

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

**ПОБУДОВА АЛГОРИТМУ НАВЧАЛЬНОГО ТРЕНУВАЛЬНОГО
СИМУЛЯТОРА НА ТЕМУ «ЗАМКНЕНІ КЛАСИ. ФУНКЦІОНАЛЬНА
ПОВНОТА БУЛЕВИХ ФУНКЦІЙ» ДИСТАНЦІЙНОГО КУРСУ
«ДИСКРЕТНА МАТЕМАТИКА» ТА ЙОГО ПРОГРАМНА РЕАЛІЗАЦІЯ**

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

Виконавець роботи Бугаєнко Радомир Ігорович

_____ «___» _____ 2024 р.

(підпис)

Науковий керівник к. ф.-м. н., доцент, Парфьонова Тетяна Олександрівна

_____ «___» _____ 2024 р.

(підпис)

Рецензент _____

ПОЛТАВА 2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ, ТЕРМІНІВ..	3
ВСТУП.....	4
1. ПОСТАНОВКА ЗАВДАННЯ.....	7
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Огляд інструментів, що використовуються у дистанційному навчанні	8
2.2. Огляд основного теоретичного матеріалу.....	9
2.3. Огляд існуючих подібних програм	13
3. ТЕОРЕТИЧНА ЧАСТИНА.....	17
3.1. Алгоритм реалізації елемента навчального симулятора «Замкнені класи булевих функцій».....	17
3.2. Алгоритм реалізації елемента навчального симулятора «Функціональна повнота булевих функцій»	23
4. ПРАКТИЧНА ЧАСТИНА	28
4.1. Блок-схема алгоритму роботи тренажера	28
4.2. Опис проектних рішень, інструментів та підходів до розробки.....	30
4.3. Інструкція користувача	36
ВИСНОВКИ.....	40
СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	41
ДОДАТОК А. КОД ПРОГРАМИ.....	43

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

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
ДК	Дистанційний курс
ДН	Дистанційне навчання
ПУЕТ	Полтавський університет економіки і торгівлі

ВСТУП

Умови сьогодення вимагають активного використання різних інтерактивних методів як складової освітнього процесу. Їх застосування допомагає студентам ефективніше засвоювати навчальний матеріал. Одним із поширених ефективних способів навчання є використання навчальних тренувальних симуляторів, які дозволяють студентам опанувати практичні навички, застосовуючи теоретичні знання з теми.

Базовою дисципліною під час підготовки фахівця спеціальності Комп'ютерні науки є «Дискретна математика», яка містить розділи різного спрямування. Одним із важливих таких розділів присвячений вивченню булевих функцій. Зокрема, освоєння теми «Замкнені класи. Функціональна повнота булевих функцій» в рамках дистанційного курсу «Дискретна математика», є особливо важливим, так як є основою для розвитку логічного мислення та розв'язання складних завдань у даній галузі. Крім того, елементи цієї теми досить широко використовуються у програмуванні, так як для створення ефективних алгоритмів та програм необхідно чітко розуміти особливості функціонування логічних операцій та булевих функцій.

Актуальність. Побудова алгоритму навчального тренувального симулятора з даної теми сприяє кращому засвоєнню матеріалу студентами та допомагає практично застосовувати отримані знання. В процесі реалізації теми крім алгоритму необхідно написати програмне забезпечення до нього. Це вимагає певних знань та вмінь, що також сприяють поглибленню знань у своїй майбутній професії. Тому створення подібний програмних продуктів є актуальним і необхідним.

Мета роботи – створити програмне забезпечення для навчального тренувального симулятора на тему «Замкнені класи. Функціональна повнота булевих функцій» дистанційного курсу «Дискретна математика».

Основні завдання роботи:

- Огляд літератури з теми «Замкнені класи. Функціональна повнота булевих функцій» навчального курсу «Дискретна математика» та подібних програм.

- Повторення та огляд основних понять теми, зокрема, операцій замикання, типів замкнених класів та їх властивостей, поняття функціональної повноти булевих функцій. Вивчення взаємозв'язку між замкненими класами та функціональною повнотою булевих функцій.

- Розробка алгоритму роботи навчального тренувального симулятора для засвоєння даної теми.

- Вибір програмних засобів для реалізації мети роботи.

- Створення програми для функціонування навчального симулятора.

Об'єкт – алгоритмізація та розробка програмного забезпечення структурного елемента дистанційного навчального курсу.

Предмет – навчальний тренувальний симулятор для вивчення теми «Замкнені класи. Функціональна повнота булевих функцій» дистанційного курсу «Дискретна математика».

Для розробки симулятора було обрано: мову програмування C#; мову розмітки XAML; технологію створення графічного інтерфейсу користувача WPF; середовище розробки Microsoft Visual Studio.

Щодо структури, то робота складається із чотирьох розділів, які в повній мірі описують хід виконання роботи. Перший розділ – один із найвідповідальніших, це постановка задачі. Тут чітко і зрозуміло сформульовано основні вимоги до процесу програмування і до результату. У інформаційному (другому) розділі здійснено огляд основних понять теми та огляд вже існуючих програмних продуктів, що використовуються у дистанційному навчанні. Третій розділ присвячено теоретичній частині, де сформульовано алгоритми роботи навчального симулятора. У четвертому розділі описана практична реалізація, в тому числі інструкція користувача.

Створений навчальний симулятор працює коректно і може бути використаний для вивчення теми дистанційного курсу «Дискретна математика» «Замкнені класи. Функціональна повнота булевих функцій». Використання таких ефективних інструментів, які враховують потреби користувачів, є актуальним та необхідним.

Обсяг пояснювальної записки: 65 сторінок, в т. ч. основна частина - 43 сторінки, джерела - 8 назв.

1. ПОСТАНОВКА ЗАВДАННЯ

Згідно мети роботи необхідно створити програмне забезпечення для навчального тренувального симулятора на тему «Замкнені класи. Функціональна повнота булевих функцій» дистанційного курсу «Дискретна математика».

Вимоги до навчального симулятора.

1) Стартова сторінка має містити наступну інформацію:

- тема тренажеру «Замкнені класи та функціональна повнота булевих функцій»;
- інформація про автора (прізвище та ініціали виконавця), університет, прізвище та ініціали керівника;
- кнопка з посиланням на теоретичні відомості з теми;
- дві кнопки «Замкнені класи булевих функцій» та «Функціональна повнота булевих функцій» (чи одна, наприклад, «Початок тренінгу», яка переводить на іншу сторінку із цими кнопками чи вкладками). Передбачити, щоб проходження тренінгу «Функціональна повнота булевих функцій» було можливим лише після проходження тренінгу з розділу «Замкнені класи булевих функцій».

В розділі «Замкнені класи булевих функцій» реалізувати теоретичні питання з прикладами щодо поняття замикання та замкнених класів, які передбачені алгоритмом.

В розділі «Функціональна повнота булевих функцій» реалізувати теоретичні питання щодо теореми Поста, а також реалізацію наступного завдання: «Визначити за допомогою теореми Поста, чи є система $\{\bar{x}, x \vee y\}$ функціонально повною.».

2) Тренажер має навчальну функцію, тому програма має постійно реагувати на дії користувача. Зокрема, в разі помилкових відповідей на кожному кроці алгоритму має з'являтися повідомлення, що інформує про помилку, містить підказки чи роз'яснення питання, а також правильну відповідь.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд інструментів, що використовуються у дистанційному навчанні

Дистанційне навчання (ДН) відіграє дуже важливу роль у сучасному світі, особливо у зв'язку з поширенням цифрових технологій та доступністю Інтернету. Розглянемо деякі з них. 1) ДН відкриває можливості для людей з усього світу отримувати освіту, навіть у віддалених районах, де недоступні традиційні навчальні заклади. 2) Студентам ДН надається можливість самостійно керувати своїм часом і вчитися у зручний для них графік та темп. 3) Використання різноманітних інтерактивних методів навчання може зробити ДН ефективним способом отримання знань і навичок. 4) ДН допомагає зекономити витрати на поїздки до навчальних закладів та інші пов'язані з навчанням витрати. 5) ДН дозволяє отримати нові знання та підвищити кваліфікацію, не втрачаючи часу на перерви у роботі або інші зобов'язання.

Для ефективної реалізації ДН суттєве значення має створення якісних дистанційних курсів (ДК). Для їх створення існує багато різних інструментів, які можна використовувати. Зупинимось на розгляді деяких, найпопулярніших з них.

1. Платформи для управління навчальним процесом, такі як Moodle, Canvas, Blackboard, Schoology, які надають засоби для створення курсів, завдань, тестів, спілкування зі студентами та інші функції.

2. Відеохостингові сервіси, такі як YouTube, Vimeo, Panopto, які дозволяють завантажувати відеоуроки та матеріали для студентів.

3. Інструменти відеоконференцій, такі як Zoom, Microsoft Teams, Google Meet, що дозволяють проводити онлайн-зустрічі, вебінари та взаємодіяти зі студентами у реальному часі.

4. Інтерактивні інструменти для створення вправ та тестів, такі як Quizlet, Kahoot, Quizizz, які допомагають створювати цікаві та взаємодійні завдання для студентів.

5. Інструменти для створення електронних книг та матеріалів, такі як Canva, Adobe Spark, Book Creator, які дозволяють створювати ілюстровані матеріали для навчання.

Розглянуті інструменти можна комбінувати та використовувати разом для створення ефективних та цікавих дистанційних курсів.

Одним із важливих структурних елементів ДК є навчальні симулятори, які відіграють важливу роль у ДН, оскільки дозволяють студентам отримати практичний досвід і вміння застосовувати набуті знання до реальних практичних задач. Застосування подібних тренувальних програм сприяє активному навчанню, де студенти можуть вчитися шляхом виконання завдань, вирішення проблем та взаємодії з матеріалом; збільшує зацікавленість студентів у навчанні. Деякі тренажери крім навчальної, мають ще й контролюючу функцію, що дозволяє оцінити рівень навичок та знань студентів, та надати відповідні рекомендації. Навчальні симулятори можуть бути адаптовані до потреб кожного студента, дозволяючи їм навчатися у власному темпі та відповідно до їхніх потреб.

Отже, використання навчальних тренажерів у ДН сприяє покращенню якості навчання та забезпечує студентам можливість розвивати практичні навички та вміння.

2.2. Огляд основного теоретичного матеріалу

Вивчення булевих функцій має суттєве значення у програмуванні, адже їх основна властивість та, що вони працюють з двома можливими значеннями: ІСТИНА (відповідне значення булевої функції 1) та ХИБНІСТЬ (відповідне значення булевої функції 0). Вони є ключовими для розробки ефективних та логічних програм.

Діапазон застосування булевих функцій досить широкий. Зокрема, булеві функції використовуються для 1) визначення умов щодо виконання певних дій у програмах (наприклад, умовні оператори if, if ... else, if...else if...else базуються на результаті булевих виразів); 2) виконання логічних операцій, таких як І (AND), АБО (OR), НЕ (NOT); 3) допомоги управління потоком виконання програми, визначаючи умови для виконання певних дій або вибору між різними гілками коду; 4) порівняння значень, виконання перевірок умов та прийняття рішень на їх основі. Крім зазначеного, булеві функції можуть бути використані для побудови будь-якої іншої булевої функції через їхню функціональну повноту, що є важливим аспектом у теорії обчислень.

Функціональна повнота тісно пов'язана із поняттями замкнених класів булевих функцій. Розглянемо деякі основні означення.

Нехай P_2 – множина всіх булевих функцій n змінних. А саме:

$$P_2 = \{y = f(x_1, \dots, x_n); x_i, y \in \{0, 1\}, i = 1, 2, \dots, n\}.$$

Виберемо підмножину $m \subset P_2$.

Замикання m – це множина всіх булевих функцій, які можна представити у вигляді формул, де використовуються функції лише із множини m . Позначення замикання – $[m]$. Наприклад, якщо $m = \{1, x_1 + x_2\}$, то замикання $[m] = L$, де L – це сукупність канонічних багаточленів Жегалкіна без кон'юнкцій змінних.

Множина m називається *функціонально замкнутою* або просто *замкнутою* якщо виконується умова $[m] = m$. Інша назва замкнутої множини – *замкнений клас*.

Розглянемо основні п'ять найважливіших замкнених класів булевих функцій.

Клас K_0 . Клас булевих функцій $f_i(x_1, \dots, x_n)$, що зберігають константу 0.

Це така множина булевих функцій, для елементів якої виконується умова:

$$f_i(0, \dots, 0) = 0,$$

Іншими словами для наборів змінних, які приймають нульові значення, функція має бути також рівною нулю.

Наприклад, еквівалентність $x_1 \sim x_2$ не належить класу K_0 , так як на наборі $x_1 = 0, x_2 = 0$ дана функція дорівнює одиниці ($x_1 \sim x_2 \notin K_0$). А диз'юнкція $x_1 \vee x_2$ належить класу K_0 ($x_1 \vee x_2 \in K_0$), так як на наборі $x_1 = 0, x_2 = 0$ дорівнює нулю.

Клас K_1 . Клас булевих функцій $f_i(x_1, \dots, x_n)$, що зберігають константу 1.

Це така множина булевих функцій, для елементів якої виконується умова:

$$f_i(1, \dots, 1) = 1,$$

Іншими словами для наборів змінних, які приймають одиничні значення, функція має бути також рівною одиниці.

Наприклад, сума по модулю 2 $x_1 + x_2$ не належить класу K_1 , так як на наборі $x_1 = 1, x_2 = 1$ дана функція дорівнює нулю ($x_1 + x_2 \notin K_1$). Кон'юнкція $x_1 \wedge x_2$ належить класу K_1 ($x_1 \wedge x_2 \in K_1$), так як на наборі $x_1 = 1, x_2 = 1$ дорівнює одиниці.

Клас L . Клас лінійних булевих функцій. Це множина, яка складається з функцій, що в алгебрі Жегалкіна представляються канонічними багаточленами у вигляді:

$$a_0 + a_1x_1 + \dots + a_nx_n,$$

Як бачимо, ці вирази не містять кон'юнкцій, тут коефіцієнти $a_0, a_1, \dots, a_n$ – це булеві змінні.

Наприклад, заперечення $\bar{x} \in L$, так як її можна представити як $\bar{x} = 1 + x$. Диз'юнкція $x_1 \vee x_2 \notin L$, так як її можна представити многочленом Жегалкіна, що містить кон'юнкцію x_1x_2 , тобто $x_1 \vee x_2 = x_1 + x_1 + x_1x_2$.

Клас S . Клас самодвоїстих булевих функцій. Це множина самодвоїстих булевих функцій. Іншими словами має задовольняти умові:

$$f_i(x_1, \dots, x_n) = \bar{f}_i(\bar{x}_1, \dots, \bar{x}_n)$$

для довільних $x_1, \dots, x_n$.

Наприклад, заперечення $\bar{x} \in S$, так як заперечення функції $\bar{\bar{x}} = x$ буде дорівнювати функції $\bar{x}$.

Диз'юнкція $x_1 \vee x_2 \notin S$, так як $\overline{\overline{x_1} \vee \overline{x_2}} = \overline{\bar{x}_1} \wedge \overline{\bar{x}_2} = x_1 \wedge x_2 \neq x_1 \vee x_2$.

Якщо позначити протилежні набори як $(x_1, \dots, x_n)$ та $(\overline{x_1}, \dots, \overline{x_n})$, то самодвоїстою є булева функція, яка приймає протилежні значення на всіх протилежних наборах.

Нехай $\preceq$ – відношення передування (частковий порядок на множині векторів $\mathbf{n}$ булевих змінних), яке представимо наступним чином:

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \preceq \beta = (\beta_1, \beta_2, \dots, \beta_n),$$

Тут має виконуватись $\alpha_i \leq \beta_i$ для всіх $i = 1, 2, \dots, n$.

Наприклад, $(0, 0, 0, 1, 0, 1) \preceq (1, 1, 0, 1, 0, 1)$.

Клас M . Клас монотонних булевих функцій.

Це множина монотонних функцій, а саме таких функцій, що для будь-яких наборів змінних $\alpha = (\alpha_1, \dots, \alpha_n)$ та $\beta = (\beta_1, \dots, \beta_n)$ задовольняють умові: $f(\alpha) \leq f(\beta)$, тобто $f(\alpha_1, \dots, \alpha_n) \leq f(\beta_1, \dots, \beta_n)$.

Тут α та β мають знаходитися у відношенні передування $\alpha \preceq \beta$.

Наприклад, $x_1 \wedge x_2 \in M$.

Система функцій, підстановкою яких може бути одержана будь-яка функція з деякої множини булевих функцій, називається функціонально повною в множині. Якщо при цьому в системі допускаються сталі 0 та 1, то вона називається ослаблено функціонально повною.

Система функцій з замкнутого класу $\mathbf{m}$ називається його базисом, якщо вона є повною в $\mathbf{m}$, а будь-яка її власна підсистема не є повною в $\mathbf{m}$.

Теорема Поста-Яблонського про функціональну повноту. Для того, щоб система булевих функцій була повною необхідно і достатньо, щоб вона цілком не містилася в жодному з замкнених класів K_0, K_1, L, M, S .

З даної теореми випливає, що кожна повна система булевих функцій з необхідністю і достатністю має принаймні одну функцію, яка не зберігає 0; принаймні одну функцію, яка не зберігає 1; принаймні одну функцію, яка не є лінійною; принаймні одну функцію, що не є монотонною; принаймні одну функцію, що не є самодвоїстою.

У таблиці 2.1 розглянуто деякі булеві функції, для яких плюс вказує на те, що відповідна функція має вказану у відповідному стовпці властивості

незбереження нуля та незбереження одиниці, властивості нелінійності, несамоодвійності та немонотонності.

Таблиця 2.1 – Властивості булевих функцій

Назва функції	Властивості				
	Незбереження		Несамодвійність	Нелінійність	Немонотонність
	0	1			
Константа 0		+	+		
Константа 1	+		+		
Заперечення	+	+			+
Кон'юнкція			+	+	
Диз'юнкція			+	+	
Імплікація	+		+	+	+
Еквіваленція	+		+		+
Сума за модулем 2		+	+		+
Штрих Шеффера	+	+	+	+	+
Стрілка Пірса	+	+	+	+	+
Заперечення імплікації		+	+	+	+

Вибравши будь-яку функцію з таблиці 2.1 так, щоб виконувалась умова теореми про функціональну повноту, одержимо певний базис. Таким чином можна отримати різні базиси, наприклад, базис Жегалкіна $\{+, \wedge, 1\}$.

Важливо відмітити, що теорема про функціональну повноту поширюється на довільну кількість змінних у функції.

2.3. Огляд існуючих подібних програм

На сьогоднішній день існує велика кількість різноманітних навчальних тренажерів, що охоплюють різні предмети, навички та області знань. Завдяки швидкому розвитку технологій і збільшенню доступності Інтернету, віртуальні навчальні середовища стають все популярнішими.

Розглянемо деякі навчальні симулятори, які розроблені студентами спеціальності "Комп'ютерні науки" ПУЕТ.

Тренажер з теми «Елементи векторної алгебри» дистанційного навчального курсу «Алгебра та геометрія. Частина 1», автор Михайловський Роман (рис. 2.1). Під час реалізації використано середовище розробки Microsoft Visual Studio та мова програмування C#.

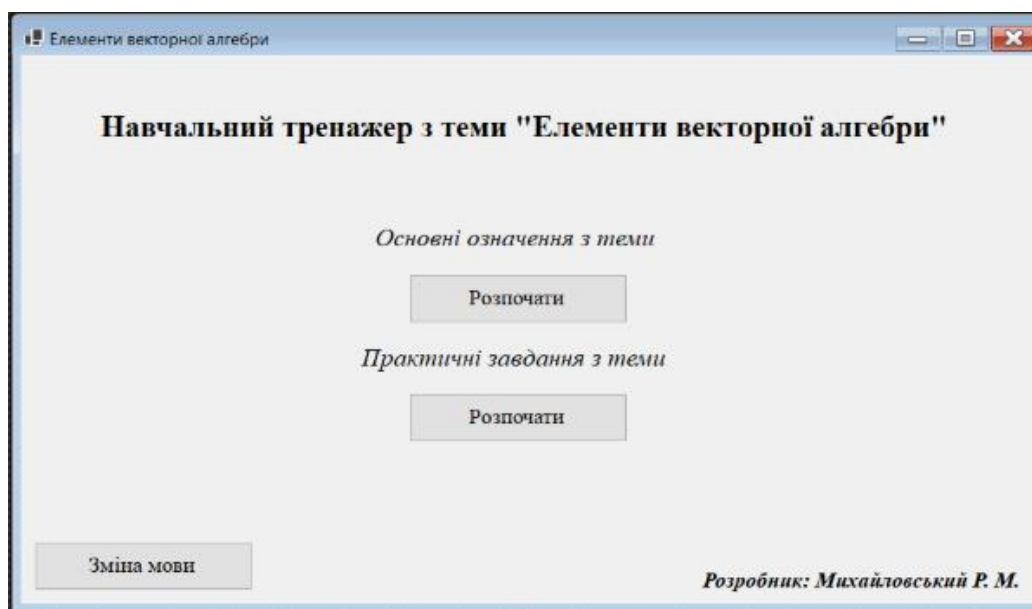


Рисунок 2.1 – Головна сторінка тренажеру Михайловського.

В програмі передбачено англomовна версія тренінгу, що дозволяє можливість проходження тренінгу іноземними студентами (рис. 2.2).

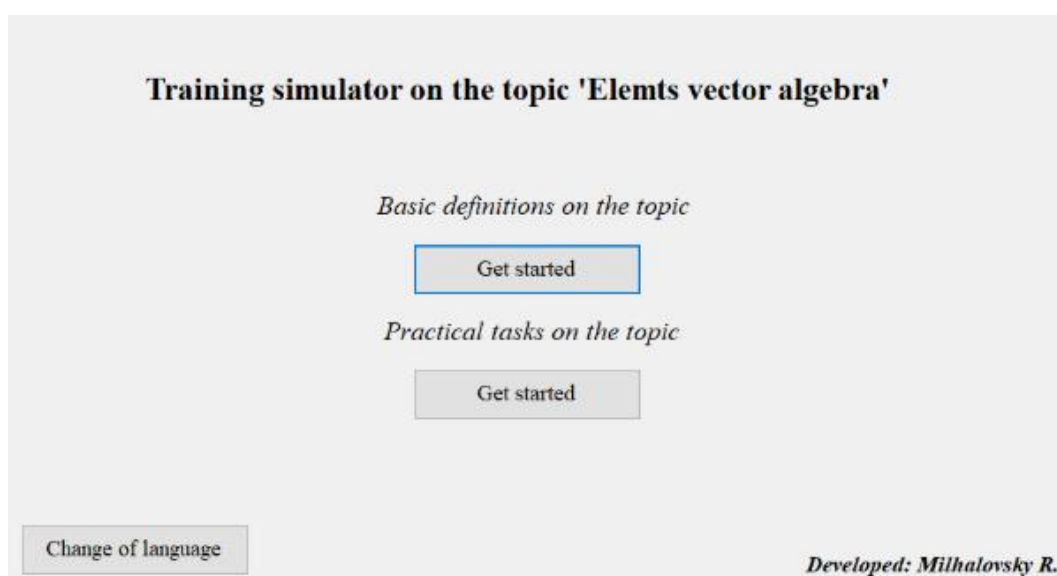


Рисунок 2.2 – Головна сторінка тренажеру на англійській мові.

В алгоритмі передбачено питання, які реалізуються різним чином, зокрема, з можливістю введення відповіді у клітинку, вибрати із запропонованих варіантів, та інше (рис.2.3).

Елементи векторної алгебри

Приклад 1. Дані дві точки $A(-3, 6, -5)$ та $B(6, -6, 3)$.
Знайти: координати та довжину вектора $\overrightarrow{AB}$.

Крок 2. Вписати відповідь. Обрахувати координати вектора $\overrightarrow{AB}$.

$\overrightarrow{AB} = (\square \square \square, \square \square \square, \square \square \square) =$
 $= (\square, \square, \square)$

$\overrightarrow{AB} = (6 - (-3), (-6) - 6, 3 - (-5))$

Повернутися до початку Відповісти

Рисунок 2.3 – Один із прикладів реалізації тренажеру Михаловського.

Далі розглянемо приклад навчального симулятора, який розроблено Хорольським В.О. Реалізовано алгоритм тренінгу на тему «Решітки та булеві алгебри» мовою C++ у середовищі Borland Builder (рис. 2.4).

Решітки та булеві алгебри

Тренажер з теми
"Решітки та булеві алгебри"
ДК "Дискретна математика"

Теми тренінгу :

- ☒ 1) Деякі поняття, необхідні для означення решітки.
- ☐ 2) Решітки.

Розпочати

Розробник: Хорольський В.О., гр. КН-41.
Каф. ММСІ, ПУЕТ, 2021 р.

Рисунок 2.4 – Симулятор «Решітки та булеві алгебри».

В оглянутих навчальних тренінгових програмах передбачені перевірка правильності відповідей користувача та реагування програми відповідними повідомленнями. Так як в першу чергу основна функція таких програм – навчальна, то варто надавати рекомендації, та й в решті решт правильні відповіді з поясненнями. На жаль, це не всіма програмами реалізується. А саме, алгоритмом тренажера для курсу «Дискретна математика» з теми «Планарні граfi. Розфарбування граfiв», розробленого Коломійцем Максимом, не передбачає наявності пояснень, а надає лише інформативні повідомлення щодо правильності вибору відповіді (рис. 2.5).

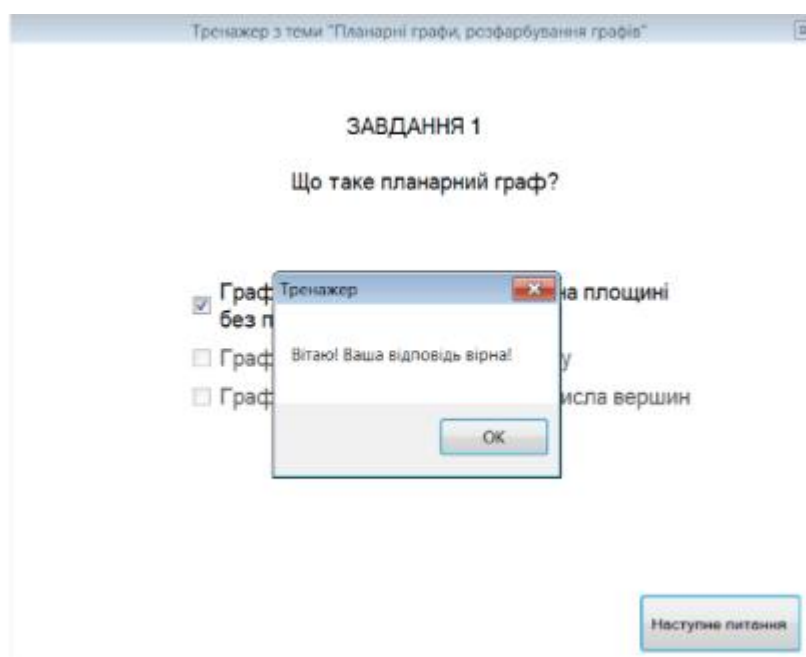


Рисунок 2.5 – Симулятор з теми «Планарні граfi. Розфарбування граfiв».

Під час програмної реалізації тренажерів необхідно передбачати можливість впроваджувати створені програми у дистанційний курс, на відповідні платформи, наприклад, Moodle. Важливим також аспектом при створенні навчальних симуляторів є вибір практичних задач та набору необхідних теоретичних питань. Тут важливо, щоб питання максимально охоплювали тему, були чітко і зрозуміло сформульовані. Це потребує достатньо високого рівня підготовки розробника.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм реалізації елемента навчального симулятора «Замкнені класи булевих функцій»

Крок 1. На екрані наступна інформація:

«Позначимо множину всіх булевих функцій n змінних через P_2 .

Тобто $P_2 = \{y = f(x_1, x_2, \dots, x_n); x_i, y \in \{0, 1\}, i = 1, 2, \dots, n\}$. Нехай $t \subset P_2$.

Замиканням t називається множина всіх булевих функцій, що можна представити у вигляді формул, використовуючи функції тільки з множини t . Позначають замикання $[t]$.

Множина t називається замкненою (замкненим класом) (або «функціонально замкненою»), якщо $[t] = t$.

Розглянемо п'ять основних замкнених класів булевих функцій.»

В подальших кроках 2-6 передбачити випадковий порядок варіантів відповідей.

Крок 2. На екрані завдання: *Вибрати одну правильну відповідь.*

Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, для елементів якої виконується умова: $f_i(0, 0, \dots, 0) = 0$, (тобто, якщо всі змінні – нулі, то і значення функції – нуль) називається

- А) класом M монотонних булевих функцій
- Б) класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1
- В) класом S самодвоїстих булевих функцій
- Г) класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0
- Д) класом L лінійних булевих функцій

Користувач має вибрати правильну відповідь.

В разі вибору відповіді Г – перехід на крок 3. В іншому випадку – з'являється повідомлення про помилку: «Вибір помилковий. Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, для елементів якої виконується умова: $f_i(0, 0, \dots, 0) = 0$,

(тобто, якщо всі змінні – нулі, то і значення функції – нуль) називається **класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0.**». Перехід на крок 3.

Крок 3. На екрані завдання: *Вибрати одну правильну відповідь.*

Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, для елементів якої виконується умова: $f_i(1, 1, \dots, 1)=1$, (тобто, якщо всі змінні – одиниці, то і значення функції – одиниця) називається

- А) класом M монотонних булевих функцій
- Б) класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1
- В) класом S самодвоїстих булевих функцій
- Г) класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0
- Д) класом L лінійних булевих функцій

В разі вибору відповіді Б – перехід на крок 4. В іншому випадку – з'являється повідомлення про помилку: «Вибір помилковий. Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, для елементів якої виконується умова: $f_i(1, 1, \dots, 1)=1$, (тобто, якщо всі змінні – одиниці, то і значення функції – одиниця) називається **класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1.**». Перехід на крок 4.

Крок 4. На екрані завдання: *Вибрати одну правильну відповідь.*

Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що складається з функцій, які в алгебрі Жегалкіна представляються канонічними многочленами вигляду:

$$a_0 + a_1x_1 + \dots + a_nx_n,$$

(тобто многочлени, що не містять «добутків»), де коефіцієнти $a_0, a_1, \dots, a_n$ є булевими змінними, називається

- А) класом M монотонних булевих функцій
- Б) класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1
- В) класом S самодвоїстих булевих функцій
- Г) класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0
- Д) класом L лінійних булевих функцій

В разі вибору відповіді Д – перехід на крок 5. В іншому випадку – з’являється повідомлення про помилку: «Вибір помилковий. Множина булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що складається з функцій, які в алгебрі Жегалкіна представляються канонічними многочленами вигляду: $a_0 + a_1x_1 + \dots + a_nx_n$, (тобто многочлени, що не містять «добутків»), де коефіцієнти $a_0, a_1, \dots, a_n$ є булевими змінними, називається **класом L лінійних булевих функцій**». Перехід на крок 5.

Крок 5. На екрані завдання: *Вибрати одну правильну відповідь.*

Множина булевих функцій, які задовольняють умові

$$f_i(x_1, x_2, \dots, x_n) = \bar{f}_i(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \text{ для будь-яких } x_1, x_2, \dots, x_n \text{ називається}$$

- А) класом M монотонних булевих функцій
- Б) класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1
- В) класом S самодвоїстих булевих функцій
- Г) класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0
- Д) класом L лінійних булевих функцій

В разі вибору відповіді В – перехід на крок 6. В іншому випадку – з’являється повідомлення про помилку: «Вибір помилковий. Множина булевих функцій, які задовольняють умові $f_i(x_1, x_2, \dots, x_n) = \bar{f}_i(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ для будь-яких $x_1, x_2, \dots, x_n$ називається **класом S самодвоїстих булевих функцій**». Перехід на крок 6.

Крок 6. На екрані завдання: *Вибрати одну правильну відповідь.*

Множина таких булевих функцій, що для довільних наборів змінних

$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ і $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, які знаходяться у відношенні передування $\alpha \preceq \beta$, задовольняють умові $f(\alpha) \leq f(\beta)$, або іншими словами:

$$f(\alpha_1, \alpha_2, \dots, \alpha_n) \leq f(\beta_1, \beta_2, \dots, \beta_n), \text{ називається}$$

- А) класом M монотонних булевих функцій
- Б) класом K_1 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 1
- В) класом S самодвоїстих булевих функцій
- Г) класом K_0 булевих функцій $f_i(x_1, x_2, \dots, x_n)$, що зберігають константу 0
- Д) класом L лінійних булевих функцій

В разі вибору відповіді А – перехід на крок 7. В іншому випадку – з’являється повідомлення про помилку: «Вибір помилковий. Множина таких булевих функцій, що для довільних наборів змінних $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ і $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, які знаходяться у відношенні передування $\alpha \preceq \beta$, задовольняють умові $f(\alpha) \leq f(\beta)$, або іншими словами: $f(\alpha_1, \alpha_2, \dots, \alpha_n) \leq f(\beta_1, \beta_2, \dots, \beta_n)$, називається **класом М монотонних булевих функцій**». Перехід на крок 7.

Крок 7. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються відповідні умови:*

- | | |
|--|---|
| ☐ $x_1 \vee x_2 \in K_0$ | ☐ $x_1 + x_2 \in K_0$ |
| ☐ $x_1 \wedge x_2 \in K_0$ | ☐ $x_1 \rightarrow x_2 \in K_0$ |
| ☐ $x_1 \sim x_2 \in K_0$ | ☐ $\bar{x} \in K_0$ |

Якщо обрано умови $x_1 \vee x_2 \in K_0$, $x_1 \wedge x_2 \in K_0$, $x_1 + x_2 \in K_0$ – перехід на крок 8. Інакше – повідомлення: «Вибір помилковий. Правильні відповіді – $x_1 \vee x_2 \in K_0$, $x_1 \wedge x_2 \in K_0$, $x_1 + x_2 \in K_0$, оскільки для цих булевих функцій виконується умова $f_i(0, 0, \dots, 0)=0$, тобто

$$x_1 \vee x_2 = 0 \vee 0 = 0, \text{ отже, } x_1 \vee x_2 \in K_0,$$

$$x_1 \wedge x_2 = 0 \wedge 0 = 0, \text{ отже, } x_1 \wedge x_2 \in K_0,$$

$$x_1 + x_2 = 0 + 0 = 0, \text{ отже, } x_1 + x_2 \in K_0.$$

Для булевих функцій $x_1 \sim x_2$, $x_1 \rightarrow x_2$, $\bar{x}$ умова $f_i(0, 0, \dots, 0)=0$ не виконується, тобто $x_1 \sim x_2 = 0 \sim 0 = 1$, $x_1 \rightarrow x_2 = 0 \rightarrow 0 = 1$, $\bar{x} = \bar{0} = 1$.

Отже, $x_1 \sim x_2 \notin K_0$, $x_1 \rightarrow x_2 \notin K_0$, $\bar{x} \notin K_0$ ». Перехід на крок 8.

Крок 8. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються відповідні умови:*

- | | |
|--|---|
| ☐ $x_1 \vee x_2 \in K_1$ | ☐ $x_1 + x_2 \in K_1$ |
| ☐ $x_1 \wedge x_2 \in K_1$ | ☐ $x_1 \rightarrow x_2 \in K_1$ |
| ☐ $x_1 \sim x_2 \in K_1$ | ☐ $\bar{x} \in K_1$ |

Якщо обрано умови $x_1 \vee x_2 \in K_1$, $x_1 \wedge x_2 \in K_1$, $x_1 \sim x_2 \in K_1$, $x_1 \rightarrow x_2 \in K_1$ – перехід на крок 8. Інакше – повідомлення: «Вибір помилковий. Правильні

відповіді – $x_1 \vee x_2 \in K_1$, $x_1 \wedge x_2 \in K_1$, $x_1 \sim x_2 \in K_1$, $x_1 \rightarrow x_2 \in K_1$, оскільки для цих булевих функцій виконується умова $f_i(1, 1, \dots, 1)=1$, тобто

$$x_1 \vee x_2 = 1 \vee 1 = 1, \text{ отже, } x_1 \vee x_2 \in K_1,$$

$$x_1 \wedge x_2 = 1 \wedge 1 = 1, \text{ отже, } x_1 \wedge x_2 \in K_1,$$

$$x_1 \sim x_2 = 1 \sim 1 = 1, \text{ отже, } x_1 \sim x_2 \in K_1,$$

$$x_1 \rightarrow x_2 = 1 \rightarrow 1 = 1, \text{ отже, } x_1 \rightarrow x_2 \in K_1.$$

Для булевих функцій $x_1 + x_2$, $\bar{x}$ умова $f_i(1, 1, \dots, 1)=1$ не виконується, тобто $x_1 + x_2 = 1 + 1 = 0$, $\bar{x} = \bar{1} = 0$. Отже, $x_1 + x_2 \notin K_1$, $\bar{x} \notin K_1$.». Перехід на крок 9.

Крок 9. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються відповідні умови:*

- | | |
|--|---|
| ☐ $x_1 \vee x_2 \in L$ | ☐ $x_1 + x_2 \in L$ |
| ☐ $x_1 \wedge x_2 \in L$ | ☐ $x_1 \rightarrow x_2 \in L$ |
| ☐ $x_1 \sim x_2 \in L$ | ☐ $\bar{x} \in L$ |

Якщо обрано умови $x_1 \sim x_2 \in L$, $x_1 + x_2 \in L$, $\bar{x} \in L$ – перехід на крок 10. Інакше – повідомлення: «Вибір помилковий. Правильні відповіді – $x_1 \sim x_2 \in L$, $x_1 + x_2 \in L$, $\bar{x} \in L$, оскільки $x_1 + x_2$ вже представлена многочленом алгебри Жегалкіна вигляду $a_0 + a_1x_1 + \dots + a_nx_n$, а також булеві функції $x_1 \sim x_2$, $\bar{x}$ в алгебрі Жегалкіна можна представити канонічними многочленами вигляду: $a_0 + a_1x_1 + \dots + a_nx_n$, а саме: $x_1 \sim x_2 = 1 + x_1 + x_2$, отже, $x_1 \sim x_2 \in L$, $\bar{x} = 1 + x$, отже, $\bar{x} \in L$.

Булеві функції $x_1 \vee x_2$, $x_1 \wedge x_2$, $x_1 \rightarrow x_2$, що записані в алгебрі Жегалкіна містять «добутки» $x_1 \wedge x_2$, тобто $x_1 \vee x_2 = x_1 + x_2 + x_1 \wedge x_2$,

$$x_1 \rightarrow x_2 = 1 + x_1 + x_1 \wedge x_2. \text{ Отже, } x_1 \vee x_2 \notin L, x_1 \wedge x_2 \notin L, x_1 \rightarrow x_2 \notin L.$$

Перехід на крок 10.

Крок 10. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються відповідні умови:*

- | | |
|---|--|
| ☐ $x_1 \vee x_2 \in S$ | ☐ $x_1 \wedge x_2 \in S$ |
| ☐ $x_1 + x_2 + x_3 \in S$ | ☐ $\bar{x} \in S$ |

Якщо обрано варіанти $x_1 + x_2 + x_3 \in S$ та $\bar{x} \in S$ – перехід на крок 11. Інакше – повідомлення: «Вибір помилковий. Одна з правильних відповідей – $x_1 + x_2 + x_3 \in S$, оскільки для цієї булевої функції виконується умова $f_i(x_1, x_2, x_3) = \bar{f}_i(\bar{x}_1, \bar{x}_2, \bar{x}_3)$. Тобто враховуючи, що $\bar{x} = 1 + x$, $x + x = 0$, $0 + x = x$, маємо

$\bar{x}_1 + \bar{x}_2 + \bar{x}_3 = 1 + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 = 1 + 1 + x_1 + 1 + x_2 + 1 + x_3 = x_1 + x_2 + x_3$, отже, $x_1 + x_2 + x_3 \in S$. Спробуйте повторити вибір, перевіrivши відповідну умову для інших функцій.».

Після повторного вибору в разі помилки з'являється повідомлення: «Вибір помилковий. Правильною крім $x_1 + x_2 + x_3 \in S$, є відповідь $\bar{x} \in S$, оскільки для цієї функції також виконується $f_i(x_1, x_2, x_3) = \bar{f}_i(\bar{x}_1, \bar{x}_2, \bar{x}_3)$, тобто $\bar{\bar{x}} = \bar{x}$, отже, $\bar{x} \in S$.

Для функцій $x_1 \vee x_2$ та $x_1 \wedge x_2$ дана умова не виконується, тобто

$$\bar{x}_1 \vee \bar{x}_2 = \bar{x}_1 \wedge \bar{x}_2 = x_1 \wedge x_2 \neq x_1 \vee x_2, \text{ отже, } x_1 \vee x_2 \notin S,$$

$$\bar{x}_1 \wedge \bar{x}_2 = \bar{x}_1 \vee \bar{x}_2 = x_1 \vee x_2 \neq x_1 \wedge x_2, \text{ отже, } x_1 \wedge x_2 \notin S. \text{ »}.$$

Перехід на крок 11.

Крок 11. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються наступні відношення передування:*

- | | |
|-------------------------------------|-------------------------------------|
| ○ $(0,1,0,0,1) \preceq (0,1,0,0,0)$ | ○ $(1,1,0,0,1) \preceq (1,1,0,1,1)$ |
| ○ $(0,0,1,0,1) \preceq (0,0,1,1,1)$ | ○ $(1,1,1,0,0) \preceq (1,0,1,0,0)$ |

Якщо обрано варіанти $(0,0,1,0,1) \preceq (0,0,1,1,1)$ та $(1,1,0,0,1) \preceq (1,1,0,1,1)$ – перехід на крок 11. Інакше – повідомлення: «Вибір помилковий. Правильними є варіанти $(0,0,1,0,1) \preceq (0,0,1,1,1)$ та $(1,1,0,0,1) \preceq (1,1,0,1,1)$.

Разом з цим не виконуються

– $(0,1,0,0,1) \preceq (0,1,0,0,0)$, так як для виділених елементів на 5 позиції не виконується умова передування $\preceq$, тобто **1** не передує **0**.

– $(1,1,1,0,0) \preceq (1,0,1,0,0)$, так як для виділених елементів на 2 позиції не виконується умова передування $\preceq$, тобто **1** не передує **0**».

Крок 12. На екрані завдання: *Вибрати варіанти відповідей, для яких виконуються відповідні умови:*

- $x_1 \vee x_2 \in M$
- $x_1 \wedge x_2 \in M$
- $x_1 \sim x_2 \in M$

Якщо обрано варіанти $x_1 \vee x_2 \in M$ та $x_1 \wedge x_2 \in M$ – завершення алгоритму. Інакше – повідомлення: «Вибір помилковий. Одна з правильних відповідей –

$x_1 \vee x_2 \in M$. Для булевої функції $x_1 \vee x_2$ виконується $0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1$. Розглянемо пари наборів, які перебувають у відношенні передування $\preceq$. $(0,0) \preceq (0,1)$, звідки $0 \preceq 1$. $(0,0) \preceq (1,1)$, звідки $0 \preceq 1$. $(0,1) \preceq (1,1)$, звідки $1 \preceq 1$. $(1,0) \preceq (1,1)$, звідки $1 \preceq 1$. Отже, $x_1 \vee x_2 \in M$. Спробуйте повторити вибір, перевіряючи відповідні умови для інших функцій.»

Після повторного вибору в разі помилки з'являється повідомлення: «Вибір помилковий. Правильною, крім $x_1 \vee x_2 \in M$, є відповідь $x_1 \wedge x_2 \in M$. Для булевої функції $x_1 \wedge x_2$ виконується $0 \wedge 0 = 0, 0 \wedge 1 = 0, 1 \wedge 0 = 0, 1 \wedge 1 = 1$. Маємо для пар наборів, які перебувають у відношенні передування $\preceq$: $(0,0) \preceq (0,1)$, звідки $0 \preceq 0$. $(0,0) \preceq (1,1)$, звідки $0 \preceq 1$. $(0,1) \preceq (1,1)$, звідки $0 \preceq 1$. $(1,0) \preceq (1,1)$, звідки $0 \preceq 1$. Отже, $x_1 \wedge x_2 \in M$.

Функція $x_1 \sim x_2 \notin M$. Для цієї функції виконується:

$$0 \sim 0 = 1, 0 \sim 1 = 0, 1 \sim 0 = 0, 1 \sim 1 = 1.$$

Тоді $(0,0) \preceq (0,1)$, звідки $1 \preceq 0$ не виконується. Отже, $x_1 \sim x_2 \notin M$ ».

Роботу алгоритму завершено.

3.2. Алгоритм реалізації елемента навчального симулятора «Функціональна повнота булевих функцій»

Крок 1. На екрані завдання: Виберіть правильні варіанти.

Теорема Поста. Для того, щоб система логічних функцій була функціонально повною, необхідно і достатньо, щоб ця система містила хоча би одну функцію, що:

- не зберігає нуль,
- не зберігає одиницю,
- не є лінійною,
- не є самодвоїстою,
- не є монотонною.

Правильним є виділення всіх варіантів.

Якщо деякі варіанти не були відмічені, то з'являється повідомлення:

«Теорема Поста. Для того, щоб система логічних функцій була функціонально повною, необхідно і достатньо, щоб ця система містила хоча би одну функцію, що: не зберігає нуль, не зберігає одиницю, не є лінійною, не є самодвоїстою, не є монотонною.». Перехід на крок 2.

Крок 2. На екрані наступне завдання:

«Визначити за допомогою теореми Поста, чи є система $\{\bar{x}, x \vee y\}$ функціонально повною.

В наступній таблиці у комірках введіть позначки:

«+», якщо властивість для відповідної булевої функції виконується,
«-» - в іншому випадку» (табл.3.1).

Таблиця 3.1 – Таблиця для введення на кроці 2 алгоритму.

<i>Назва властивості</i>	$\bar{x}$	$x \vee y$
Збереження 0	☐	☐
Збереження 1	☐	☐
Лінійність	☐	☐
Самодвоїстість	☐	☐
Монотонність	☐	☐

Користувачу надається можливість заповнити кожен рядок по черзі, тобто на кожному етапі введення активні лише дві клітинки поточного рядка. Після введення здійснюється перевірка.

Якщо введено хоча б в одну комірку *першого* рядка неправильний символ, то з'являється наступне повідомлення з поясненням:

«Збереження 0 для булевої функції $\bar{x}$ не виконується, так як для $x = 0$
 $\bar{x} = \bar{0} = 1 \neq 0$.

Збереження 0 для булевої функції $x \vee y$ виконується, так як для $x = 0$,
 $y = 0$ $x \vee y = 0 \vee 0 = 0$.».

Для *другого* рядка в разі помилки повідомлення матиме вигляд:

«Збереження 1 для булевої функції $\bar{x}$ не виконується, так як для $x = 1$
 $\bar{x} = \bar{1} = 0 \neq 1$.

Збереження 1 для булевої функції $x \vee y$ виконується, так як для $x = 1$,
 $y = 1$ $x \vee y = 1 \vee 1 = 1$.».

Для *третього* рядка в разі помилки повідомлення матиме вигляд:

«Лінійність для булевої функції $\bar{x}$ виконується, так як її можна задати многочленом Жегалкіна $\bar{x} = 1 + x$.

Лінійність для булевої функції $x \vee y$ не виконується, так як її можна задати многочленом Жегалкіна $x \vee y = x + y + xy$, яка містить кон'юнкцію xy .».

Для *четвертого* рядка в разі помилки повідомлення матиме вигляд:

«Самодвоїстість для булевої функції $\bar{x}$ виконується, так як $\bar{\bar{x}} = x$.

Самодвоїстість для булевої функції $x \vee y$ не виконується, так як
 $\overline{x \vee y} = \bar{x} \wedge \bar{y} = x \wedge y \neq x \vee y$.».

Для *n'ятого* рядка в разі помилки повідомлення матиме вигляд:

«Монотонність для булевої функції $\bar{x}$ не виконується, так як не
справджується $\bar{0} = 1 \not\leq \bar{1} = 0$.

«Монотонність для булевої функції $\bar{x}$ не виконується, так як

$$\bar{0} = 1 \not\leq \bar{1} = 0$$

Монотонність для булевої функції $x \vee y$ виконується, так як для пар, що перебувають у відношенні $\preceq$ справедливо наступне:

$(0, 0) \preceq (0, 1)$, звідки $0 \preceq 1$ (так як $0 \vee 0 = 0, 0 \vee 1 = 1$).

$(0, 0) \preceq (1, 1)$, звідки $0 \preceq 1$. (так як $0 \vee 0 = 0, 1 \vee 1 = 1$).

$(0, 1) \preceq (1, 1)$, звідки $1 \preceq 1$ (так як $0 \vee 1 = 1, 1 \vee 1 = 1$).

$(1, 0) \preceq (1, 1)$, звідки $1 \preceq 1$. (так як $1 \vee 0 = 1, 1 \vee 1 = 1$).»

Після заповнення всіх клітин та перевірки введених значень програмою таблиця матиме вигляд (табл. 3.2).

Таблиця 3.2 – Таблиця кроку 2 алгоритму після правильного заповнення.

<i>Назва властивості</i>	$\bar{x}$	$x \vee y$
Збереження 0	☐	☐
Збереження 1	☐	☐
Лінійність	☐	☐
Самодвоїстість	☐	☐
Монотонність	☐	☐

Необхідно передбачити, щоб таблиця 2 на наступному кроці не зникла.

Перехід на крок 3.

Крок 3. На екрані завдання: *Вибрати одну правильну відповідь.*

Система булевих функцій $\{\bar{x}, x \vee y\}$ є:

- функціонально повною, так як в таблиці у кожному рядку є хоча б один «+»
- не функціонально повною, так як в стовпцях таблиці є «+»
- функціонально повною, так як в таблиці у кожному рядку є хоча б один «-»
- не функціонально повною, так як в таблиці у кожному стовпці є хоча б один «-»

Повідомлення в разі помилкового вибору: «Система булевих функцій $\{\bar{x}, x \vee y\}$ є функціонально повною, так як в таблиці у кожному рядку є хоча б один «-», тобто для кожного класу є хоча б одна функція, яка цьому класу не належить. За теоремою Поста така система функцій є функціонально повною.».

Алгоритм завершено.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Блок-схема алгоритму роботи тренажера

Блок-схема візуалізує алгоритм функціонування програми. Це полегшує процес створення та розуміння коду для розробників та інших учасників проекту. Дана графічна модель може бути важливою складовою документації програмного проекту, де вона зберігає інформацію про логіку програми.

Блок-схема алгоритму реалізації тренажеру «Замкнені класи. Функціональна повнота булевих функцій» представлена на рис.4.1

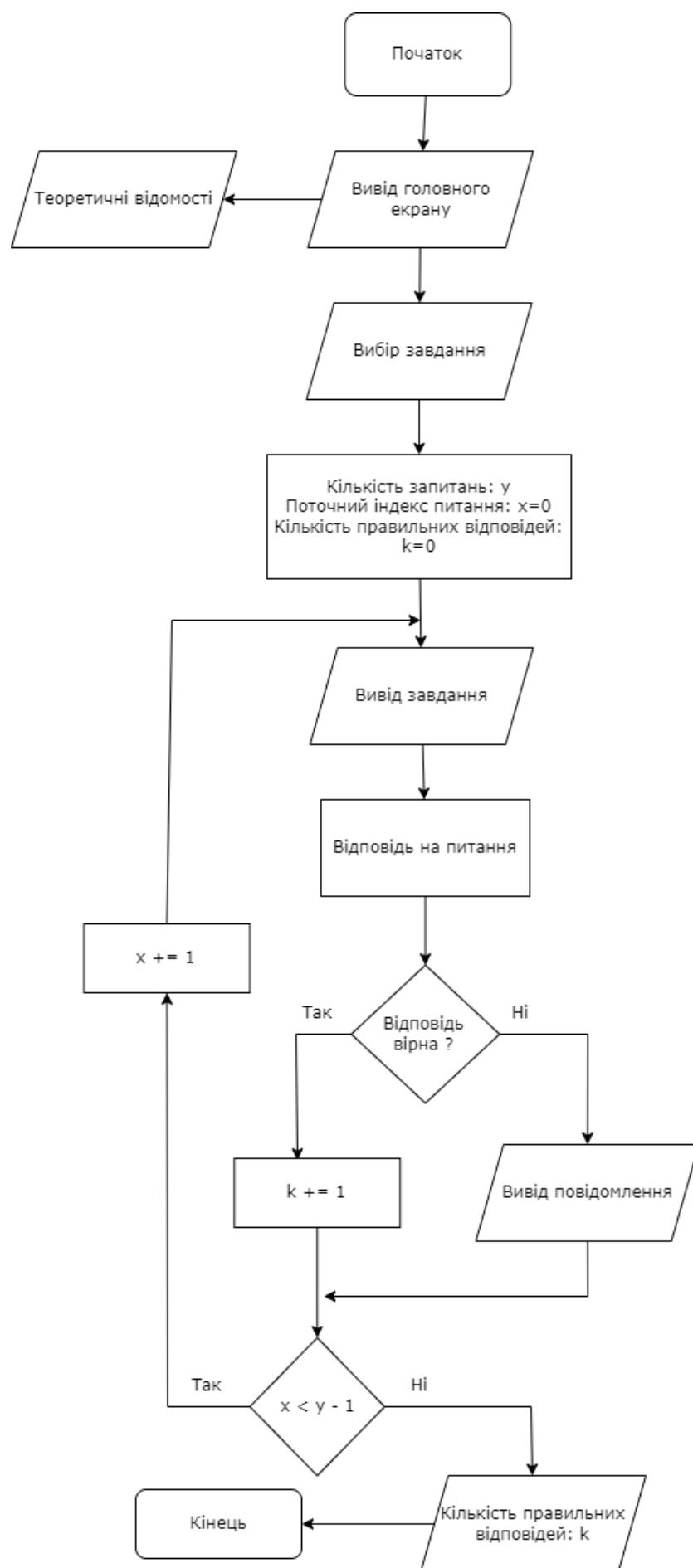


Рисунок 4.1 – Блок-схема алгоритму роботи тренажера

4.2. Опис проектних рішень, інструментів та підходів до розробки

Для розробки тренажера було обрано:

- мову програмування C#;
- мову розмітки XAML;
- технологію створення графічного інтерфейсу користувача WPF;
- середовище розробки Microsoft Visual Studio.

C# – це універсальна об'єктно-орієнтована мова програмування високого рівня, яка працює на платформах з відкритим вихідним кодом, а також на архітектурі Microsoft Windows .NET. (рис. 4.2). Вона дозволяє розробникам використовувати широкий набір .NET бібліотек та інструментів для створення високоякісних додатків [4].



Рисунок 4.2 – Мова програмування «C#»

C# це потужна та гнучка мова програмування, яку щодня використовують багато розробників. Популярність C# значно зростає, наразі вона вважається шостою за популярністю мовою програмування після Python, Java, C, C++ та JavaScript. Мова постійно розвивається завдяки великій спільноті, яка використовує її вже майже 20 років [5].

Як об'єктно-орієнтована мова програмування, C# дотримується принципів інкапсуляції, успадкування та поліморфізму. З роками C# значно розвинулася, а регулярні оновлення додають сучасні функції, такі як асинхронне програмування

та інтегровану мову запитів (LINQ), що гарантує її актуальність у швидкоплинному технологічному середовищі.

Мова програмування C# підходить для розробки:

- десктопних додатків;
- веб-додатків;
- хмарних API;
- корпоративних крос-платформених додатків;
- мобільної розробки;
- розробки ігор.

Основні переваги використання мови C#:

- Простота у вивченні та висока продуктивність. C# мова високого рівня з простим синтаксисом, що робить її доступною для програмістів усіх рівнів кваліфікації.
- Крос-платформна підтримка. Можливість створення додатків для Windows, Linux, macOS та ін..
- ООП. C# є об'єктно-орієнтованою мовою програмування, що дозволяє розробникам організовувати код у багаторазові компоненти, що призводить до більш структурованого коду.
- Продуктивність. Має автоматичне керування виділенням та звільненням пам'яті, що підвищує продуктивність і зменшує ймовірність помилок.
- Безпека. Надає вбудовані функції безпеки, які допомагають захистити додаток від хакерських атак.
- Сильно типізована. Суворі перевірки типів виконуються під час компіляції, що підвищує надійність коду.
- Висока масштабованість. C# дозволяє масштабувати та змінювати програму завдяки особливостям статичного кодування.

XAML – це декларативна мова розмітки, яка використовується переважно для створення користувацьких інтерфейсів у додатках, розроблених на платформі Microsoft .NET (рис.4.3).



```

<Window x:Class="ContentPropertySample.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="400" Width="500" >
    <Grid x:Name="ParentGrid">
        <Button Height="50" Margin="10,10,10,10" Content="Hello XAML" />
        <Button Height="80" Margin="10,10,10,10" />
        <Rectangle Height="20" Fill="Green"></Rectangle>
    </Button>
    <Button Margin="10,10,10,10">
        <StackPanel Orientation="Vertical">
            <Ellipse Height="60" Width="60" Fill="Red"/>
            <TextBlock TextAlignment="Center"><Run Text=" Red Circle"/></TextBlock>
            <TextBlock TextAlignment="Center"><Run Text=" " /></TextBlock>
            <Rectangle Height="20" Width="60" Fill="Green"></Rectangle>
            <TextBlock TextAlignment="Center"><Run Text=" Green " /></TextBlock>
        </StackPanel>
    </Button>
    </Grid>
</Window>

```

Рисунок 4.3 – Мова розмітки «XAML»

XAML пов'язаний з такими технологіями, як WPF (Windows Presentation Foundation), UWP (Universal Windows Platform) та Xamarin Forms.

XAML сприяє розділенню інтерфейсу користувача (UI) та логіки програми. Інтерфейс визначається в XAML, в той час як логіка може бути реалізована в окремому файлі коду, що сприяє розвитку архітектури Model-View-ViewModel (MVVM). Він надає різноманітні елементи керування макетами, такі як Grid, StackPanel та Canvas, які допомагають розробникам організовувати та позиціонувати елементи інтерфейсу.

Основні варіанти використання XAML [6]:

- Розробка десктопних додатків. Використовується при розробці десктопних додатків з використанням WPF або UWP для визначення користувацького інтерфейсу, макета, елементів керування.
- Розробка мобільних додатків. Xamarin Forms використовує XAML для створення кросплатформних мобільних додатків для iOS, Android та Windows, використовуючи спільну кодову базу.
- Інтерактивна візуалізація даних (діаграми та графіки). XAML підходить для створення інтерактивних діаграм та графіків. Розробники можуть визначати макет і зовнішній вигляд візуалізацій даних у XAML, а прив'язка даних може використовуватися для динамічного оновлення візуалізацій на основі зміни даних.

- Розробка анімацій. XAML підтримує визначення анімацій та переходів, дозволяючи розробникам створювати візуально привабливі та інтерактивні користувацькі інтерфейси з плавними переходами.
- Розробка ігрових інтерфейсів. XAML дозволяє розробникам проектувати та компонувати елементи інтерфейсу для ігрових меню.

WPF – це безкоштовна графічна підсистема з відкритим вихідним кодом, розроблена компанією Microsoft для візуалізації користувацьких інтерфейсів на базі Windows. Спочатку відома як «Avalon», WPF була випущена у 2006 році (рис. 4.4).



Рисунок 4.4 – Технологія для створення графічного інтерфейсу користувача «Wpf»

WPF побудований на основі багаторівневої архітектури, що складається з трьох основних шарів [7] (рис. 4.5):

- Managed Layer. Надає розробникам WPF API, який містить класи та бібліотеки для створення додатків.
- Presentation Core. Відповідає за основний рендеринг та обробку графіки, включаючи макет, малювання та рендеринг тексту.
- Milcore. Взаємодіє з операційною системою, забезпечуючи графікою та віконну функціональність.

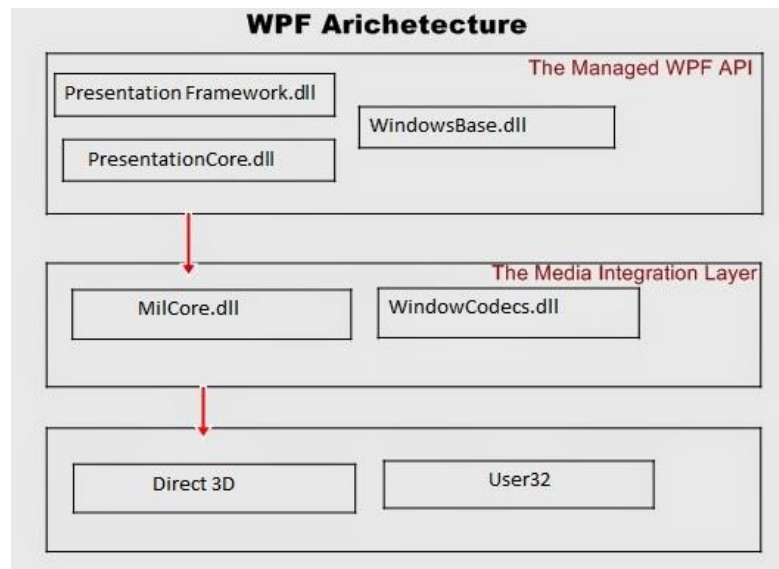


Рисунок 4.5 – Архітектура «WPF»

WPF надає систему стилів та шаблонів, яка дозволяє розробникам визначати візуальний вигляд та поведінку елементів інтерфейсу. Стили можна повторно використовувати в різних елементах, а шаблони дозволяють налаштовувати структури управління. Також є можливість створювати плавні анімації та переходи, покращуючи взаємодію з користувачем. Розробники можуть визначати анімацію за допомогою XAML, щоб додати динамічні ефекти до елементів інтерфейсу.

Переваги WPF [7]:

- Підтримка інтегрованих середовищ. WPF підтримується популярними середовищами розробки, такими як Visual Studio, що надають надійний набір інструментів для проектування, розробки та налагодження WPF-додатків.
- Створення адаптивних інтерфейсів. WPF автоматично впорядковує та змінює розміри елементів інтерфейсу на основі їхніх властивостей та доступного простору.
- Локалізація. WPF підтримує локалізацію, що полегшує створення додатків, які можна адаптувати для різних мов та регіонів.

Microsoft Visual Studio – це інтегроване середовище розробки (IDE), яке дозволяє створювати, тестувати та налагоджувати програмне забезпечення (рис. 4.6).

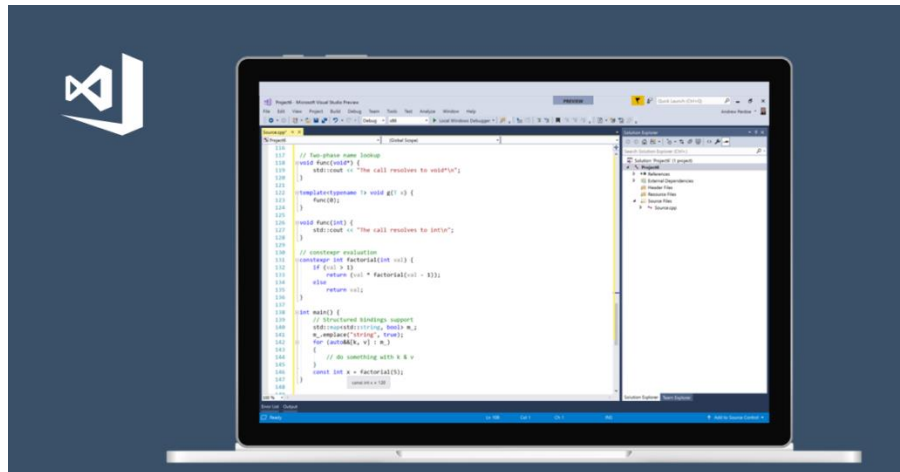


Рисунок 4.6 – Середовище розробки «Microsoft Visual Studio»

Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як: Windows API, Windows Forms, Windows Presentation Foundation, Windows Store та Microsoft Silverlight.

До складу Visual Studio також входить, конструктор форм для створення графічних додатків, конструктор для розробки дизайну, конструктор класів та конструктор схем баз даних [8].

Можливості та переваги Visual Studio:

- підтримує багато мов програмування C#, C++, JavaScript, Python та інші мовні мови;
- можна створювати надійні та масштабовані програми за допомогою різноманітних бібліотек та фреймворків, таких як Xamarin, Azure та .NET Framework;
- легко інтегрується з популярними системами контролю версій (Git), що дозволяє ефективно співпрацювати;
- пропонує безліч функцій для підвищення продуктивності, таких як IntelliSense, що дозволяє розробникам писати код більш ефективно;
- має активну спільноту, яка надає доступ до великої кількості ресурсів, документації, навчальних посібників;

– надає підтримку налагодження для всіх підтриманих мов програмування. Налагодження можна виконувати локально, віддалено та під час розробки.

4.3. Інструкція користувача

Коли користувач запускає застосунок, він потрапляє на головне вікно, де у нього є можливість переглянути тему тренажеру, автора та керівника даної роботи. Також у головному меню знаходиться декілька кнопок, що дають змогу користувачеві переглянути теоретичний матеріал або розпочати тестування (рис. 4.7).

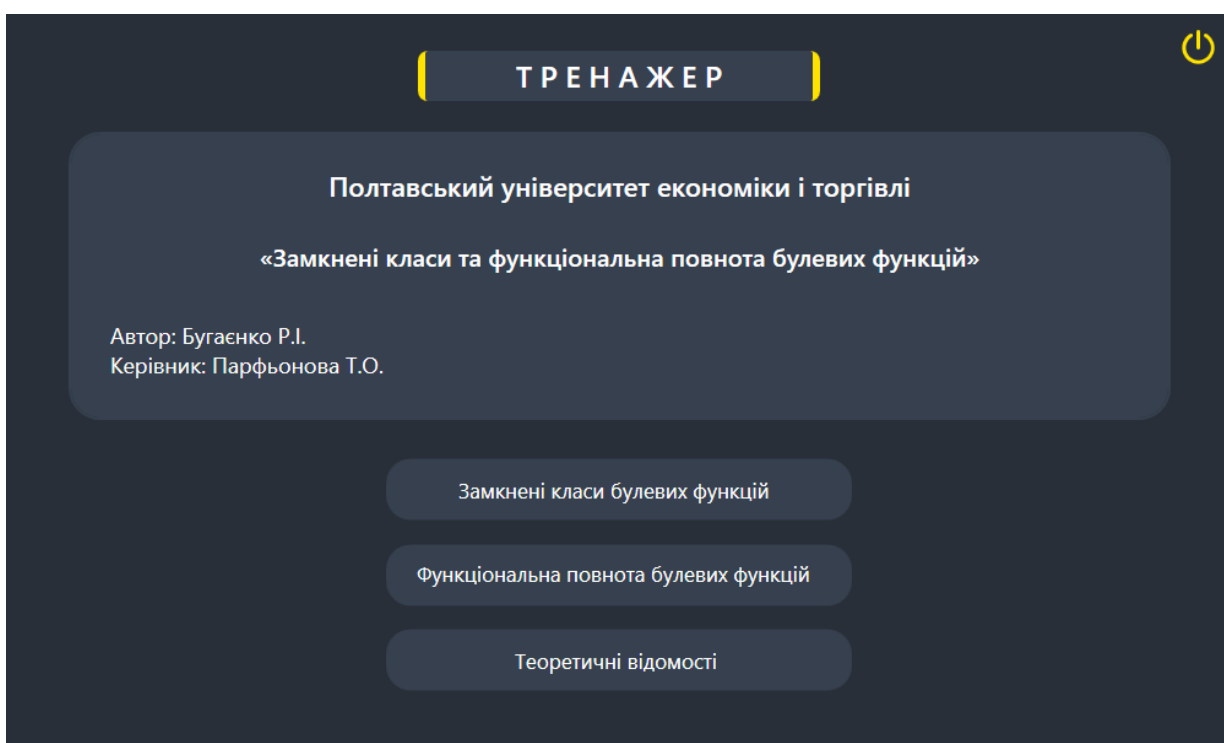


Рисунок 4.7 – Головне вікно застосунку

Щоб перейти до теоретичних відомостей, користувачу потрібно натиснути на відповідну кнопку, після чого його буде перенаправлено до вікна, де можна буде переглянути документ з теорією до теми тренажеру (рис.4.8).

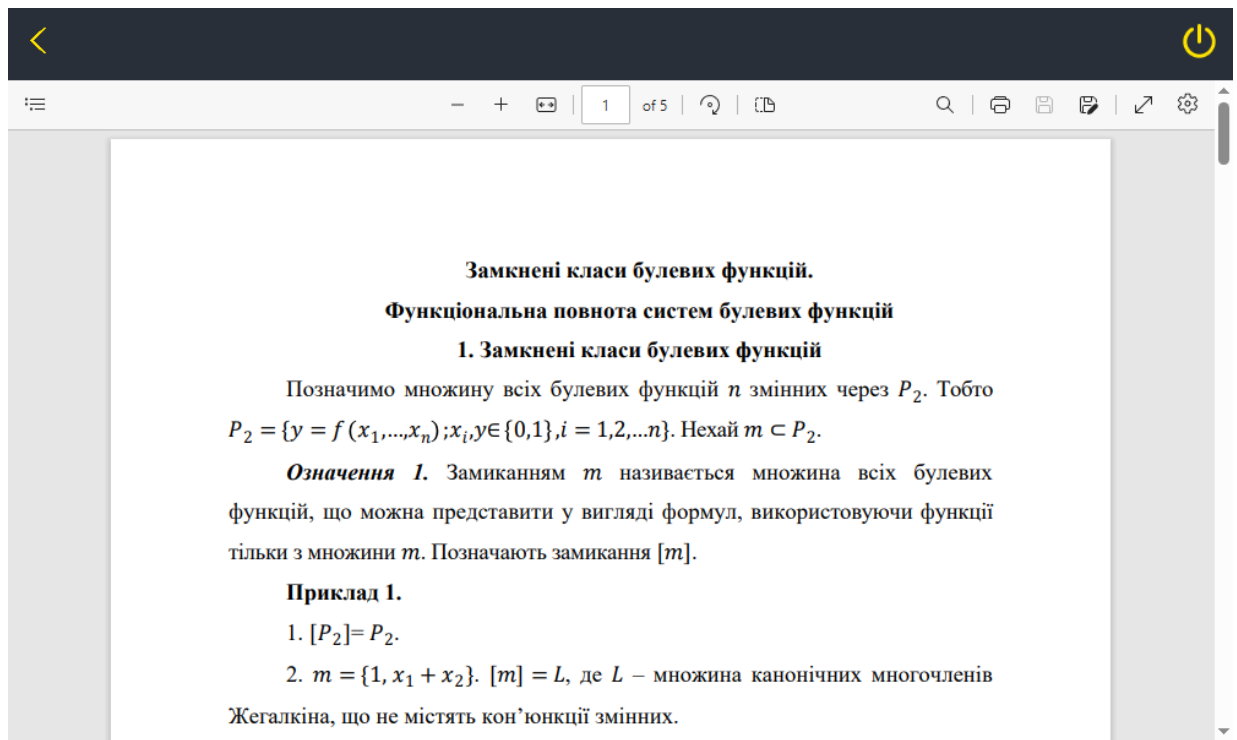


Рисунок 4.8 – Вікно теоретичних відомостей

Після переходу безпосередньо до тестування користувача перенаправляє до відповідного вікна на якому відображено: номер питання, текст питання та варіанти відповідей. Для того, щоб підтвердити власну відповідь, користувачу потрібно натиснути кнопку «Відповісти» (рис. 4.9).

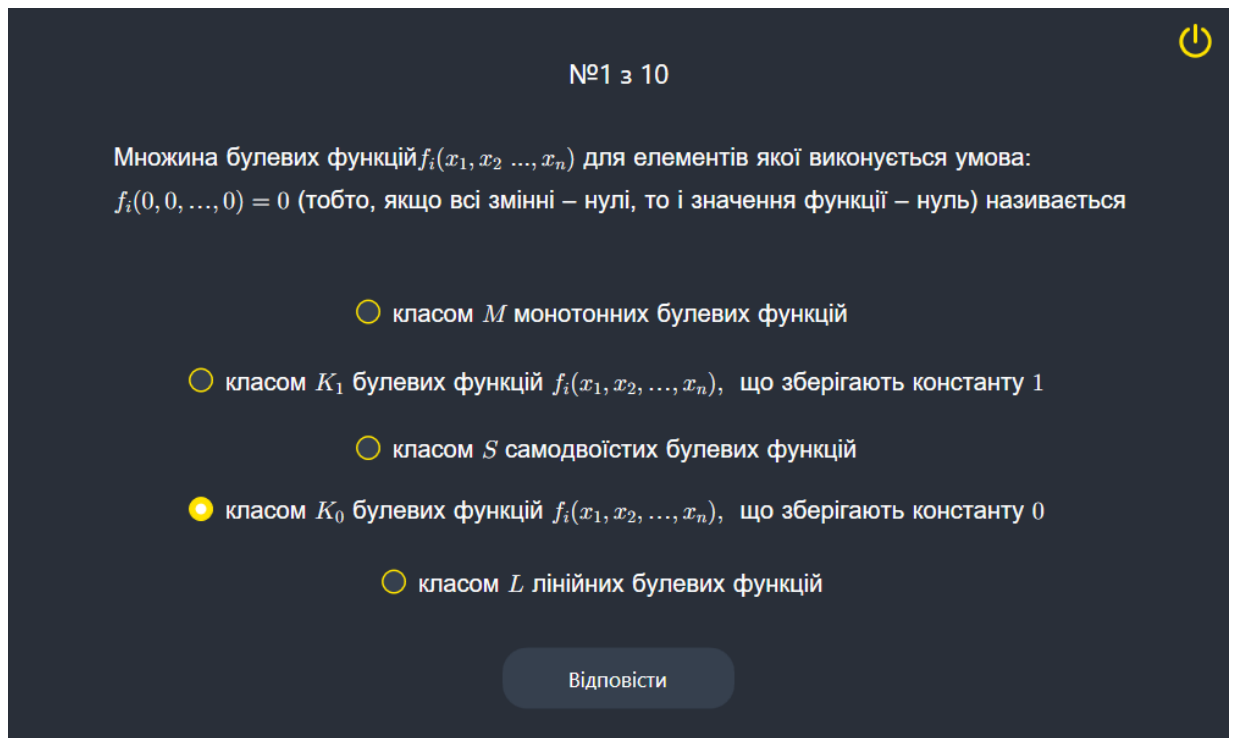


Рисунок 4.9 – Вікно тестування

У разі неправильної відповіді перед користувачем з'явиться модальне вікно з відповідним повідомленням та текстом правильної відповіді (рис. 4.10). Після закриття цього вікна користувач перейде на наступний крок тесту.

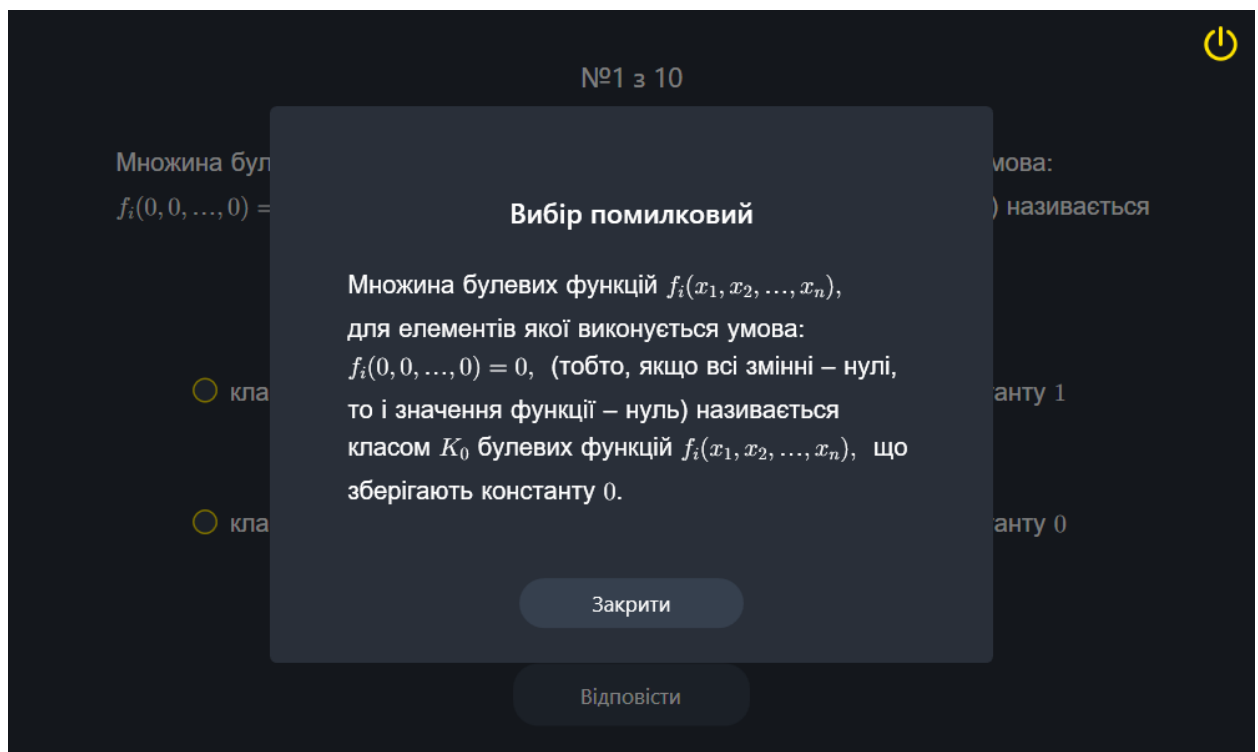


Рисунок 4.10 – Модальне вікно неправильної відповіді

Коли користувач відповість на останнє питання, його автоматично перенаправить до вікна результатів, де він зможе переглянути кількість правильних відповідей та повернутися до головного меню за допомогою кнопки «На головну» (рис. 4.11).

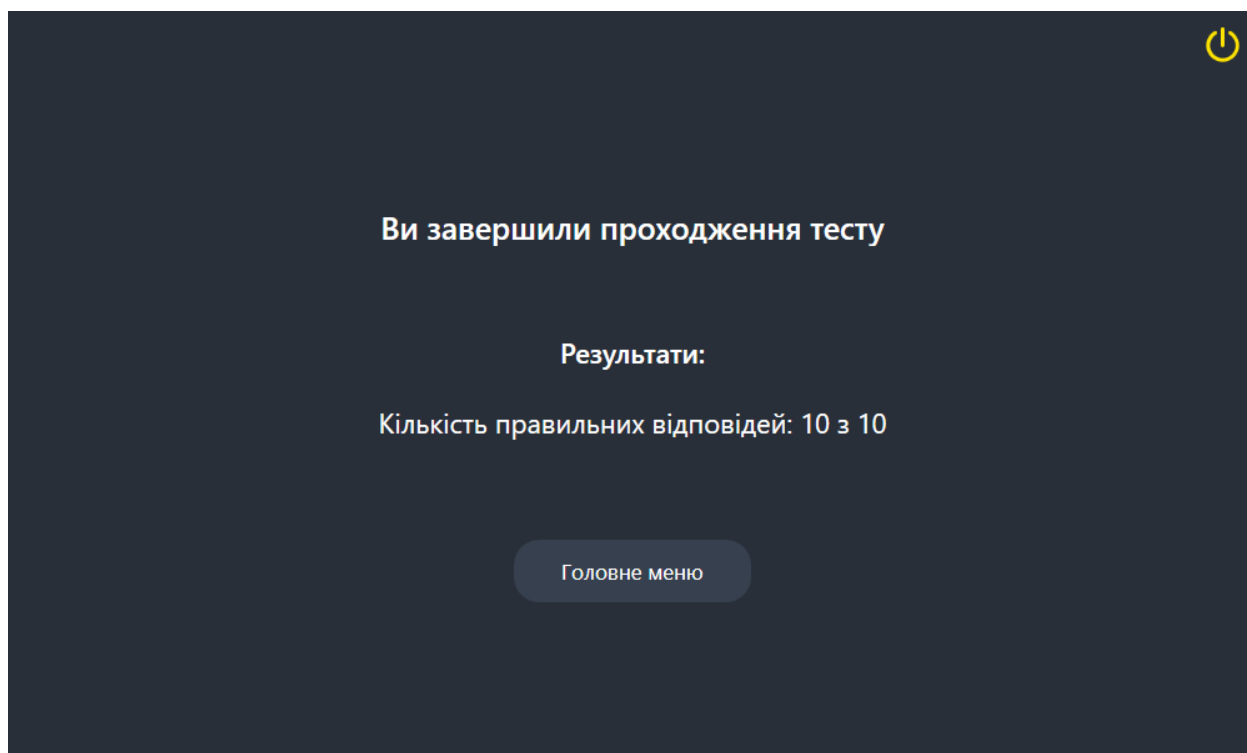


Рисунок 4.11 – Вікно результатів

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи успішно розроблено алгоритм навчального тренувального симулятора на тему «Замкнені класи. Функціональна повнота булевих функцій» для дистанційного курсу «Дискретна математика». Для цього було виконано ряд завдань, зокрема, здійснено огляд існуючих подібних програм, опрацьовано необхідний теоретичний матеріал теми, обрано програмне забезпечення для реалізації алгоритму, побудовано власне алгоритм та здійснена його програмна реалізація.

Завершена програмна реалізація навчального тренувального симулятора дозволяє створити інтерактивне середовище для вивчення замкнених класів та функціональної повноти булевих функцій. Це сприяє покращенню якості навчання та кращому розумінню теми, надає можливість студентам активно взаємодіяти з матеріалом та задачами, що неодмінно сприяє підвищенню рівня засвоєння матеріалу та покращенню навичок у розв'язанні практичних задач дискретної математики.

Результати роботи можуть бути впроваджені в навчальний процес та апробовані на II Всеукраїнській науково-практичній інтернет-конференції «Сучасні тенденції підготовки майбутніх фахівців у закладах професійної (професійно-технічної), фахової передвищої та вищої освіти» (Полтавський державний аграрний університет).

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Гаркуша С. В. Методичні рекомендації щодо виконання кваліфікаційної роботи студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра / С. В. Гаркуша, О. В. Ольховська, О. О. Черненко. – Полтава: ПУЕТ, 2023. – 68 с.
2. Ємець О. О. Дискретна математика: навчальний посібник для самостійного вивчення навчальної дисципліни студентами денної форми навчання спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра / О. О. Ємець, Т. О. Парфьонова. – Вид 3-тє, допов. і перероб. – Полтава: ПУЕТ, 2023. – 282 с. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/12869>
3. Ліхоузова Т. А. Дискретна математика. Практикум [Електронний ресурс]: навч. посібник для студ. спеціальностей 121 «Інженерія програмного забезпечення», 126 «Інформаційні системи та технології» / Т. А. Ліхоузова. – Київ: КПІ ім. Ігоря Сікорського, 2020. – 62 с. – Режим доступу: [file:///C:/Users/owner/Downloads/DM_praktykum%20\(4\).pdf](file:///C:/Users/owner/Downloads/DM_praktykum%20(4).pdf)
4. Introduction to C#: definition and utilities [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.mytaskpanel.com/introduction-to-csharp/> – Назва з екрану.
5. Why learn C# in 2024? An expert's answer [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.telerikacademy.com/blog/details/why-learn-csharp-in-2023-an-experts-answer> – Назва з екрану.
6. What is XAML and use cases of XAML? [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.devopsschool.com/blog/what-is-xaml-and-use-cases-of-xaml/> – Назва з екрану.

7. What is WPF and use cases of WPF? [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.devopsschool.com/blog/what-is-wpf-and-use-cases-of-wpf/> – Назва з екрану.

8. Visual Studio – Everything you need to know [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://licenseq.com/visual-studio-everything-you-need-to-know/> – Назва з екрану.

ДОДАТОК А. КОД ПРОГРАМИ