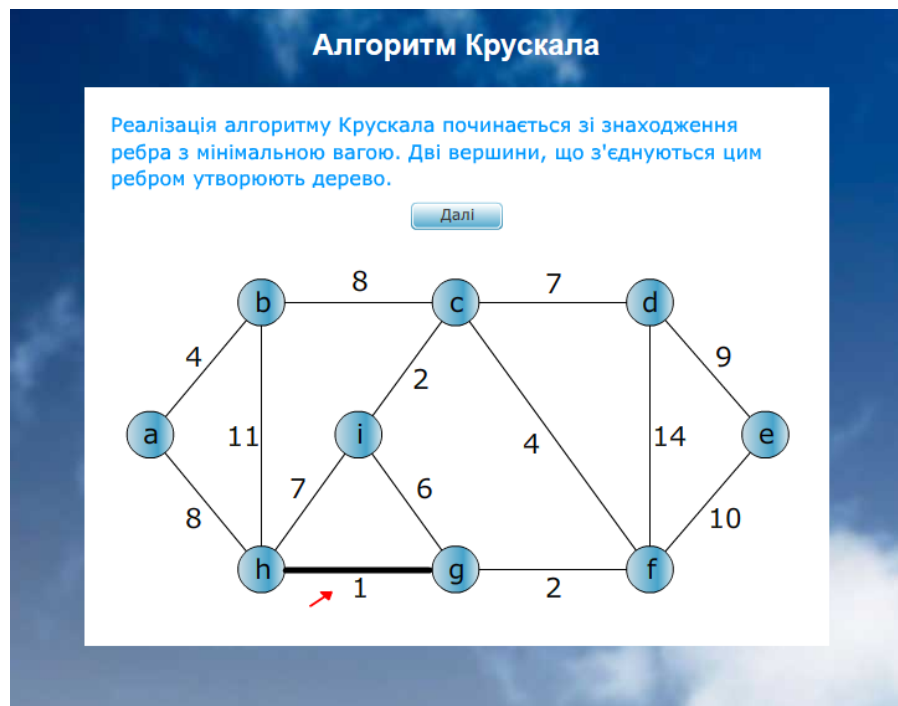


Рисунок 2.2 – Графічний інтерфейс інтерактивного компонента

Другий розглянутий інтерактивний компонент має нестандартний графічний інтерфейс користувача, насичений анімацією та графічними об'єктами. Під час роботи інтерактивний компонент взаємодіє з користувачем лише через кнопку «Далі», яка переводить інтерактивний компонент на наступний крок алгоритму.

Проаналізувавши розглянуті вище інтерактивні компоненти з математичних дисциплін, можемо зробити висновок, що як приклад програмної реалізації, доцільно використати перший розглянутий інтерактивний компонент, оскільки, відповідно до постановки задачі, графічний інтерфейс інтерактивного компонента, що розробляється не потребує насичення анімацією та графічними об'єктами, як в другому розглянутому інтерактивному компоненті, а потребує постійної взаємодії з користувачем через засоби елементів керування графічного інтерфейсу.

Як наслідок, для програмної реалізації інтерактивного компонента з теми «Модифікований симплекс-метод» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» доцільно обрати мову програмування Java. Для розробки графічного інтерфейсу інтерактивного компонента слід використати елементи керування з інструментарію бібліотеки Swing.

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Алгоритмізація інтерактивного компонента

В рамках кваліфікаційної роботи проводиться алгоритмічна розробка та програмна реалізація інтерактивного компонента дистанційного курсу «Методи оптимізації та дослідження операцій» на прикладі модифікованого симплекс-методу»»

Керівником кваліфікаційної роботи було надано текст лекції з теми «Модифікований симплекс-метод», який наведено в підрозділі 1.1 та приклад з підрозділу 3.1.

Для розгляду алгоритму роботи інтерактивного компонента, використаємо основні формули МСМ та позначення з підрозділу 1.1.

Розглянемо послідовність кроків з прикладу, який виконує користувач програмного продукту, що розробляється.

Приклад. Розв'язати ЗЛП МСМ:

$$F = 9x_1 + 10x_2 + 16x_3 \rightarrow \max; \quad (0.6)$$

$$\begin{cases} 18x_1 + 15x_2 + 12x_3 + x_4 = 360; \\ 6x_1 + 4x_2 + 8x_3 + x_5 = 192; \\ 5x_1 + 3x_2 + 3x_3 + x_6 = 180; \\ x_j \geq 0, \quad j = 1, 2, \dots, 6. \end{cases} \quad (0.7)$$

Крок 1. На екрані виводиться умова задачі та запитання: «Чи задана ЗЛП в канонічній формі?». Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Так» здійснюється перехід на 2-й крок алгоритму. Якщо обрано відповідь «Ні» – на екрані з'являється повідомлення: «Вибрана відповідь неправильна. Задана ЗЛП в канонічній формі, оскільки:

- 1) цільова функція (0.6) прямує до максимуму;
- 2) в системі обмежень (0.7) всі обмеження є рівностями;
- 3) всі змінні невід'ємні.

Крок 2. На екрані виводиться запитання: «Чи потребує задана ЗЛП знаходження базисного допустимого розв'язку, шляхом введення штучного базису (одичної матриці)?» Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Ні» здійснюється перехід на 3-й крок алгоритму. Якщо обрано відповідь «Так» – на екрані з'являється повідомлення: «Вибрана відповідь неправильна. Задана ЗЛП не потребує знаходження базисного допустимого розв'язку, є базис (P_4, P_5, P_6) ».

Крок 3. На екрані виводиться повідомлення «Утворіть базисний розв'язок x . Позначте значення базисних змінних». Користувач одну за одну заповнює порожні клітини числовими значеннями та позначає значення базисних змінних, як показано на рисунку 3.1.

При правильному заповненні активізується наступна клітина. Якщо введено неправильне значення, виводиться повідомлення про помилку: «Значення x_i введено неправильно». Якщо позначено не базисні змінні, виводиться повідомлення про помилку: «Базисна змінна вибрана неправильно.».

Якщо заповнено всі клітини та правильно позначено базисні змінні – перехід на крок 4.

$$x = \left(\begin{array}{|c|} \hline \input{checkbox} \\ \hline 0 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \input{checkbox} \\ \hline 0 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \input{checkbox} \\ \hline 0 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \input{checkbox} \\ \hline 360 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \input{checkbox} \\ \hline 192 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \input{checkbox} \\ \hline 180 \\ \hline \end{array} \right)$$

Рисунок 3.1 – Утворення базисного розв'язку x та позначення базисних змінних

Крок 4. На екрані виводиться повідомлення «Утворіть матрицю $B = (P_{i_1}, \dots, P_{i_m})$ ». Користувач одну за одну заповнює порожні клітини числовими значеннями, як показано на рисунку 3.2.

$$B = (P_4, P_5, P_6) = \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

Рисунок 3.2 – Матриця $B = (P_4, P_5, P_6)$

При правильному заповненні активізується наступна клітина. Якщо введено неправильне значення, виводиться повідомлення про помилку: «Введено значення неправильно». При заповненні всіх клітин – перехід на крок 5.

Крок 5. На екрані виводиться повідомлення «Чи є матриця

$B = (P_4, P_5, P_6) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ одиничною матрицею $B^{-1} = A_1, A_2, A_3$?». Користувач

може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Так» здійснюється перехід на 6-й крок алгоритму. Якщо обрано відповідь «Ні» – на екрані з'являється повідомлення: «Вибрана відповідь

неправильна. Матриця $B = (P_4, P_5, P_6) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ є одиничною матрицею.

Крок 6. На екрані виводиться повідомлення: «Якою буде матриця B^{-1} ? Введіть її елементи.». Користувач одну за одною заповнює порожні клітини числовими значеннями, як показано на рисунку 3.3.

$$B^{-1} = \begin{array}{ccc} \boxed{1} & \boxed{0} & \boxed{0} \\ \boxed{0} & \boxed{1} & \boxed{0} \\ \boxed{0} & \boxed{0} & \boxed{1} \end{array}$$

Рисунок 3.3 – Матриця B^{-1}

У випадку, якщо матрицю B^{-1} утворено правильно – здійснюється перехід на 7-й крок алгоритму.

У випадку, якщо матрицю B^{-1} утворено неправильно – на екран виводиться повідомлення про помилку: «Матрицю B^{-1} утворено

неправильно! Очевидно, що $B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (A_1, A_2, A_3)$.» Здійснюється

перехід на 7-й крок алгоритму.

Крок 7. На екрані виводиться повідомлення: «Дайте відповідь на запитання: «Чим відрізняється допоміжна симплекс-таблиця МСМ від звичайної симплекс-таблиці? (позначте правильні варіанти відповіді):

- 1) **Допоміжна симплекс-таблиця МСМ** має додаткові рядки (з номерами $m + 1, m + 2, \dots, m + k$), де записують значення $\Delta_1^{(2)}, \dots, \Delta_n^{(2)}; \dots; \Delta_1^{(k)}, \dots, \Delta_n^{(k)}$, що отримують в процесі розв'язування задачі.
- 2) **Звичайна симплекс-таблиця** має додаткові рядки (з номерами $m + 1, m + 2, \dots, m + k$), де записують значення $\Delta_1^{(2)}, \dots, \Delta_n^{(2)}; \dots; \Delta_1^{(k)}, \dots, \Delta_n^{(k)}$, що отримують в процесі розв'язування задачі, а **допоміжна симплекс-таблиця МСМ** – не має.
- 3) Крім додаткових рядків справа до **допоміжної симплекс-таблиці МСМ** приписують додаткові k стовпців для координат векторів $\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(k)}$, що розраховують при розв'язуванні задачі, а до **звичайної симплекс-таблиці** – ні.
- 4) Крім додаткових рядків справа до **звичайної симплекс-таблиці** приписують додаткові k стовпців для координат векторів $\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(k)}$, що розраховують при розв'язуванні задачі, а до **допоміжної симплекс-таблиці МСМ** – ні.»

У випадку, якщо користувач позначив правильні варіанти відповіді – перехід на крок 8.

У випадку, якщо користувач позначив неправильні варіанти відповіді, на екран виводиться повідомлення: «Відповідь не правильна! На відміну від звичайної симплекс-таблиці, допоміжна таблиця МСМ має додаткові рядки (з номерами $m + 1, m + 2, \dots, m + k$), де записують значення $\Delta_1^{(1)}, \dots, \Delta_n^{(1)}; \Delta_1^{(2)}, \dots, \Delta_n^{(2)}; \dots; \Delta_1^{(k)}, \dots, \Delta_n^{(k)}$, що отримують в процесі розв'язування задачі. Крім додаткових рядків справа до таблиці

приписують додаткові k стовпців для координат векторів $\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(k)}$, що розраховують при розв'язуванні задачі.»

Крок 8. На екрані виводиться повідомлення: «Дайте відповідь на запитання: «Чим відрізняється основна симплекс-таблиця МСМ від звичайної симплекс-таблиці? (позначте правильні варіанти відповіді):

- 1) **В основну симплекс-таблицю МСМ** записують стовпці A_i – стовпці з матриці B^{-1} .
- 2) **В основну симплекс-таблицю МСМ** записують всі вектори умов P_j .
- 3) В $m + 1$ рядку **основної симплекс-таблиці МСМ** записують координати вектора Ω .
- 4) В $m + 1$ рядку **основної симплекс-таблиці МСМ** записують оцінки Δ_j .
- 5) **Основна симплекс-таблиця МСМ** має додатковий стовпець P_s^* – координати вектора P_s в базисі $P_{i_1}, P_{i_2}, \dots, P_{i_m}$ – вектора, який потрібно ввести в базис на наступній ітерації
- 6) **Основна симплекс-таблиця МСМ** не має додатковий стовпець P_s^* – координати вектора P_s в базисі $P_{i_1}, P_{i_2}, \dots, P_{i_m}$ – вектора, який потрібно ввести в базис на наступній ітерації.

У випадку, якщо користувач позначив правильні варіанти відповіді – перехід на крок 9.

У випадку, якщо користувач позначив неправильні варіанти відповіді, на екран виводиться повідомлення: «Відповідь не правильна! Основна симплекс-таблиця МСМ відрізняється від звичайної симплекс-таблиці тим що:

- 1) в таблицю записують стовпці A_i – стовпці з матриці B^{-1} (а не вектори умов P_j);
- 2) в $m + 1$ рядку записують координати вектора Ω (а не оцінки Δ_j);

- 3) має додатковий стовпець P_s^* – координати вектора P_s в базисі $P_{i_1}, P_{i_2}, \dots, P_{i_m}$ – вектора, який потрібно ввести в базис на наступній ітерації.»

Крок 9. На екрані виводиться умова задачі та повідомлення: «Складіть основну симплекс-таблицю МСМ. Запишіть координати вектора $\Omega^{(1)} = (\omega_1^{(1)}, \omega_2^{(1)}, \omega_3^{(1)})$. Обчисліть $F_0 = (c_s, P_0)$ та запишіть отримані числові значення у відповідні комірки таблиці». Користувач одну за одною заповнює порожні клітини основної таблиці МСМ числовими значеннями (див. рисунок 3.4).

У разі, якщо значення координат вектора $\Omega^{(1)} = (\omega_1^{(1)}, \omega_2^{(1)}, \omega_3^{(1)})$ введено неправильно, на екрані виводиться повідомлення про помилку: «Координати вектора $\Omega^{(1)} = (\omega_1^{(1)}, \omega_2^{(1)}, \omega_3^{(1)})$ обчислено неправильно! Оскільки $\Omega^{(1)} = C_s B^{-1}$, то:

$$\omega_1^{(1)} = (c_s, A_1) = 0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 = 0;$$

$$\omega_2^{(1)} = (c_s, A_2) = 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 = 0;$$

$$\omega_3^{(1)} = (c_s, A_3) = 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 = 0.»$$

У разі, якщо значення $F_0 = (c_s, P_0)$ введено неправильно, на екран виводиться повідомлення про помилку: «Значення $F_0 = (c_s, P_0)$ обчислено неправильно! Обчислення необхідно проводити так: $F_0 = (c_s, P_0) = 0 \cdot 360 + 0 \cdot 192 + 0 \cdot 180 = 0$.»

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 10.

$\overbrace{\hspace{10em}}^{B^{-1}}$

i	Базис	C_s	P_0	A_1	A_2	A_3
1	P_4	0	360	1	0	0
2	P_5	0	192	0	1	0
3	P_6	0	180	0	0	1
4			0	0	0	0

$$F_0 \quad \underbrace{\hspace{10em}}_{\Omega^{(1)}=C_\delta B^{-1}}$$

Рисунок 3.4– Основна таблиця МСМ

Крок 10. На екрані виводиться повідомлення «Складіть допоміжну таблицю МСМ. Обчисліть $\Delta^{(1)}$ та запишіть отримані числові значення у відповідні комірки допоміжної симплекс–таблиці МСМ.». Користувач одну за одною заповнює порожні комірки допоміжної таблиці МСМ числовими значеннями (див. рисунок 3.5).

i	Базис	C_δ	P_0	9	10	16	0	0	0	$\Omega^{(1)}$...
				P_1	P_2	P_3	P_4	P_5	P_6		
1	P_4	0	360	18	15	12	1	0	0	0	...
2	P_5	0	192	6	4	8	0	1	0	0	...
3	P_6	0	180	5	3	3	0	0	1	0	...
4			$\Delta^{(1)}$	-9	-10	-16	0	0	0		
...				

Рисунок 3.5 – Допоміжна таблиця МСМ на 10-му кроці алгоритму інтерактивного компонента

У разі, якщо значення c_j введено неправильно, на екран виводиться повідомлення про помилку: «Значення c_j введено неправильно! Значення c_j – це коефіцієнт цільової функції біля змінної x_j . Отже, $c_1 = 9; c_2 = 10; c_3 = 16; c_4 = c_5 = c_6 = 0.$ »

У разі, якщо значення елементів C_δ введено неправильно, на екран виводиться повідомлення про помилку: «Значення елементів C_δ введено

неправильно! $C_\delta = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.».

У разі, якщо значення елементів P_0 введено неправильно, на екран виводиться повідомлення про помилку: «Значення елементів P_0 введено не

правильно!

$$P_0 = \begin{pmatrix} 360 \\ 192 \\ 180 \end{pmatrix} . \rangle .$$

У разі, якщо значення елементів P_i введено неправильно, на екран виводиться повідомлення про помилку: «Значення елементів P_i введено неправильно!

$$P_1 = \begin{pmatrix} 18 \\ 6 \\ 5 \end{pmatrix}; P_2 = \begin{pmatrix} 15 \\ 4 \\ 3 \end{pmatrix}; P_3 = \begin{pmatrix} 12 \\ 8 \\ 3 \end{pmatrix}; P_4 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; P_5 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; P_6 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} . \rangle .$$

У разі, якщо значення $\Delta^{(1)}$ введено неправильно, на екран виводиться повідомлення: «Значення $\Delta^{(1)}$ обчислено неправильно! $\Delta_j = \Omega P_j - c_j$, отже $\Delta_1^{(1)} = (\Omega^{(1)}, P_1) - c_1 = 0 \cdot 18 + 0 \cdot 6 + 0 \cdot 5 - 9 = -9$.

$$\text{Аналогічно: } \Delta_2^{(1)} = -10; \Delta_3^{(1)} = -16; \Delta_4^{(1)} = \Delta_5^{(1)} = \Delta_6^{(1)} = 0 .$$

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 11.

Крок 11. На екрані виводиться повідомлення «Чи є отриманий розв’язок оптимальним?» Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Так» – на екрані з’являється повідомлення: «Вибрана відповідь неправильна. Серед чисел $\Delta_j^{(1)}$ є від’ємні, отже розв’язок не є оптимальним.» Якщо обрано відповідь «Ні» – здійснюється перехід на 12-й крок алгоритму.

Крок 12. На екрані виводиться повідомлення «Знайдіть s з умови $|\Delta_s| = \max_{\substack{\Delta_j < 0 \\ 1 \leq j \leq n}} |\Delta_j|$. Позначте в допоміжній таблиці напрямний стовпець s селектором.» (див. рисунок 3.6).

У випадку, якщо напрямний стовпець s вибрано правильно – здійснюється перехід на 13-й крок алгоритму.

У випадку, якщо напрямний стовпець s вибрано неправильно – на екран виводиться повідомлення про помилку: «Напрямний стовпець s обрано неправильно! Направний стовпець s знаходять з умови

$$|\Delta_s| = \max_{\substack{\Delta_j < 0 \\ 1 \leq j \leq n}} |\Delta_j|, \text{ і це } P_3 \text{ .} \gg.$$

і	Базис	C_δ	P_0	9	10	16	0	0	0	$\Omega^{(1)}$...
				P_1	P_2	P_3	P_4	P_5	P_6		
1	P_4	0	360	18	15	12	1	0	0	0	...
2	P_5	0	192	6	4	8	0	1	0	0	...
3	P_6	0	180	5	3	3	0	0	1	0	...
4			$\Delta^{(1)}$	-9	-10	-16	0	0	0		
...			...								

Рисунок 3.6 – Допоміжна таблиця МСМ

на 12-му кроці алгоритму інтерактивного компонента

Крок 13. На екрані виводиться основна симплекс-таблиця МСМ та повідомлення: «Запишіть координати вектор-стовпця P_3^* у відповідні комірки таблиці.». Користувач одну за одною заповнює порожні клітини основної симплекс-таблиці МСМ числовими значеннями, як показано на рисунку 3.7.

У разі, якщо координати вектор-стовпця P_3^* введено неправильно, на екран виводиться повідомлення: «Координати вектора P_3^* обчислено не

правильно! Значення $P_3^* = B^{-1} \cdot P_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 12 \\ 8 \\ 3 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \\ 3 \end{pmatrix} \gg.$

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 14.

$\overbrace{\hspace{10em}}^{B^{-1}}$

і	Базис	C_δ	P_0	A_1	A_2	A_3	P_3^*
1	P_4	0	360	1	0	0	12
2	P_5	0	192	0	1	0	8
3	P_6	0	180	0	0	1	3

4			0	0	0	0	-16
			\square F_0	$\underbrace{\hspace{10em}}_{\Omega^{(1)}=C_\delta B^{-1}}$			\square $\Delta_3^{(1)}$

Рисунок 3.7 – Основна симплекс-таблиця МСМ на 13-му кроці алгоритму інтерактивного компонента

Крок 14. На екрані виводиться повідомлення «Чи можна сказати, що цільова функція задачі необмежена?». Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Ні» здійснюється перехід на 15-й крок алгоритму. Якщо обрано відповідь «Так» – на екрані з'являється повідомлення: «Вибрана відповідь неправильна. Не можна сказати, що цільова функція ЗЛП необмежена, оскільки існує $\beta_{i_k,s} > 0$ в стовпці P_3^* .»

Крок 15. На екрані виводиться основна симплекс-таблиця МСМ та повідомлення: «Обчисліть значення $\frac{b_{i_r}}{\beta_{i_r,s}}$ та запишіть їх у відповідні комірки таблиці.». Користувач одну за одною заповнює порожні клітини основної симплекс-таблиці МСМ числовими значеннями, як показано на рисунку 3.8.

У разі, якщо значення $\frac{b_{i_r}}{\beta_{i_r,s}}$ введено неправильно, на екран виводиться

повідомлення: «Значення $\frac{b_{i_r}}{\beta_{i_r,s}}$ обчислено неправильно!

$$\frac{b_{i_r}}{\beta_{i_r,s}} = \begin{pmatrix} 360 \\ 192 \\ 180 \end{pmatrix} : \begin{pmatrix} 12 \\ 8 \\ 3 \end{pmatrix} = \begin{pmatrix} 30 \\ 24 \\ 60 \end{pmatrix} .».$$

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 16.

				$\underbrace{\hspace{10em}}_{B^{-1}}$				
i	Базис	C _δ	P ₀	A ₁	A ₂	A ₃	P ₃ [*]	$\frac{b_{i_r}}{\beta_{i_r,s}}$

1	P_4	0	360	1	0	0	12	30
2	P_5	0	192	0	1	0	8	24
3	P_6	0	180	0	0	1	3	60
4			0	0	0	0	-16	

\square F_0 $\underbrace{\hspace{10em}}_{\Omega^{(1)}=C_\delta B^{-1}}$ $\square \uparrow \Delta_3^{(1)}$

Рисунок 3.8 – Основна симплекс-таблиця МСМ на 15-му кроці алгоритму інтерактивного компонента

Крок 16. На екрані виводиться основна симплекс-таблиця МСМ та

повідомлення: «Знайдіть $\theta = \min_{\substack{\beta_{i,s} > 0 \\ 1 \leq k \leq m}} \frac{b_{i,k}}{\beta_{i,s}} = \frac{b_{i,r}}{\beta_{i,s}}$. Позначте напрямний рядок r та розв'язальний елемент $\beta_{i,s}$ селектором. Перерахуйте основну симплекс-таблицю МСМ та заповніть її числовими значеннями.». Користувач вибирає напрямний рядок r та розв'язальний елемент $\beta_{i,s}$ селектором та одну за одну заповнює порожні комірки основної таблиці МСМ числовими значеннями (див. рисунок 3.9).

і	Базис	C_δ	P_0	B^{-1}			P^*_3	$\frac{b_{i,r}}{\beta_{i,s}}$
				A_1	A_2	A_3		
1	P_4	0	360	1	0	0	12 <input type="radio"/>	30 <input type="radio"/>
2	P_5	0	192/24	0/0	1/1/8	0/0	8 <input checked="" type="radio"/>	24 <input checked="" type="radio"/>
3	P_6	0	180	0	0	1	3 <input type="radio"/>	60 <input type="radio"/>
4			0	0	0	0	-16	

\square F_0 $\underbrace{\hspace{10em}}_{\Omega^{(1)}=C_\delta B^{-1}}$ $\square \uparrow \Delta_3^{(1)}$

Рисунок 3.9 – Основна таблиця МСМ на 16-му кроці алгоритму інтерактивного компонента

У випадку, якщо напрямний рядок r вибрано неправильно – на екран виводиться повідомлення про помилку: «Напрямний рядок r обрано неправильно!».

У випадку, якщо розв'язальний елемент $\beta_{i,s}$ вибрано неправильно – на екран виводиться повідомлення про помилку: «Розв'язальний елемент $\beta_{i,s}$ обрано неправильно! Розв'язальний елемент $\beta_{i,s}$ – напрямний рядок r ; напрямний стовпець s ».

У випадку, якщо елементи векторів P_0 або A_i в рядку таблиці записано неправильно – на екран виводиться повідомлення про помилку: «Помилка! Елементи векторів P_0, A_1, \dots, A_m в рядку нової симплекс-таблиці, в якій записаний вектор, що вводиться до базису, одержують з елементів цього ж рядка вихідної таблиці, діленням їх на величину розв'язувального елементу.».

Перехід на крок 17.

Крок 17. На екрані виводиться перша основна симплекс-таблиця та повідомлення: «Складіть другу основну симплекс-таблицю МСМ, шляхом перерахунку першої.».

Користувач одну за одною заповнює порожні клітини таблиці, як показано на рисунку 3.10.

i	Базис	C_δ	P_0	A_1	A_2	A_3
1	P_4	0	72	1	-3/2	0
2	P_3	16	24	0	1/8	0
3	P_6	0	108	0	-3/8	1

Рисунок 3.10 – Друга основна симплекс-таблиця МСМ на 17-му кроці алгоритму інтерактивного компонента

У випадку, якщо значення C_δ записано неправильно – на екран виводиться повідомлення про помилку: «Значення в стовпці C_δ , в рядку вектора, що вводиться, записано неправильно! В стовпці C_δ в рядку

вектора, що вводиться, проставляють величину C_k , де k – індекс вектора, що вводиться до базису. Отже, $C_{k=3} = 16$.»

У випадку, якщо значення P_0 записано неправильно – на екран виводиться повідомлення про помилку: « P_0 записано неправильно! Значення P_0 в рядку нової симплекс-таблиці, в якій записаний вектор, що вводиться до базису, одержують з елемента цього ж рядка вихідної таблиці, діленням його на величину розв'язувального елемента.

Елементи стовпців вектора P_0 , A_i нової симплекс-таблиці обчислюються за **правилом прямокутника**.

Для обчислення будь-якого з цих елементів за допомогою розв'язального елемента знаходять три числа:

- 1) число, що стоїть в вихідній симплекс-таблиці на місці шуканого елемента нової симплекс-таблиці;
- 2) число, що стоїть в вихідній симплекс-таблиці на перетині рядка, в якому знаходиться шуканий елемент нової симплекс-таблиці, і стовпця, відповідного вектору, що вводиться в базис;
- 3) число, що стоїть в новій симплекс-таблиці на перетині стовпця, в якому стоїть шуканий елемент, і рядка вектора, що вводиться в базис (як відзначено вище, цей рядок одержано з рядка вихідної симплекс-таблиці діленням її елементів на розв'язувальний елемент).

Ці три числа разом з розв'язальним елементом утворюють прямокутник, три вершини (№1, №2, р.е.) якого відповідають числам, що знаходяться в вихідній симплекс-таблиці, а четверте (№3') – числу, з нової симплекс-таблиці. Для визначення шуканого елемента нової симплекс-таблиці від першого числа (№1) віднімають добуток другого (№2) і третього (№3') (рисунок 1, 2):

$$\boxed{\text{№ } I'} = \boxed{\text{№ } I} - \boxed{\text{№ } 2} \cdot \boxed{\text{№ } 3'}$$

Рисунок 1 – Схема обчислення нового елемента симплекс-таблиці

№1		№2
№3/№3'		<i>p.e.</i>

Рисунок 2 – Схема для обчислень. В клітині *p.e.* стоїть розв'язальний елемент

Перехід на крок 18.

Крок 18. На екрані виводиться повідомлення: «Запишіть координати вектора $\Omega^{(2)} = (\omega_1^{(2)}, \omega_2^{(2)}, \omega_3^{(2)})$. Обчисліть $F_0 = (c_\delta, P_0)$ та запишіть у відповідні комірки таблиці». Користувач одну за одну заповнює порожні клітини другої основної таблиці МСМ числовими значеннями, як показано на рисунку 3.11.

i	Базис	C_δ	P_0	A_1	A_2	A_3
1	P_4	0	72	1	-3/2	0
2	P_3	16	24	0	1/8	0
3	P_6	0	108	0	-3/8	1
4			384	0	2	0

\square
 F_0 $\underbrace{\hspace{10em}}_{\Omega^{(2)}}$

Рисунок 3.11 – Друга основна таблиця МСМ на 18-му кроці алгоритму інтерактивного компонента

У разі, якщо значення координат вектора $\Omega^{(2)} = (\omega_1^{(2)}, \omega_2^{(2)}, \omega_3^{(2)})$ введено неправильно. На екран виводиться повідомлення: «Координати вектора

$\Omega^{(2)} = (\omega_1^{(2)}, \omega_2^{(2)}, \omega_3^{(2)})$ обчислено неправильно! $\Omega^{(2)} = C_\delta B^{-1}$, отже

$$\omega_1^{(2)} = (c_\delta, A_1) = 0 \cdot 1 + 16 \cdot 0 + 0 \cdot 0 = 0;$$

$$\omega_2^{(2)} = (c_\delta, A_2) = 0 \cdot (-3/2) + 16 \cdot 1/8 + 0 \cdot (-3/2) = 2;$$

$$\omega_3^{(2)} = (c_\delta, A_3) = 0 \cdot 0 + 16 \cdot 0 + 0 \cdot 1 = 0.$$

У разі, якщо значення $F_0 = (c_\delta, P_0)$ введено неправильно, на екран виводиться повідомлення про помилку: «Значення $F_0 = (c_\delta, P_0)$ обчислено неправильно! Обчислення необхідно проводити так: $F_0 = (c_\delta, P_0) = 0 \cdot 72 + 16 \cdot 24 + 0 \cdot 108 = 384$.»

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 19.

Крок 19. На екрані виводиться допоміжна симплекс-таблиця МСМ та повідомлення «Обчисліть $\Delta^{(2)}$ та запишіть отримані числові значення у відповідні комірки допоміжної симплекс-таблиці МСМ.». Користувач одну за одною заповнює відповідні порожні комірки допоміжної таблиці МСМ числовими значеннями (див. рисунок 3.12).

i	Базис	C_δ	P_0	9	10	16	0	0	0	$\Omega^{(1)}$	$\Omega^{(2)}$...
				P_1	P_2	P_3	P_4	P_5	P_6			
1	P_4	0	360	18	15	12	1	0	0	0	0	...
2	P_5	0	192	6	4	8	0	1	0	0	2	...
3	P_6	0	180	5	3	3	0	0	1	0	0	...
4			$\Delta^{(1)}$	-9	-10	-16	0	0	0			
5			$\Delta^{(2)}$	3	-2	0	0	2	0			
...					

Рисунок 3.12 – Допоміжна таблиця МСМ на 19-му кроці алгоритму інтерактивного компонента

У разі, якщо значення $\Delta^{(2)}$ введено неправильно, на екран виводиться повідомлення: «Значення $\Delta^{(2)}$ обчислено неправильно! $\Delta_j = \Omega P_j - c_j$, отже $\Delta_1^{(2)} = (\Omega^{(2)}, P_1) - c_1 = 0 \cdot 18 + 2 \cdot 6 + 0 \cdot 5 - 9 = 3$.

Аналогічно: $\Delta_2^{(2)} = -2; \Delta_3^{(2)} = 0; \Delta_4^{(2)} = 0; \Delta_5^{(2)} = 2; \Delta_6^{(2)} = 0$.

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 20.

Крок 20. На екрані виводиться повідомлення «Чи є отриманий розв’язок оптимальним?» Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Так» – на екрані з’являється повідомлення: «Вибрана відповідь неправильна. Серед чисел $\Delta_j^{(2)}$ є від’ємні, отже розв’язок не є оптимальним.» Якщо обрано відповідь «Ні» – здійснюється перехід на 21-й крок алгоритму.

Крок 21. На екрані виводиться повідомлення «Знайдіть s з умови $|\Delta_s| = \max_{\substack{\Delta_j < 0 \\ 1 \leq j \leq n}} |\Delta_j|$. Позначте в допоміжній таблиці напрямний стовпець s селектором.» (див. рисунок 3.13).

У випадку, якщо напрямний стовпець s вибрано правильно – здійснюється перехід на 22-й крок алгоритму.

У випадку, якщо напрямний стовпець s вибрано неправильно – на екран виводиться повідомлення про помилку: «Напрямний стовпець s обрано неправильно! Напрямний стовпець s знаходять з умови $|\Delta_s| = \max_{\substack{\Delta_j < 0 \\ 1 \leq j \leq n}} |\Delta_j|$, тобто P_2 ».

i	Базис	C_δ	P_0	9	10	16	0	0	0	$\Omega^{(1)}$	$\Omega^{(2)}$...
				P_1	P_2	P_3	P_4	P_5	P_6			
1	P_4	0	360	18	15	12	1	0	0	0	0	...
2	P_5	0	192	6	4	8	0	1	0	0	2	...
3	P_6	0	180	5	3	3	0	0	1	0	0	...
4			$\Delta^{(1)}$	-9	-10	-16	0	0	0			
5			$\Delta^{(2)}$	3 <input type="radio"/>	-2 <input checked="" type="radio"/>	0 <input type="radio"/>	0 <input type="radio"/>	2 <input type="radio"/>	0 <input type="radio"/>			
...					

Рисунок 3.13 – Допоміжна таблиця МСМ на 19-му кроці алгоритму інтерактивного компонента

Крок 22. На екрані виводиться допоміжна і друга основна симплекс-таблиці МСМ та повідомлення: «Запишіть координати вектор-

стовпця P_2^* у відповідні комірки таблиці.». Користувач одну за одною заповнює порожні клітини основної симплекс-таблиці МСМ числовими значеннями, як показано на рисунку 3.14.

i	Базис	C_δ	P_0	A_1	A_2	A_3	P_2^*
1	P_4	0	72	1	-3/2	0	9
2	P_3	16	24	0	1/8	0	1/2
3	P_6	0	108	0	-3/8	1	3/2
4			384	0	2	0	-2

\square F_0 $\underbrace{\hspace{10em}}_{\Omega^{(2)}}$ $\square \uparrow \Delta_2^{(2)}$

Рисунок 3.14 – Основна симплекс-таблиця МСМ на 22-му кроці алгоритму інтерактивного компонента

У разі, якщо координати вектор-стовпця P_2^* введено неправильно, на екран виводиться повідомлення: «Координати вектора P_2^* обчислено не правильно! Значення

$$P_2^* = B^{-1} \cdot P_2 = (A_1, A_2, A_3) \cdot P_2 = \begin{pmatrix} 1 & -3/2 & 0 \\ 0 & 1/8 & 0 \\ 0 & -3/8 & 1 \end{pmatrix} \cdot \begin{pmatrix} 15 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \cdot 15 + (-3/2) \cdot 4 + 0 \cdot 3 \\ 0 \cdot 15 + 1/8 \cdot 4 + 0 \cdot 3 \\ 0 \cdot 15 + (-3/8) \cdot 4 + 1 \cdot 3 \end{pmatrix} = \begin{pmatrix} 9 \\ 1/2 \\ 3/2 \end{pmatrix}$$

».

Після заповнення всіх комірок правильними значеннями – перехід на крок 23.

Крок 23. На екрані виводиться повідомлення «Чи можна сказати, що цільова функція задачі необмежена?». Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Ні» здійснюється перехід на 24-й крок алгоритму. Якщо обрано відповідь «Так» – на екрані з'являється повідомлення: «Вибрана відповідь неправильна. Не можна сказати, що цільова функція ЗЛП необмежена, оскільки існує $\beta_{i,s} > 0$ в стовпці P_2^* .»

Крок 24. На екрані виводиться друга основна симплекс-таблиці МСМ та повідомлення: «Обчисліть значення $\frac{b_r}{\beta_{i,s}}$ та запишіть у відповідні комірки таблиці.». Користувач одну за одною заповнює порожні клітини основної симплекс-таблиці МСМ числовими значеннями, як показано на рисунку 3.15.

i	Базис	C_δ	P_0	A_1	A_2	A_3	P_2^*	$\frac{b_r}{\beta_{i,s}}$
1	P_4	0	72	1	-3/2	0	9	8
2	P_3	16	24	0	1/8	0	1/2	48
3	P_6	0	108	0	-3/8	1	3/2	72
4			384	0	2	0	-2	

\square F_0 $\underbrace{\hspace{10em}}_{\Omega^{(2)}}$ $\square \uparrow \Delta_2^{(2)}$

Рисунок 3.15 – Основна симплекс-таблиця МСМ на 24-му кроці алгоритму інтерактивного компонента

У разі, якщо значення $\frac{b_r}{\beta_{i,s}}$ введено неправильно, на екран виводиться

повідомлення: «Значення $\frac{b_r}{\beta_{i,s}}$ обчислено неправильно!

$$\frac{b_r}{\beta_{i,s}} = \begin{pmatrix} 72 \\ 24 \\ 108 \end{pmatrix} : \begin{pmatrix} 9 \\ 1/2 \\ 3/2 \end{pmatrix} = \begin{pmatrix} 8 \\ 48 \\ 72 \end{pmatrix}. \text{ »}.$$

Після заповнення правильними значення – перехід на крок 25.

Крок 25. На екрані виводиться повідомлення «Знайдіть $\theta = \min_{\substack{\beta_{i,s} > 0 \\ 1 \leq k \leq m}} \frac{b_{i_k}}{\beta_{i_k,s}} = \frac{b_r}{\beta_{i,s}}$. Позначте напрямний рядок r та розв'язальний елемент $\beta_{i,s}$ селектором. Перерахуйте основну таблицю МСМ.». Користувач вибирає напрямний рядок r та розв'язальний елемент $\beta_{i,s}$ селектором та одну за одною заповнює порожні комірки основної таблиці МСМ числовими значеннями (див. рисунок 3.16).

заповнює порожні клітини основної таблиці МСМ числовими значеннями (див. рисунок 3.17).

i	Базис	C_δ	P_0	A_1	A_2	A_3
1	P_2	10	8	1/9	-1/6	0
2	P_3	16	20	-1/18	5/24	0
3	P_6	0	96	-3/2	-1/8	1
			400	2/9	5/3	0

\square
 F_0 $\underbrace{\hspace{10em}}_{\Omega^{(3)}}$

Рисунок 3.17 – Третя основна симплекс-таблиця МСМ на 26-му кроці алгоритму інтерактивного компонента

У разі, якщо значення координат вектора $\Omega^{(3)} = (\omega_1^{(3)}, \omega_2^{(3)}, \omega_3^{(3)})$ введено неправильно, на екран виводиться повідомлення: «Координати вектора $\Omega^{(3)} = (\omega_1^{(3)}, \omega_2^{(3)}, \omega_3^{(3)})$ обчислено неправильно! $\Omega^{(3)} = C_\delta B^{-1}$, отже

$$\omega_1^{(3)} = (c_\delta, A_1) = 10 \cdot 1/9 + 16 \cdot (-1/18) + 0 \cdot (-3/2) = 2/9;$$

$$\omega_2^{(3)} = (c_\delta, A_2) = 10 \cdot (-1/6) + 16 \cdot 5/24 + 0 \cdot (-1/8) = 5/3; \text{»}.$$

$$\omega_3^{(3)} = (c_\delta, A_3) = 10 \cdot 0 + 16 \cdot 0 + 0 \cdot 1 = 0.$$

У разі, якщо значення $F_0 = (c_\delta, P_0)$ введено неправильно, на екран виводиться повідомлення про помилку: «Значення $F_0 = (c_\delta, P_0)$ обчислено неправильно! Обчислення необхідно проводити так: $F_0 = (c_\delta, P_0) = 10 \cdot 8 + 16 \cdot 20 + 0 \cdot 96 = 400$.».

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 27.

Крок 27. На екрані виводиться повідомлення «Обчисліть $\Delta^{(3)}$ та запишіть отримані числові значення у відповідні комірки допоміжної симплекс-таблиці МСМ.» Користувач одну за одну заповнює відповідні порожні комірки допоміжної таблиці МСМ числовими значеннями (див. рисунок 3.18).

i	Базис	C_δ	P_0	9	10	16	0	0	0	$\Omega^{(1)}$	$\Omega^{(2)}$	$\Omega^{(3)}$...
				P_1	P_2	P_3	P_4	P_5	P_6				
1	P_4	0	360	18	15	12	1	0	0	0	0	2/9	...

2	P_5	0	192	6	4	8	0	1	0	0	2	5/3	...
3	P_6	0	180	5	3	3	0	0	1	0	0	0	...
4			$\Delta^{(1)}$	-9	-10	-16 ↑	0	0	0				
5			$\Delta^{(2)}$	3	-2 ↑	0	0	2	0				
6			$\Delta^{(3)}$	5	0	0	2/9	5/3	0				
...						

Рисунок 3.18 – Допоміжна таблиця МСМ
на 27-му кроці алгоритму інтерактивного компонента

У разі, якщо значення $\Delta^{(3)}$ введено неправильно, на екран виводиться повідомлення: «Значення $\Delta^{(3)}$ обчислено неправильно! $\Delta_j = \Omega P_j - c_j$,

отже $\Delta_1^{(3)} = (\Omega^{(3)}, P_1) - c_1 = 2/9 \cdot 18 + 5/3 \cdot 6 + 0 \cdot 5 - 9 = 5$.

Аналогічно: $\Delta_2^{(3)} = 0; \Delta_3^{(3)} = 0; \Delta_4^{(3)} = 2/9; \Delta_5^{(3)} = 5/3; \Delta_6^{(3)} = 0$.

Після заповнення всіх комірок таблиці правильними значеннями – перехід на крок 28.

Крок 28. На екрані виводиться повідомлення «Чи є отриманий розв’язок оптимальним?» Користувач може обрати два варіанта відповіді «Так», «Ні». У випадку вибору варіанта «Так» здійснюється перехід на 29-й крок алгоритму. Якщо обрано відповідь «Ні» – на екрані з’являється повідомлення: «Вибрана відповідь неправильна! Всі $\Delta_j^{(3)} \geq 0$, отже отримано оптимальний розв’язок.». Перехід на 29-й крок алгоритму.

Крок 29. На екрані виводиться повідомлення «Отже, $\Delta_j^{(3)} \geq 0$, таким чином: $x^* = (0; 8; 20; 0; 0; 96)$ дає максимальне $F_{\max} = 400$ значення цільовій функції. Задача розв’язана.».

3.2 Блок-схема алгоритму роботи інтерактивного компонента

Блок-схему алгоритму роботи інтерактивного компонента (див. рисунок 3.19 – 3.24) оформлено згідно стандарту [8].

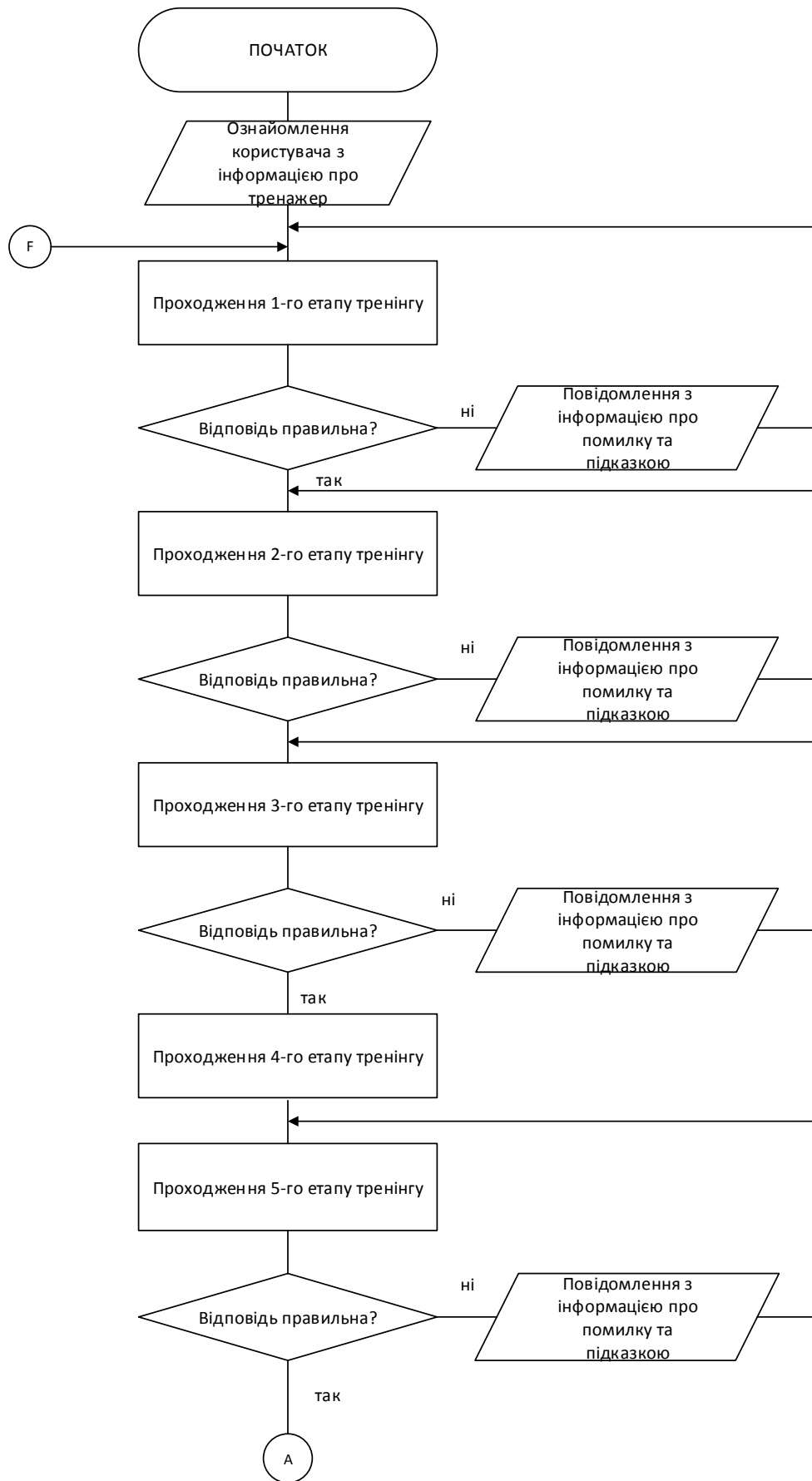


Рисунок 3.19 – Блок-схема алгоритму роботи інтерактивного компонента

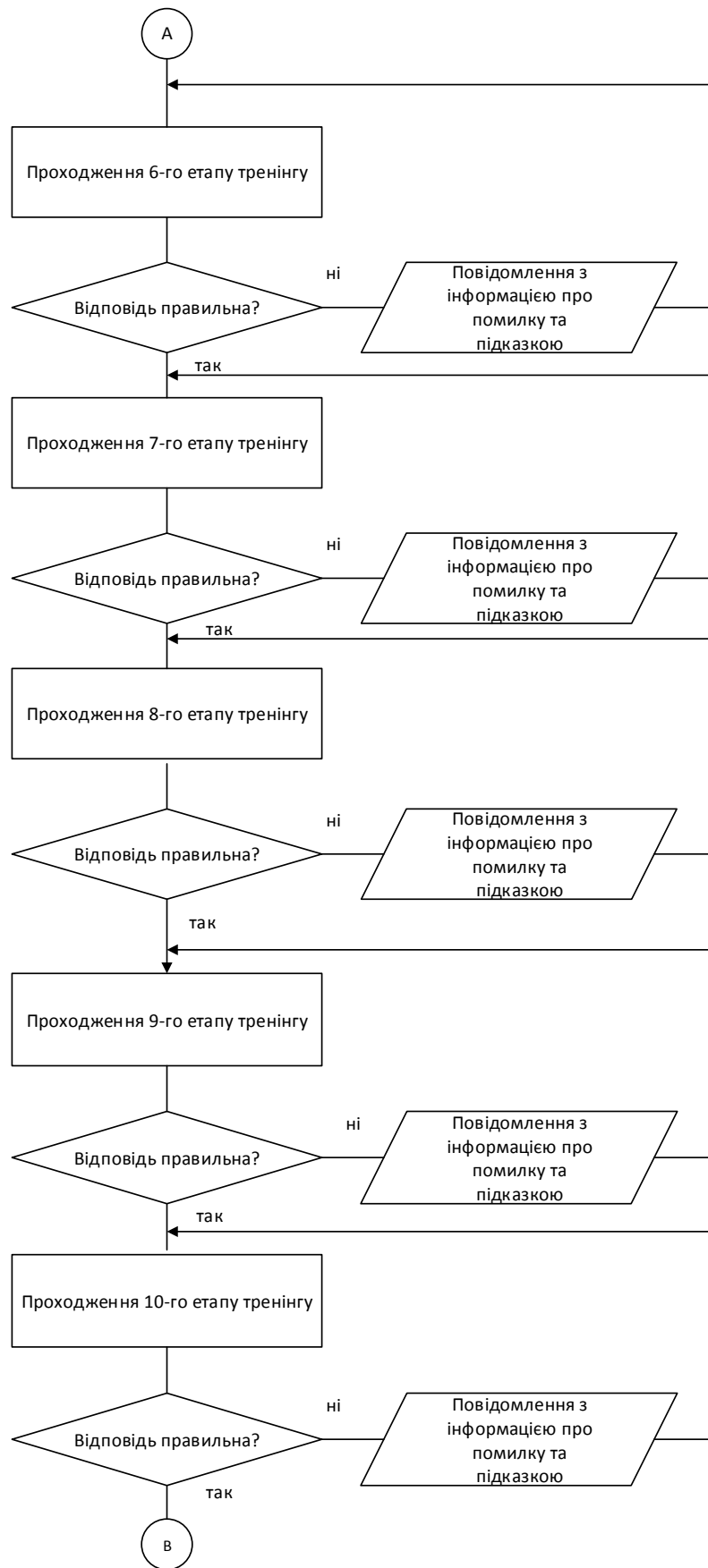


Рисунок 3.20 – Блок-схема алгоритму роботи інтерактивного компонента (продовження)

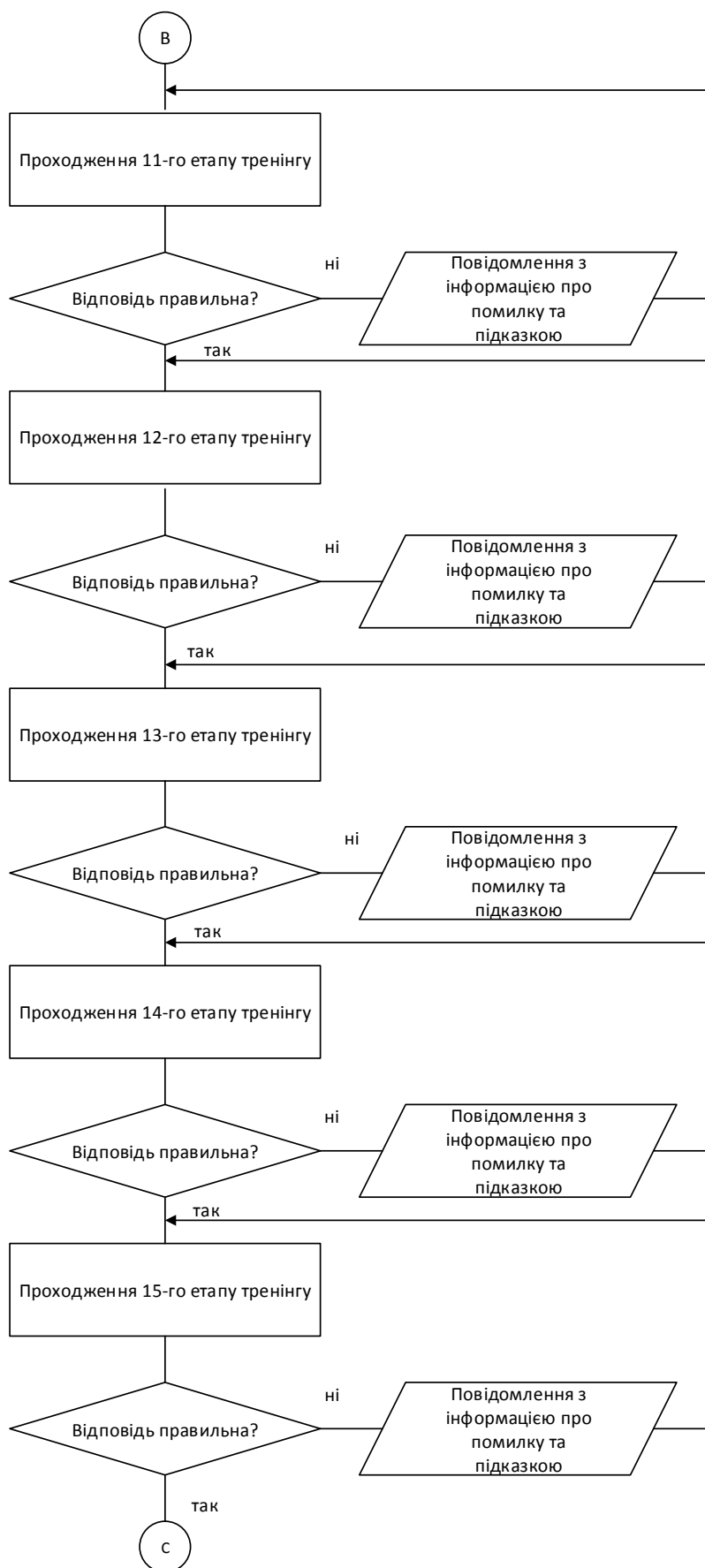


Рисунок 3.21 – Блок-схема алгоритму роботи інтерактивного компонента (продовження)

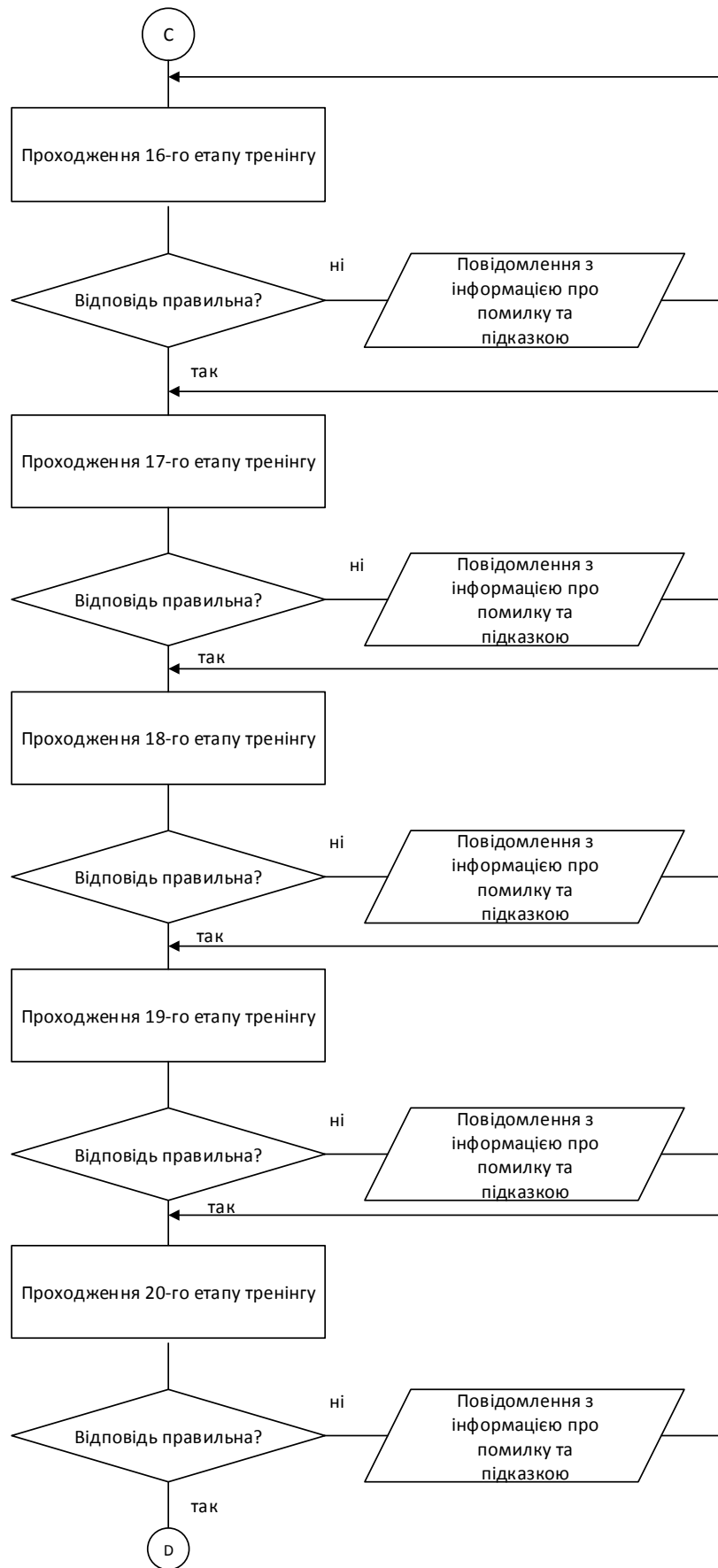


Рисунок 3.22 – Блок-схема алгоритму роботи інтерактивного компонента (продовження)

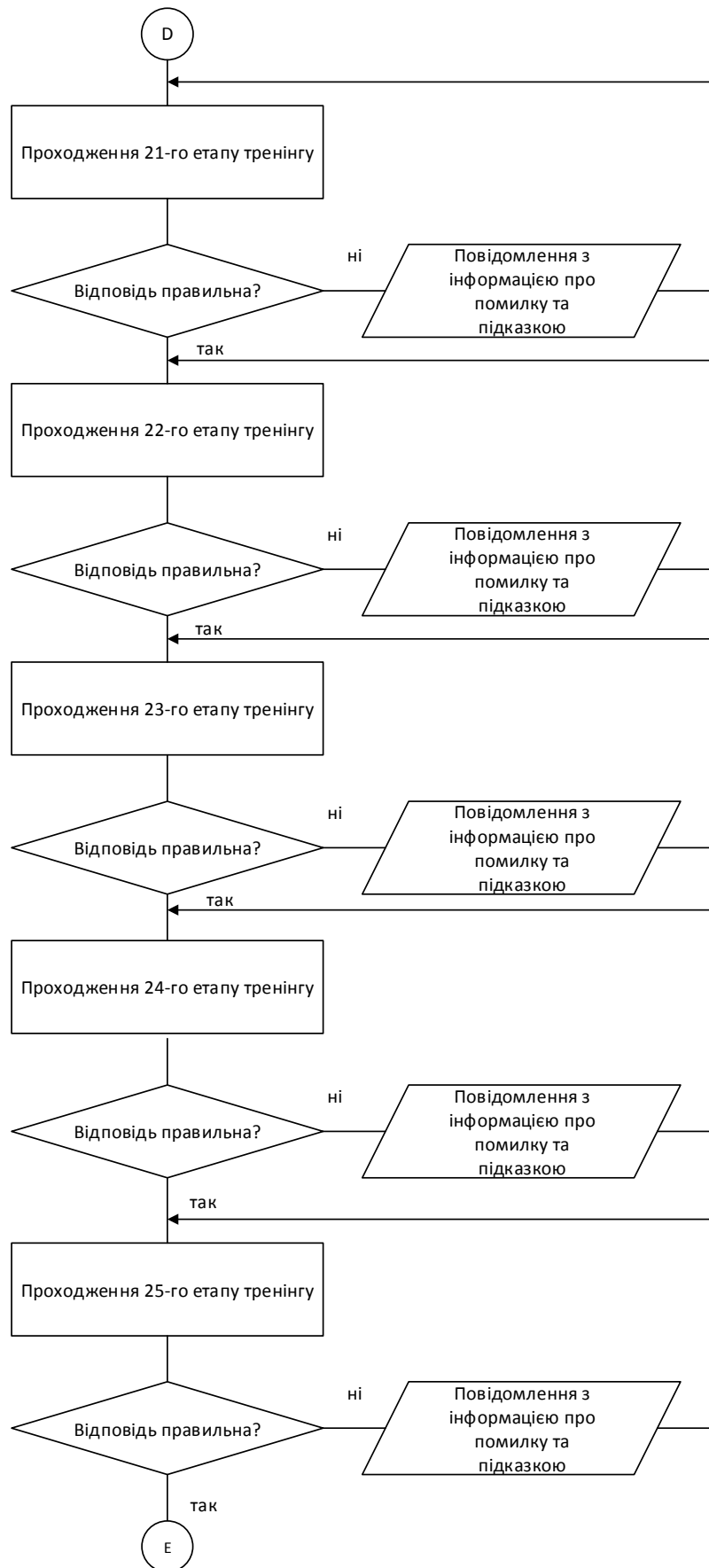


Рисунок 3.23 – Блок-схема алгоритму роботи інтерактивного компонента (продовження)

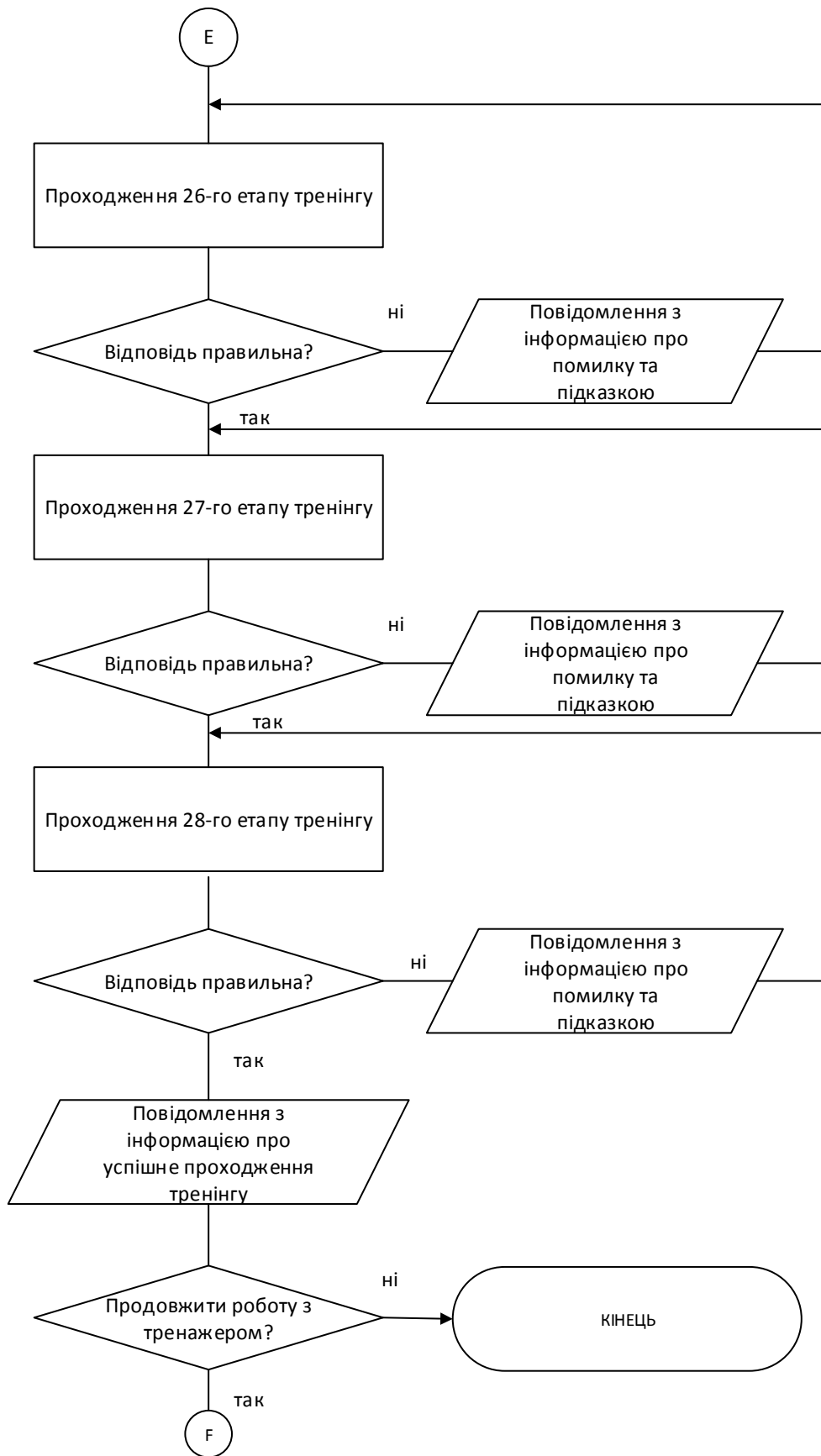


Рисунок 3.24 – Блок-схема алгоритму роботи інтерактивного компонента (продовження)

3.3 Обґрунтування вибору програмного забезпечення для реалізації інтерактивного компонента

Для реалізації інтерактивного компонента з теми «Модифікований симплекс-метод» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» було обрано мову програмування Java [9-13].

Вибір на користь Java було зроблено, оскільки дана мова програмування має ряд переваг. Нижче наведено основні з них.

Простота. Java є простою у вивченні і ефективною у використанні мовою програмування. Для тих, хто володіє певним досвідом програмування, опанування мови Java не складе особливих труднощів.

Безпека. Java забезпечує безпеку, вкладаючи аплет в середовище виконання Java і не надаючи йому доступ до інших частин операційної системи комп'ютера. Можливість завантаження аплетів зі збереженням при цьому впевненості у неможливості нанесення шкоди системі та порушення системи безпеки багато експертів і користувачі вважають найбільш новаторським аспектом Java.

Переносимість. Це основна особливість Інтернет, оскільки ця глобальна мережа з'єднує безліч різних типів комп'ютерів і операційних систем. Щоб програма Java могла виконуватися буквально на будь-якому комп'ютері, підключеному до Інтернету, був потрібен метод забезпечення виконання цієї програми в різноманітних системах. Наприклад, стосовно до аплету це означає, що один і той же аплет повинен мати можливість завантажуватися і виконуватися на безлічі процесорів, операційних систем і браузерів, підключених до Інтернету. Створення різних версій аплетів для різних комп'ютерів зовсім не раціонально. Один і той же код повинен працювати на всіх комп'ютерах.

Об'єктна орієнтованість. Об'єктна модель Java проста і легко розширювана. В той же час елементарні типи, такі як цілі числа, зберігаються у вигляді високопродуктивних компонентів, що не є об'єктами. Java має чіткий, практичний, прагматичний підхід до об'єктів.

Стійкість. Багатоплатформенне середовище веб пред'являє до програм підвищені вимоги, оскільки вони повинні надійно працювати в різноманітних системах. Тому здатність створювати стійкі програми була одним із головних пріоритетів при проектуванні Java. Для забезпечення надійності Java накладає ряд обмежень в декількох найбільш важливих областях, що змушує програміста виявляти помилки на ранніх етапах розробки програми. У той же час Java позбавляє від занепокоєння з приводу багатьох найбільш поширених помилок програмування. Оскільки Java суворо типізована мова, перевірка коду виконується під час компіляції. Однак перевірка коду здійснюється і під час виконання. У результаті більшість програмних помилок, що важко виявити, та які часто ведуть до виникнення важко відтворюваних ситуацій під час виконання, у програмі Java просто неможливі. Передбачуваність коду в різних ситуаціях – одна з основних особливостей Java.

Багатопотоковість. Мова Java підтримує написання багатопотокових програм, котрі можуть одночасно виконувати багато дій. Система часу виконання Java містить витончене, але разом з тим складне вирішення задачі синхронізації множини процесів, яке дозволяє створювати діючі без перебоїв інтерактивні системи. Простий в застосуванні підхід до організації багатопотокової обробки, реалізований в Java, дозволяє програмісту зосередити свою увагу на конкретній поведінці програми, а не на створенні багатозадачної підсистеми.

Архітектурна нейтральність. Виконуючи компіляцію програм у проміжне представлення, яке називається кодом віртуальної машини, Java дозволяє створювати багатоплатформенні програми. Цей код може виконуватися в будь-якій системі, яка реалізує віртуальну машину Java.

Інтерпретованість і висока продуктивність. Виконуючи компіляцію програм у проміжне представлення, що називається кодом віртуальної машини, Java дозволяє створювати багатоплатформенні

програми. Цей код може виконуватися в будь-якій системі, котра реалізує віртуальну машину Java. Перші спроби отримання багатоплатформених рішень привели до досягнення поставленої цілі за рахунок зниження продуктивності. Код віртуальної машини Java був ретельно спроектований так, щоб за рахунок використання JIT-компіляції його можна було з високою продуктивністю легко перетворювати в машинозалежний код. Системи часу виконання Java, які надають цю функцію, зберігають всі переваги коду, який не залежить від платформи.

Розподілений характер. Мова Java призначена для розподіленого середовища Інтернет, оскільки вона підтримує протоколи сімейства TCP/IP. Фактично звернення до ресурсу через адресу URL не дуже відрізняється від звернення до файлу. Java підтримує також дистанційний виклик методів (Remote Method Invocation – RMI). Ця властивість дозволяє програмам викликати методи по мережі.

Динамічний характер. Програми Java містять значний обсяг інформації часу виконання, яка використовується для перевірки повноважень і надання доступу до об'єктів під час виконання. Це дозволяє виконувати безпечно і технічно виправдане динамічне зв'язування коду. Ця обставина виключно важлива для стійкості середовища Java, в якому невеликі фрагменти коду віртуальної машини можуть динамічно оновлюватися в діючій системі.

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмного продукту

Розроблений тренажер реалізовано мовою програмування Java у відповідності до парадигми об'єктно-орієнтованого програмування та алгоритму, який наведено в пункті 3.1.

Програмний продукт в повному обсязі забезпечує можливість впровадження в LMS Moodle віртуального навчального середовища ПУЕТ без нанесення шкоди системі та порушення системи безпеки, а також забезпечує стабільність та високу продуктивність під час роботи.

Також, забезпечена можливість завантаження і виконання програмного продукту на безлічі процесорів, операційних систем і браузерів, підключених до Інтернету.

Тренажер має зручний та зрозумілий інтерфейс користувача, який розроблений з використанням інструментарію бібліотеки Swing в інтегрованому середовищі розробки Apache NetBeans.

Використання інтегрованого середовища розробки, зокрема Apache NetBeans, дозволило відслідкувати та виправити програмні помилки, оскільки Apache NetBeans в повному обсязі надає розробнику потужні засоби для відладки програмних продуктів, що в свою чергу значно прискорює процес розробки.

Програмний продукт супроводжено повною програмною документацією, а саме:

- 1) алгоритмом (наведено в пункті 3.1);
- 2) блок-схемою (наведено в пункті 3.2);
- 3) інструкцією користувача (наведено в пункті 4.2);
- 4) повним вихідним кодом програмної реалізації (подано на компакт-диску з додатку Г).

4.2 Інструкція по використанню

Розроблений в рамках кваліфікаційної роботи тренажер з дисципліни «Методи оптимізації та дослідження операцій» реалізує розв'язування задач лінійного програмування модифікованим симплекс-методом. Реалізовано один приклад, який демонструє можливості тренінга користувачу по знаходженню розв'язку ЗЛП.

Для можливості використання розробленого програмного продукту, користувачу необхідно мати наступне:

- 1) IBM або MAC PC-сумісний комп'ютер;
- 2) високошвидкісне підключення до мережі Інтернет;
- 3) Інтернет-браузер;
- 4) JRE – мінімальну реалізацію віртуальної машини Java;
- 5) права доступу.

Оскільки, інструкція по використанню є досить об'ємною, її подано в додатку Б.

ВИСНОВКИ

МОДО – одна з фундаментальних дисциплін, яка забезпечує сучасний світогляд майбутнього спеціаліста з інформаційних технологій. Досягається це шляхом об'єднання теорії, створення математичної моделі та розрахунку. Традиційний спосіб викладання даної дисципліни не викликає ніяких складностей. Сучасні способи навчання передбачають пошук нових форм викладання МОДО. Одним з таких способів є використання віртуальних інтерактивному компонентів при викладанні цієї дисципліни.

Розроблений віртуальний тренажер дозволить студентам на практиці засвоїти навички розв'язування задач лінійного програмування модифікованим симплекс-методом. Як наслідок, це надасть можливість підвищити якість професійно-практичної підготовки студентів.

Основні результати роботи:

- ознайомлено з законодавчим та нормативно-правовим забезпеченням дистанційного навчання [3, 2];
- розглянуто провідні технології розробки інтерактивному компонентів для Web-середовищ;
- розглянуто основні формули та алгоритм МСМ;
- розглянуто відомі тренажери з математичних дисциплін;
- зроблено обґрунтований вибір програмного забезпечення для реалізації інтерактивного компонента;
- розроблено алгоритм інтерактивного компонента;
- реалізовано тренажер з теми «Модифікований симплекс-метод» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» з дотриманням основних вимог до програмного продукту;
- проведено тестування розробленого програмного продукту;
- розроблено програмну документацію;

- набуто навичок розробки програмного забезпечення для Web-середовища на мові програмування Java.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасні цифрові системи та технології дистанційного навчання в освіті : монографія / В. Ю. Биков та ін. ; за ред. В. Ю. Бикова, О. М. Спіріна. Київ : ІЦО НАПН України, 2022. 245 с.
2. Положення про дистанційне навчання. Затверджено Наказом Міністерства освіти і науки України від 25.04.2013 року, №466 / [Електронний ресурс]. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/z0703-13/>
3. Наказ Міністерства освіти і науки, молоді та спорту України від 15 вересня 2011 року № 1064 «Про впровадження педагогічного експерименту з дистанційного навчання на базі Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі» [Електронний ресурс]. – Режим доступу : http://search.ligazakon.ua/l_doc2.nsf/link1/MUS17092.html
4. Дистанційне навчання (педагогічний експеримент у Полтавському університеті економіки і торгівлі) / за заг. ред. В. М. Артеменка. Полтава : ПУЕТ, 2012. – 127 с.
5. Ємець О.О. Методи оптимізації та дослідження операцій: Навчально-методичний посібник до самостійного вивчення дисципліни для студентів спеціальності 6.080204 «Соціальна інформатика» та напряму 6.040302 «Інформатика» за кредитно-модульною системою організації навчального процесу / О.О.Ємець. – Полтава: РВВ ПУСКУ, 2009. - 76 с.
6. Ємець, О.О. Методи оптимізації та дослідження операцій: Навчально-методичний посібник за кредитно-модульною системою організації навчального процесу [Електронний ресурс] / О.О. Ємець – Полтава: ПУЕТ, 2013. – Режим доступу: <http://194.44.39.212/Library/2013/6/CYemecOO/136metCL/zmist.htm>. - (04.02.2014).

7. Ємець, О.О. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 8.080204 «Соціальна інформатика» / О.О.Ємець. - Полтава: РВВ ПУСКУ, 2009. – 41 с.
8. ДСТУ ISO 5807:2019 (ISO 5807:1985, IDT). Інформаційні технології. Символи документації, параметри програм і систем, схеми даних та лінії. [Чинний від 2020-01-01]. Київ : ДП «УкрНДНЦ», 2019. 30 с. (Національні стандарти України).
9. Кей С. Х. Java 2. Бібліотека професіонала Java 2./ С. Х. Кей, К. Р. Гарі - М.: Вільямс, 2008. - Т. I: Основи. – 816 с.
- 10.Брнакевич І. Є., Кардаш А. І. Об'єктно-орієнтоване програмування (на базі мови Java) : навч. посіб. Львів : ЛНУ ім. Івана Франка, 2022. 182 с.
- 11.Горстманн К. С. Java. Бібліотека професіонала. Т. 1. Основи : пер. з англ. Київ : Діалектика, 2021. 864 с.
- 12.Програмування та архітектура програмних систем на платформі Java : навч. посіб. / В. В. Кауліна та ін. Харків : ХНУРЕ, 2023. 240 с.
- 13.Коноваленко І. В. Програмування мовою Java : навч. посіб. Тернопіль : ТНТУ ім. І. Пулюя, 2023. 220 с.
- 14.Трохимчук Р. М. Комбінаторний аналіз: алгоритми та застосування : навч. посіб. Київ : Твімс інтер, 2011. 348 с.
- 15.Роберт С. Алгоритми на Java./ С. Роберт, У. Кевін- 4-е вид.- М.: Вільямс, 2012. - 848 с.
- 16.Бублик В. В. Об'єктно-орієнтоване програмування на Java : підручник. Київ : КПІ ім. Ігоря Сікорського, 2020. 324 с.
- 17.Моок С. Essential ActionScript 3.0. Beijing ; Boston : O'Reilly Media, 2007. 944 p.
- 18.Фрімен Е., Робсон Е. Патерни проектування : пер. з англ. Харків : Фабула, 2021. 672 с.
- 19.Rosenzweig G. ActionScript 3.0 Game Programming University [Текст] / G. Rosenzweig. – Que Publishing, 2011. – 567.

20. Fulton J. Essential Guide to Flash Games: Building Interactive Entertainment with ActionScript [Текст] / J. Fulton, S. Fulton. – Friends of ED, 2010. – 655.
21. Нефьодов О.І. Тренажер з теми «Розв’язування рекурентних співвідношень» дистанційного навчального курсу «Дискретна математика» та його програмуванняж./ Нефьодов О.І.// Інформатика та системні науки (ІСН–2013): Всеукраїнської науково-практичної конференції (21-23 березня 2013, Полтава). – Полтава: РВВ ПУЕТ, 2013 – 37-40 с.
22. Центр дистанційного навчання Сумського університету. [Електронний ресурс]: <http://dl.sumdu.edu.ua/ua/aboutus/department/kruskala> – Назва з екрану

ДОДАТОК А

Вихідний код програмної реалізації інтерактивного компонента

У зв'язку з значним обсягом програмного коду (понад 6000 рядків), лише частину його наведено в цьому додатку.

```

package edu.msm;

import java.awt.CardLayout;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.*;
import java.awt.event.*;

public class
ModifiedSimplexMethodSimulatorUI
extends javax.swing.JApplet {

    int stage;
    //step 9
    int x[] = {0, 0, 0, 360, 192, 180};
    int B[] = {1, 0, 0,
               0, 1, 0,
               0, 0, 1
    };
    int B_inverse[] = {1, 0, 0, 0, 1, 0, 0, 0,
1};
    int Cb[] = {0, 0, 0};
    int P0[] = {360, 192, 180};
    int Omega[] = {0, 0, 0};
    int F0 = 0;
    // step 10
    String Step10[] = {"9", "10", "16", "0", "0",
"0", //Ci 1..6
    "18", "6", "5", "15", "4", "3", "12",
"8", "3", "1", "0", "0", "0", "1", "0", "0", "0",
"1", //Pi 7..24
    "0", "0", "0", //Cb 25..27
    "360", "192", "180", //P0 28..30
    "0", "0", "0", //Omega 31..33
    "-9", "-10", "-16", "0", "0", "0",
//Delta 34..40
};
    int c_i[] = {9, 10, 16, 0, 0, 0};
    int P_i[] = {18, 6, 5,
    15, 4, 3,
    12, 8, 3,
    1, 0, 0,
    0, 1, 0,
    0, 0, 1
    };
    int delta_1[] = {-9, -10, -16, 0, 0, 0};

```

```

// step 13
int P_3[] = {12, 8, 3};
// step 15
int theta_i[] = {30, 24, 60};
// step 16
String row2[] = {"24", "0", "1/8", "0"};
// step 17
String mainTable2[] = {"1", "0", "0", "-3/2",
"1/8", "-3/8", "0", "0", "1", "0", "16", "0",
"72", "24", "108"};
// step 18
String Step18[] = {"0", "2", "0", "384"}; //
Omega 1..3, F0
// step19
String Step19[] = {"3", "-2", "0", "0", "2",
"0"}; // delta2
// step22
String Step22[] = {"9", "1/2", "3/2"};
// step24
String Step24[] = {"8", "48", "72"}; // theta
1..3
// step25
String Step25[] = {"8", "1/9", "-1/6", "0"};
// step26
String Step26[] = {"1/9", "-1/18", "-3/2", "-
1/6", "5/24", "-1/8", "0", "0", "1", // Aij
"10", "16", "0", //Cb
"8", "20", "96", //P0
"2/9", "5/3", "0", //Omega
"400" //F0};
String Step27[] = {"5", "0", "0", "2/9", "5/3", "0"};

public void init() {
    try {
        for
(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels())
{
            if
("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClas
sName());

                break;

            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ModifiedSimple
xMethodSimulatorUI.class

```

```

.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ModifiedSimple
xMethodSimulatorUI.class

.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ModifiedSimple
xMethodSimulatorUI.class

.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch
(javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(ModifiedSimple
xMethodSimulatorUI.class

.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        setSize(600, 550);
        try {
            java.awt.EventQueue.invokeLater(new
Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
...

private void
jButtonStep1NextActionPerformed(java.awt.event.Ac
tionEvent evt) {
    // TODO add your handling code here

    if (jRadioButtonStep1Yes.isSelected()) {
        CardLayout cl = (CardLayout)
(jPanelMain.getLayout());
        cl.show(jPanelMain, "step2");
    } else if
(jRadioButtonStep1No.isSelected()) {

```

```

        String message = "Вибрана відповідь
неправильна!\n"
        + "Задана ЗЛП в канонічній формі,
оскільки:\n"
        + "1) цільова функція прямує до
максимуму;\n"
        + "2 в системі обмежень всі обмеження
є рівностями;\n"
        + "3) всі змінні невід'ємні";
        JOptionPane.showMessageDialog(new
JFrame(), message, "Помилка!",
JOptionPane.ERROR_MESSAGE);
    }
}
private void
jButtonStep3NextActionPerformed(java.awt.event.Ac
tionEvent evt) {
    // TODO add your handling code here:
    int value = 0;
    byte i = 0;
    String message = "";
    if (stage <= 5) {
        try {
            switch (stage) {
                case 0:
                    value =
Integer.parseInt(jTextFieldStep3X1.get
Text());
                    i = 1;
                    break;
                case 1:
                    value =
Integer.parseInt(jTextFieldStep3X2.get
Text());
                    i = 2;
                    break;
                case 2:
                    value =
Integer.parseInt(jTextFieldStep3X3.get
Text());
                    i = 3;
                    break;
                case 3:
                    value =
Integer.parseInt(jTextFieldStep3X4.get
Text());
                    i = 4;
                    break;
                case 4:
                    value =
Integer.parseInt(jTextFieldStep3X5.get
Text());
                    i = 5;

```



```

        }
    }

    switch (stage) {
        case 1:

jTextFieldStep3X1.setEnabled(false);

jTextFieldStep3X2.setEnabled(true);
        jTextFieldStep3X2.requestFocus();
        break;
        case 2:

jTextFieldStep3X2.setEnabled(false);

jTextFieldStep3X3.setEnabled(true);
        jTextFieldStep3X3.requestFocus();
        break;
        case 3:

jTextFieldStep3X3.setEnabled(false);

jTextFieldStep3X4.setEnabled(true);
        jTextFieldStep3X4.requestFocus();
        break;
        case 4:

jTextFieldStep3X4.setEnabled(false);

jTextFieldStep3X5.setEnabled(true);
        jTextFieldStep3X5.requestFocus();
        break;
        case 5:

jTextFieldStep3X5.setEnabled(false);

jTextFieldStep3X6.setEnabled(true);
        jTextFieldStep3X6.requestFocus();
        break;
    }
    if (jCheckBoxStep3X1.isSelected() ||
jCheckBoxStep3X2.isSelected() ||
jCheckBoxStep3X3.isSelected()) {
        message = "<html><p>Базисна змінна
вибрана <span style=\"font: bold; color:
red;\">неправильно!</span></p></html>";
        JOptionPane.showMessageDialog(new
JFrame(), message, "Помилка!",
JOptionPane.ERROR_MESSAGE);
    } else if (jCheckBoxStep3X4.isSelected()
&& jCheckBoxStep3X5.isSelected() &&
jCheckBoxStep3X6.isSelected()) {

```

```

        message = "<html><p>Базисні змінні  

вибрано <span style=\"font: bold; color:  

green;\">правильно!</span></p></html>";
        JOptionPane.showMessageDialog(new
JFrame(), message, "",
JOptionPane.INFORMATION_MESSAGE);
        stage = 0; //reset
        CardLayout cl = (CardLayout)
(jPanelMain.getLayout());
        cl.show(jPanelMain, "step4");

    } else if
(jCheckBoxStep3X4.isSelected() ||
jCheckBoxStep3X5.isSelected() ||
jCheckBoxStep3X6.isSelected()) {
        message = "<html><p>Позначено не всі  

базисні змінні!</p></html>";
        JOptionPane.showMessageDialog(new
JFrame(), message, "Помилка!",
JOptionPane.ERROR_MESSAGE);
    }

}

...
private void
jButtonStep10NextActionPerformed(java.awt.event.A
ctionEvent evt) {
    // TODO add your handling code here:
    String value = "";
    String message = "";
    if (stage <= Step10.length) {
        switch (stage) {
            case 0:
                value =
jTextFieldStep10C1.getText();
                break;
            case 1:
                value =
jTextFieldStep10C2.getText();
                break;
            case 2:
                value =
jTextFieldStep10C3.getText();
                break;
            case 3:
                value =
jTextFieldStep10C4.getText();
                break;
            case 4:
                value =
jTextFieldStep10C5.getText();
                break;
            case 5:

```

```
        value =
jTextFieldStep10C6.getText();
        break;
    case 6:
        value =
jTextFieldStep10P11.getText();
        break;
    case 7:
        value =
jTextFieldStep10P12.getText();
        break;
    case 8:
        value =
jTextFieldStep10P13.getText();
        break;
    case 9:
        value =
jTextFieldStep10P21.getText();
        break;
    case 10:
        value =
jTextFieldStep10P22.getText();
        break;
    case 11:
        value =
jTextFieldStep10P23.getText();
        break;
    case 12:
        value =
jTextFieldStep10P31.getText();
        break;
    case 13:
        value =
jTextFieldStep10P32.getText();
        break;
    case 14:
        value =
jTextFieldStep10P33.getText();
        break;
    case 15:
        value =
jTextFieldStep10P41.getText();
        break;
    case 16:
        value =
jTextFieldStep10P42.getText();
        break;
    case 17:
        value =
jTextFieldStep10P43.getText();
        break;
    case 18:
```

```
        value =
jTextFieldStep10P51.getText();
        break;
    case 19:
        value =
jTextFieldStep10P52.getText();
        break;
    case 20:
        value =
jTextFieldStep10P53.getText();
        break;
    case 21:
        value =
jTextFieldStep10P61.getText();
        break;
    case 22:
        value =
jTextFieldStep10P62.getText();
        break;
    case 23:
        value =
jTextFieldStep10P63.getText();
        break;
    case 24:
        value =
jTextFieldStep10Cb1.getText();
        break;
    case 25:
        value =
jTextFieldStep10Cb2.getText();
        break;
    case 26:
        value =
jTextFieldStep10Cb3.getText();
        break;
    case 27:
        value =
jTextFieldStep10P0_1.getText();
        break;
    case 28:
        value =
jTextFieldStep10P0_2.getText();
        break;
    case 29:
        value =
jTextFieldStep10P0_3.getText();
        break;
    case 30:
        value =
jTextFieldStep10Omega1.getText();
        break;
    case 31:
```

```

        value =
jTextFieldStep10Omega2.getText();
        break;
        case 32:
            value =
jTextFieldStep10Omega3.getText();
            break;
        case 33:
            value =
jTextFieldStep10Delta11.getText();
            break;
        case 34:
            value =
jTextFieldStep10Delta12.getText();
            break;
        case 35:
            value =
jTextFieldStep10Delta13.getText();
            break;
        case 36:
            value =
jTextFieldStep10Delta14.getText();
            break;
        case 37:
            value =
jTextFieldStep10Delta15.getText();
            break;
        case 38:
            value =
jTextFieldStep10Delta16.getText();
            break;
    }

    if (stage < 6) {
        if
(value.compareTo(Step10[stage]) == 0) { //Ci
            stage++;
        } else if (value.length() == 0) {
            message = "Необхідно
заповнити!";

JOptionPane.showMessageDialog(new JFrame(),
"Помилка!", JOptionPane.ERROR_MESSAGE);
            message,
        } else {
            message = "";

JOptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new ImageIcon(getClass().getResource("/img/step10
-error1.png")));
        }
    }

```

```

        } else if (stage >= 6 && stage < 24)
{ //Pi
        if
(value.compareTo(Step10[stage]) == 0) {
                stage++;
        } else if (value.length() == 0) {
                message = "Необхідно
заповнити!";

JOptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE);
        } else {
                message = "";

JOptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new
ImageIcon(getClass().getResource("/img/step10-
error4.png")));
        }
        } else if (stage >= 24 && stage < 27)
{ //Cb
        if
(value.compareTo(Step10[stage]) == 0) {
                stage++;
        } else if (value == "") {
                message = "Значення введено неправильно!";

JOptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE);
        } else {
                message = "";

JOptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new
ImageIcon(getClass().getResource("/img/step10-
error2.png")));
        }
        } else if (stage >= 27 && stage < 30)
{ //P0
        if
(value.compareTo(Step10[stage]) == 0) {
                stage++;
        } else if (value == "") {
                message = "Значення введено неправильно!";

JOptionPane.showMessageDialog(new JFrame(), message,
"Помилка!", JOptionPane.ERROR_MESSAGE);
        } else {
                message = "";

```

```

OptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new
ImageIcon(getClass().getResource("/img/step10-
error3.png")));
    }
    } else if (stage >= 30 && stage < 33)
{ //Omega
    if
(value.compareTo(Step10[stage]) == 0) {
        stage++;
    } else if (value == "") {
        message = "Значення введено
неправильно!";
        JOptionPane.showMessageDialog(new
JFrame(), message, "Помилка!",
JOptionPane.ERROR_MESSAGE);
    } else {
        message = "";

OptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new ImageIcon(getClass().getResource("/img/step9-
error1.png")));
    }
    } else if (stage >= 33 && stage < 39)
{ //Delta
    if
(value.compareTo(Step10[stage]) == 0) {
        stage++;
    } else if (value == "") {
        message = "Значення введено
неправильно!";

OptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE);
    } else {
        message = "";

OptionPane.showMessageDialog(new JFrame(),
message, "Помилка!", JOptionPane.ERROR_MESSAGE,
new
ImageIcon(getClass().getResource("/img/step10-
error5.png")));
    }
    }
    }
    switch (stage) {
        case 1:
jTextFieldStep10C1.setEnabled(false);

```

```
jTextFieldStep10C2.setEnabled(true);

jTextFieldStep10C2.requestFocus();
    break;
    case 2:

jTextFieldStep10C2.setEnabled(false);

jTextFieldStep10C3.setEnabled(true);

jTextFieldStep10C3.requestFocus();
    break;
    case 3:

jTextFieldStep10C3.setEnabled(false);

jTextFieldStep10C4.setEnabled(true);

jTextFieldStep10C4.requestFocus();
    break;
    case 4:

jTextFieldStep10C4.setEnabled(false);

jTextFieldStep10C5.setEnabled(true);

jTextFieldStep10C5.requestFocus();
    break;
    case 5:

jTextFieldStep10C5.setEnabled(false);

jTextFieldStep10C6.setEnabled(true);

jTextFieldStep10C6.requestFocus();
    break;
    case 6:

jTextFieldStep10C6.setEnabled(false);

jTextFieldStep10P11.setEnabled(true);

jTextFieldStep10P11.requestFocus();
    break;
    case 7:

jTextFieldStep10P11.setEnabled(false);

jTextFieldStep10P12.setEnabled(true);

jTextFieldStep10P12.requestFocus();
    break;
```

```
        case 8:
jTextFieldStep10P12.setEnabled(false);
jTextFieldStep10P13.setEnabled(true);
jTextFieldStep10P13.requestFocus();
        break;
        case 9:
jTextFieldStep10P13.setEnabled(false);
jTextFieldStep10P21.setEnabled(true);
jTextFieldStep10P21.requestFocus();
        break;
        case 10:
jTextFieldStep10P21.setEnabled(false);
jTextFieldStep10P22.setEnabled(true);
jTextFieldStep10P22.requestFocus();
        break;
        case 11:
jTextFieldStep10P22.setEnabled(false);
jTextFieldStep10P23.setEnabled(true);
jTextFieldStep10P23.requestFocus();
        break;
        case 12:
jTextFieldStep10P23.setEnabled(false);
jTextFieldStep10P31.setEnabled(true);
jTextFieldStep10P31.requestFocus();
        break;
        case 13:
jTextFieldStep10P31.setEnabled(false);
jTextFieldStep10P32.setEnabled(true);
jTextFieldStep10P32.requestFocus();
        break;
        case 14:
jTextFieldStep10P32.setEnabled(false);
jTextFieldStep10P33.setEnabled(true);
```

```
jTextFieldStep10P33.requestFocus();
    break;
    case 15:

jTextFieldStep10P33.setEnabled(false);

jTextFieldStep10P41.setEnabled(true);

jTextFieldStep10P41.requestFocus();
    break;
    case 16:

jTextFieldStep10P41.setEnabled(false);

jTextFieldStep10P42.setEnabled(true);

jTextFieldStep10P42.requestFocus();
    break;
    case 17:

jTextFieldStep10P42.setEnabled(false);

jTextFieldStep10P43.setEnabled(true);

jTextFieldStep10P43.requestFocus();
    break;
    case 18:

jTextFieldStep10P43.setEnabled(false);

jTextFieldStep10P51.setEnabled(true);

jTextFieldStep10P51.requestFocus();
    break;
    case 19:

jTextFieldStep10P51.setEnabled(false);

jTextFieldStep10P52.setEnabled(true);

jTextFieldStep10P52.requestFocus();
    break;
    case 20:

jTextFieldStep10P52.setEnabled(false);

jTextFieldStep10P53.setEnabled(true);

jTextFieldStep10P53.requestFocus();
    break;
    case 21:
```

```
jTextFieldStep10P53.setEnabled(false);  
  
jTextFieldStep10P61.setEnabled(true);  
  
jTextFieldStep10P61.requestFocus();  
    break;  
    case 22:  
  
jTextFieldStep10P61.setEnabled(false);  
  
jTextFieldStep10P62.setEnabled(true);  
  
jTextFieldStep10P62.requestFocus();  
    break;  
    case 23:  
  
jTextFieldStep10P62.setEnabled(false);  
  
jTextFieldStep10P63.setEnabled(true);  
  
jTextFieldStep10P63.requestFocus();  
    break;  
    case 24:  
  
jTextFieldStep10P63.setEnabled(false);  
  
jTextFieldStep10Cb1.setEnabled(true);  
  
jTextFieldStep10Cb1.requestFocus();  
    break;  
    case 25:  
  
jTextFieldStep10Cb1.setEnabled(false);  
  
jTextFieldStep10Cb2.setEnabled(true);  
  
jTextFieldStep10Cb2.requestFocus();  
    break;  
    case 26:  
  
jTextFieldStep10Cb2.setEnabled(false);  
  
jTextFieldStep10Cb3.setEnabled(true);  
  
jTextFieldStep10Cb3.requestFocus();  
    break;  
    case 27:  
  
jTextFieldStep10Cb3.setEnabled(false);  
  
jTextFieldStep10P0_1.setEnabled(true);
```

```
jTextFieldStep10P0_1.requestFocus();
    break;
    case 28:

jTextFieldStep10P0_1.setEnabled(false);

jTextFieldStep10P0_2.setEnabled(true);

jTextFieldStep10P0_2.requestFocus();
    break;
    case 29:

jTextFieldStep10P0_2.setEnabled(false);

jTextFieldStep10P0_3.setEnabled(true);

jTextFieldStep10P0_3.requestFocus();
    break;
    case 30:

jTextFieldStep10P0_3.setEnabled(false);

jTextFieldStep10Omega1.setEnabled(true);

jTextFieldStep10Omega1.requestFocus();
    break;
    case 31:

jTextFieldStep10Omega1.setEnabled(false);

jTextFieldStep10Omega2.setEnabled(true);

jTextFieldStep10Omega2.requestFocus();
    break;
    case 32:

jTextFieldStep10Omega2.setEnabled(false);

jTextFieldStep10Omega3.setEnabled(true);

jTextFieldStep10Omega3.requestFocus();
    break;
    case 33:

jTextFieldStep10Omega3.setEnabled(false);

jTextFieldStep10Delta11.setEnabled(true);

jTextFieldStep10Delta11.requestFocus();
    break;
    case 34:
```

```

jTextFieldStep10Delta11.setEnabled(false);

jTextFieldStep10Delta12.setEnabled(true);

jTextFieldStep10Delta12.requestFocus();
    break;
    case 35:

jTextFieldStep10Delta12.setEnabled(false);

jTextFieldStep10Delta13.setEnabled(true);

jTextFieldStep10Delta13.requestFocus();
    break;
    case 36:

jTextFieldStep10Delta13.setEnabled(false);

jTextFieldStep10Delta14.setEnabled(true);

jTextFieldStep10Delta14.requestFocus();
    break;
    case 37:

jTextFieldStep10Delta14.setEnabled(false);

jTextFieldStep10Delta15.setEnabled(true);

jTextFieldStep10Delta15.requestFocus();
    break;
    case 38:

jTextFieldStep10Delta15.setEnabled(false);

jTextFieldStep10Delta16.setEnabled(true);

jTextFieldStep10Delta16.requestFocus();
    break;
    case 39:

jTextFieldStep10Delta16.setEnabled(false);
    stage = 0; //reset
    CardLayout cl = (CardLayout)
(jPanelMain.getLayout());
    cl.show(jPanelMain, "step11");
    }
}

...
private void jButtonStep21delta6StateChanged(javax.swing.
event.ChangeEvent evt) {
    if (jRadioButtonStep21delta6.isSelected()) {

```

```
jRadioButtonStep21delta6.setText("<html>&uarr;</html>");
    } else {
        jRadioButtonStep21delta6.setText(" ");
    }
    jButtonStep21Next.setEnabled(true);
}
```

ДОДАТОК Б

Інструкція по використанню

Нижче буде наведено інструкцію по використанню інтерактивного компонента, в якій розглянуто роботу з етапами тренінгу, що відрізняються за методом відповіді (заповнення текстових полів, вибір селектором, позначення «Прапорцем», комбінована відповідь) на запитання та завдання інтерактивного компонента. Також буде подано інструкцію, щодо дій, в разі неправильної відповіді на запитання або завдання тренінгу.

Метод відповіді: «Вибір селектором». Завдання такого типу потребують позначення варіанта відповіді, шляхом натискання лівої кнопки миші на області селектора та підтвердження вибору кнопкою «Продовжити», яка розміщена на екрані (див. рисунок Б.1).

У разі допущення помилки, на екран виводиться повідомлення з інформацією про помилку та підказкою (див. рисунок Б.1). Після ознайомлення з інформацією, у діалоговому вікні, необхідно натиснути кнопку «Ок», обрати правильну відповідь, вищевказаним методом та натиснути кнопку «Продовжити» для переходу на наступний етап тренінгу.

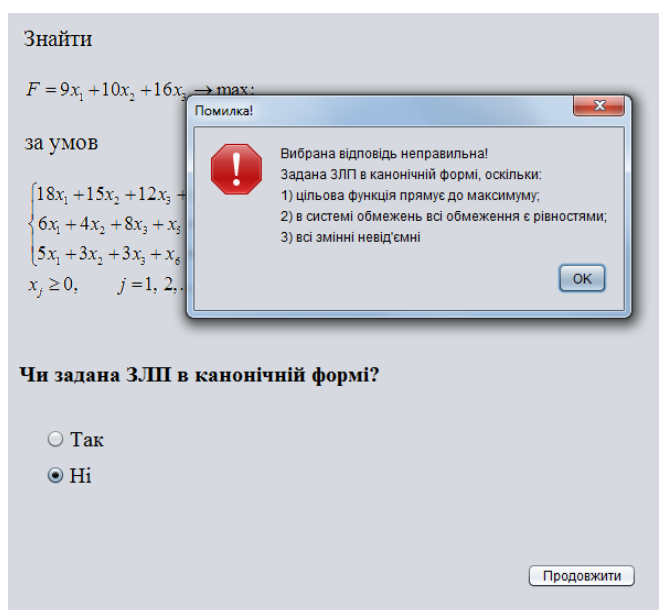


Рисунок Б.1 – Вибір варіанта відповіді методом «Вибір селектором» та діалогове вікно з повідомлення про помилку та інформацією *Комбінований метод відповіді: «Заповнення текстових полів та вибір селектором»*. Необхідно одне за одним заповнити порожні текстові поля числовими значеннями, шляхом введення останніх з клавіатури, щоразу підтверджуючи введення кнопкою «Продовжити».

У разі, якщо залишити текстове поле пустим та натиснути кнопку «Продовжити» – на екран виведеться повідомлення про те, що необхідно заповнити текстове поле значенням.

У разі, якщо ввести неправильне значення – на екран виведеться повідомлення про помилку (див. рисунок Б.2).

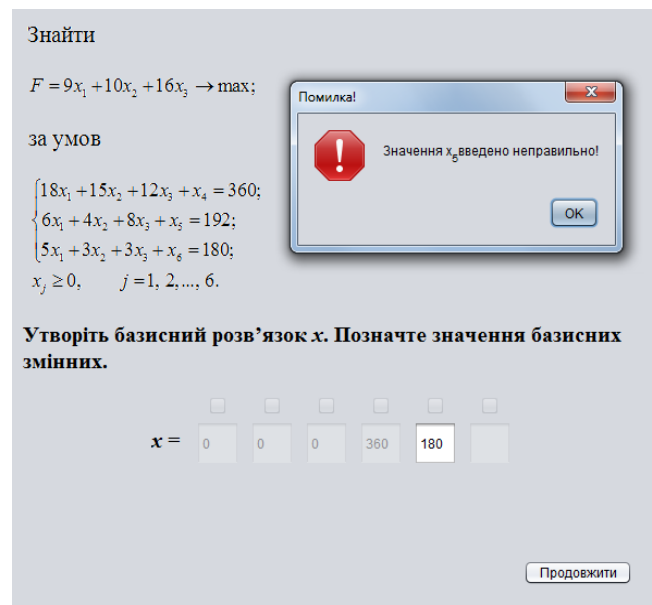


Рисунок Б.2 – Заповнення текстових полів числовими значеннями методом відповіді «Заповнення текстових полів» та повідомлення про помилку

Далі необхідно позначити базиси «Прапорцями», шляхом натискання лівої кнопки миші на областях «Прапорців» та натиснути кнопку «Продовжити».

Якщо всі базиси обрано правильно – на екран виводиться повідомлення з інформацією про успішне проходження цього етапу тренінгу (див. рисунок Б.3).

Після натискання кнопки «Ок», у діалоговому вікні інформаційного повідомлення, відбувається перехід на наступний етап тренінгу.

Знайти

$$F = 9x_1 + 10x_2 + 16x_3 \rightarrow \max;$$

за умов

$$\begin{cases} 18x_1 + 15x_2 + 12x_3 + x_4 = 360; \\ 6x_1 + 4x_2 + 8x_3 + x_5 = 192; \\ 5x_1 + 3x_2 + 3x_3 + x_6 = 180; \\ x_j \geq 0, \quad j=1, 2, \dots, 6. \end{cases}$$

Утворіть базисний розв'язок x . Позначте значення базисних змінних.

$x =$ 0 0 0 360 192 180

Продовжити

Базисні змінні вибрано **правильно!**

ОК

Рисунок Б.3 – Позначення базисних змінних методом відповіді «Позначення прапорцем» та інформаційне повідомлення

У разі, якщо позначено не всі базиси – на екран виводиться повідомлення про помилку (див. рисунок Б.4). У цьому випадку, користувачу необхідно повернутися і обрати всі базисні змінні та натиснути кнопку «Продовжити».

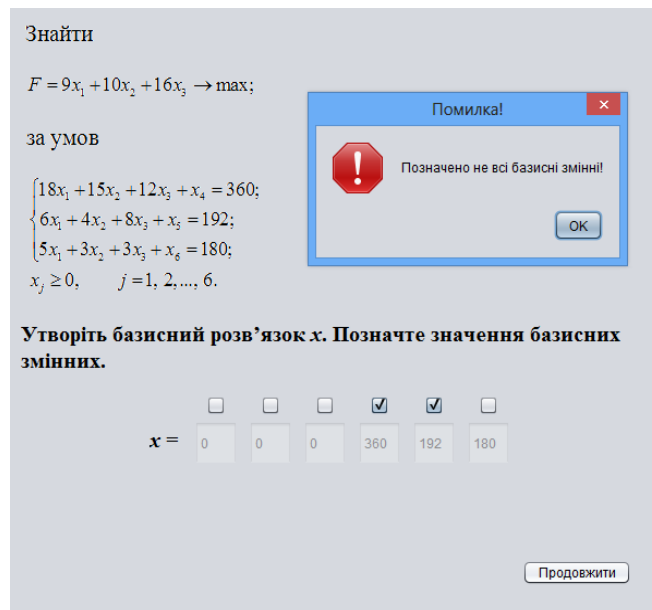


Рисунок Б.4 – Позначення базисних змінних методом відповіді «Позначення прапорцем» та повідомлення про помилку

У разі, якщо базиси позначено неправильно – на екран виводиться повідомлення про помилку (див. рисунок Б.5). У цьому випадку, необхідно повернутися і позначити базиси правильно, після чого натиснути кнопку «Продовжити».

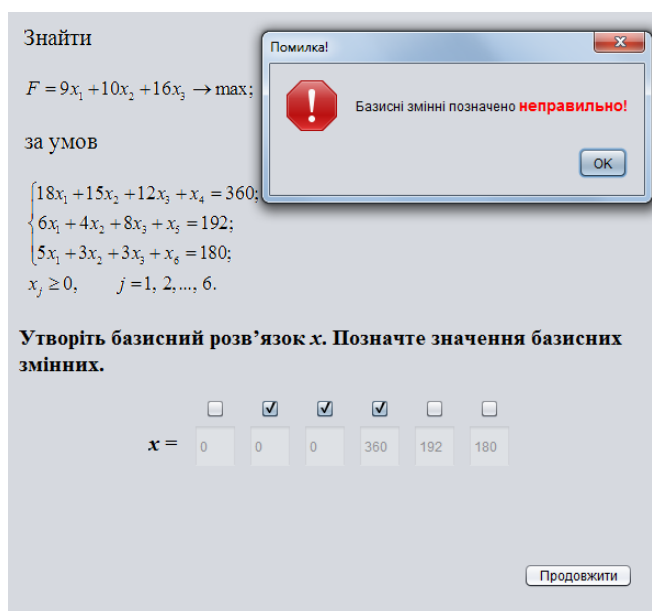


Рисунок Б.5 – Позначення базисних змінних методом відповіді «Позначення прапорцем»

та повідомлення про помилку:

«Базисні змінні позначено неправильно!»

Метод відповіді: «Вибір селектором». Необхідно дати відповідь на запропоноване запитання, шляхом натискання лівої кнопки миші на областях «Прапорців» та натиснути кнопку «Продовжити» (див. рисунок Б.6).

У разі допущення помилки, на екран виведеться повідомлення про помилку та підказка у вигляді обрамлення міток з правильними та неправильними варіантами відповідей, зеленим чи червоним кольорами відповідно (див. рисунок Б.6). У цьому випадку, необхідно повернутися і позначити правильні варіанти відповіді «Прапорцями», у відповідності до підказки, після чого натиснути кнопку «Продовжити» для переходу на наступний етап тренінгу.

Всі інші запитання і завдання інтерактивного компонента мають аналогічні, щодо розглянутих вище, методи відповіді та інформування про помилки і підказки, а отже їх розгляд є недоцільним.

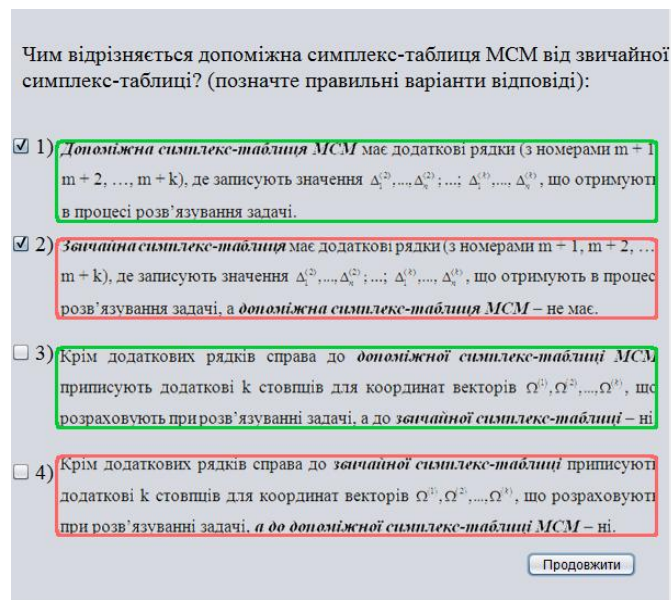


Рисунок Б.6 – Позначення базисних змінних та підказка