

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

« _____ » _____ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

**на тему
АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА З ТЕМИ
«АЛГЕБРА ЖЕГАЛКІНА, СПОСОБИ ПОБУДОВИ ПОЛНОМІВ
ЖЕГАЛКІНА» ДИСТАНЦІЙНОГО КУРСУ «ДИСКРЕТНА
МАТЕМАТИКА»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Бурко Андрій Олександрович

_____ « ____ » _____ 2021р.
(підпис)

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

_____ « ____ » _____ 2021р.
(підпис)

ПОЛТАВА 2021 р.

РЕФЕРАТ

Записка: 53 с., 17 рис., 1 додаток, 14 джерел.

Предмет розробки – програма для елементів тренажеру для навчання темі «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Мета роботи – створення тренажера для навчання студентів дистанційного навчального курсу «Дискретна математика» темі «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Методи, які були використані для розв’язування задачі – платформа Unity та середовище розробки Visual Studio.

Розроблено алгоритм роботи тренажера.

Інформація в тренажері видається наступним чином: спочатку виводиться коротка довідка, потім практичне завдання.

При правильній відповіді користувач отримує повідомлення «Правильна відповідь», у іншому випадку користувач отримує повідомлення «Неправильна відповідь» та правильну відповідь.

Результати роботи впроваджені в дистанційному навчальному курсі «Дискретна математика».

Ключові слова: ТРЕНАЖЕР, АЛГЕБРА ЖЕГАЛКІНА, ПОБУДОВА ПОЛІНОМІВ ЖЕГАЛКІНА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	4
ВСТУП	5
1. ПОСТАНОВКА ЗАДАЧІ.....	7
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	9
2.1. Огляд робіт, де розглянуте аналогічне до теми роботи завдання.....	9
2.2. Позитивні аспекти та вади оглянутих робіт.....	13
2.3. Необхідність та актуальність теми роботи.....	13
3. ТЕОРЕТИЧНА ЧАСТИНА	16
3.1. Огляд матеріалу за темою роботи	16
3.2. Алгоритмізація задачі за темою роботи	20
3.3. Розробка блок-схеми, яка підлягає програмуванню	31
4. ПРАКТИЧНА ЧАСТИНА	33
4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи.....	33
4.2. Опис процесу програмної реалізації	37
4.3. Опис програми, її інструкція	42
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А. КОД ПРОГРАМИ	52

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

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
\wedge	бінарна операція кон'юнкції
\oplus	бінарна операція суми по модулю два
Константа нуль	$1 \oplus 1 = 0$
Операція заперечення	$\neg x = x \oplus 1$
Операція диз'юнкції	$x \vee y = x \wedge y \oplus x \oplus y$

ВСТУП

На сьогоднішній день практично кожна особа на світі користується новітніми технологіями. Люди стали гостро залежити від мережевого спілкування та інтернет-ресурсів. Мережа інтернет дозволяє не тільки спілкуватися або швидко знаходити інформацію. На просторах інтернету не важко знайти терабайти навчальних матеріалів серед яких є і предмет роботи даної роботи, а саме програма для навчання.

Метою роботи є створення тренажера для навчання студентів дистанційного навчального курсу «Дискретна математика» темі «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Об'єктом розробки в даній роботі є створення тренажеру для систем дистанційного навчання.

Предметом розробки є програма для тренажеру для навчання темі «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Для створення програми було використано платформу Unity та середовище розробки Visual Studio.

Перелік використаних методів полягає у застосуванні алгебри Жегалкіна, поліномів Жегалкіна.

Тренажер готовий до використання в дистанційному курсі «Дискретна математика».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку основних вимог до змісту і постановку задачі для реалізації програми-тренажера. У другому розділі описано:

- Огляд робіт, де розглянуте аналогічне до теми роботи завдання.
- Позитивні аспекти та вади оглянутих робіт.
- Необхідність та актуальність теми роботи

У третьому розділі розглянуто:

- Матеріал за темою роботи.

- Алгоритмізацію задачі за темою роботи.
- Розробку блок-схеми.

У четвертому розділі описано:

- Обґрунтування вибору програмних засобів.
- Процес програмної реалізації.
- Роботу програми.

Обсяг пояснювальної записки: 53 стор., в т.ч. основна частина - 41 стор, джерела - 14 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Головною задачею є програмна реалізація тренажера.

Головним завданням програми є навчання студентів темі «Дискретна математика» з теми «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Дане програмне забезпечення буде створено у середовищі розробки MS Visual Studio, з використанням мови програмування C#.

З метою виконання роботи необхідно:

- знати основні закони алгебри Жегалкіна;
- вміти доводити монотонність, лінійність і самодвоїстість булевих функцій;
- вміти будувати поліном Жегалкіна способом невизначених коефіцієнтів;
- вміти будувати поліном Жегалкіна за допомогою трикутника Паскаля.

Основними завданнями роботи є:

1. Створення програми для навчання обраній темі;
2. Створення пояснювальної записки з описом програмного забезпечення;
3. Розробка алгоритму роботи програми;
4. Розробка блок-схем алгоритму роботи програми;
5. Тестування створеної програми.

Основними вимогами до програмного забезпечення є:

1. Зручний та зрозумілий графічний дизайн;
2. Покрокова подача теоретичної інформації;
3. Перевірка на правильність введеної відповіді та виведення відповідного повідомлення після вводу.

У зв'язку з цим потрібно реалізувати такі функції:

1. Відкрити теоретичний матеріал по темі;

2. Перейти до виконання завдань по темі;
3. Повернутися на головне меню;
4. Вихід з тренажера в будь-який момент.

Розглянемо основні задачі роботи по кожному розділу:

- Інформаційна частина
 - зробити огляд робіт за схожою тематикою;
 - описати позитивні та негативні аспекти робіт;
 - вказати актуальність роботи;
- Теоретична частина
 - розробити алгоритм задачі;
 - створити блок-схему програми;
- Практична частина
 - обґрунтувати вибір програмних засобів;
 - описати процес програмної реалізації;
 - описати роботу програми.

Пропонуються наступні завдання для їх вирішення.

1. Побудувати поліном Жегалкіна для $g = (x \vee y) z$.
2. Метод трикутника Паскаля. Побудувати поліном Жегалкіна для функції f . Візьмемо функцію $f(x_1, x_2, x_3) = x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$.
3. Метод невизначених коефіцієнтів. Знайдемо поліном Жегалкіна для функції $f(x_1, x_2, x_3) = (x_1 x_2 \vee x_3) \rightarrow \bar{x}_2$, використовуючи метод невизначених коефіцієнтів.
4. Перетворити $\varphi = x \vee y \vee \bar{z}$ до формули алгебри Жегалкіна.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд робіт, де розглянуте аналогічне до теми роботи завдання

Самостійна робота студентів будь-яких форм навчання потребує наявності засобів, що полегшують вивчення матеріалу. Одним із інструментів при організації самостійної роботи є комп'ютерні тренажери, які все частіше застосовують в дистанційному навчанні. Тренажери призначені для вивчення і закріплення різноманітних практичних навичок.

При виконанні роботи були розглянуті деякі існуючі в ПУЕТ тренажери з різних дисциплін. Першим розглядався тренажер для дистанційного курсу «Елементи комбінаторної оптимізації», написаний мовою програмування Java у вигляді Java-аплету [2].

Під час проходження тренажера потрібно відповідати на поставленні питання, а для деяких питань також потрібно розрахувати результат та ввести його у відповідне поле. Елементи комбінаторних конфігурацій генеруються динамічно.

Тренажери, розроблені в ПУЕТ для дистанційного навчального курсу кафедри ММСІ «Методи оптимізації та дослідження операцій»:

«Графічний метод розв'язування задач лінійного програмування» – розроблений у вигляді Java-аплету в середовищі програмування NetBeans IDE [3] (рис. 2.1).

Реалізований у вигляді тестів. Для користувачів є можливість переходити між питаннями самостійно.

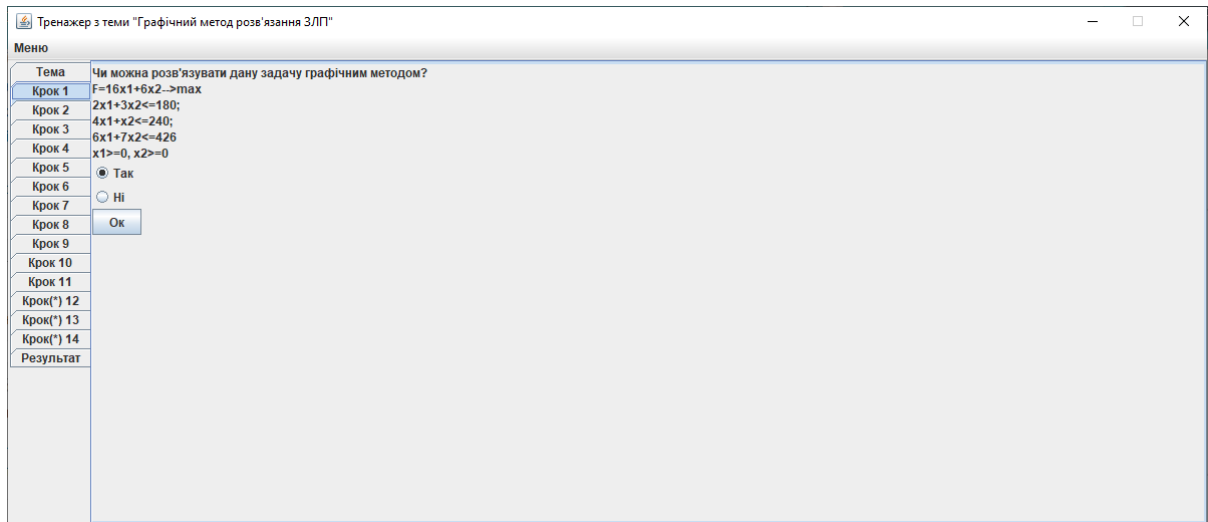


Рисунок 2.1 – Графічний метод розв’язування задач лінійного програмування

«М-метод для ЗЛП» – розроблений у вигляді Java-аплету в середовищі програмування NetBeans IDE, дозволяє вивчити М-метод для задач лінійного програмування (рис. 2.2) [4].

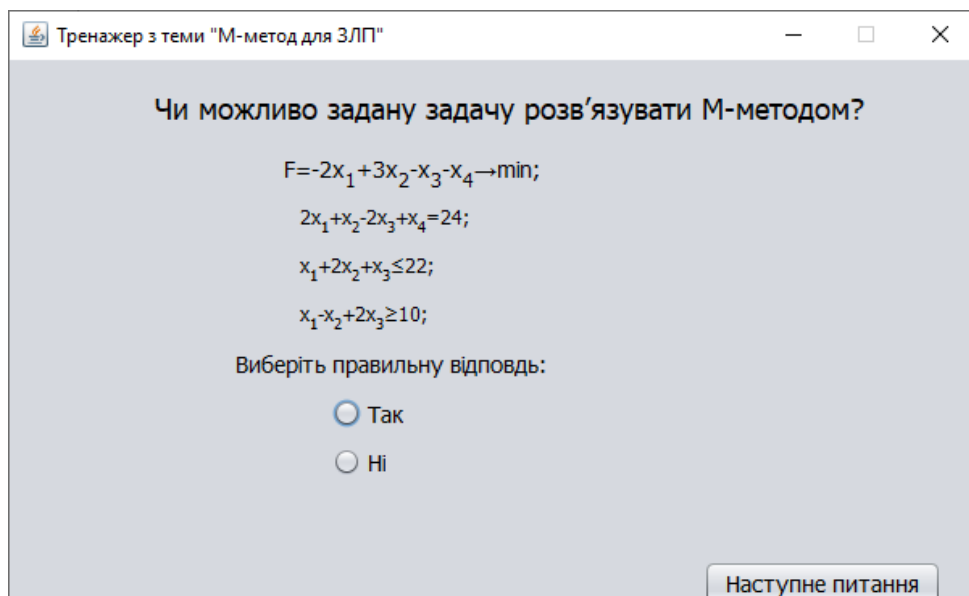


Рисунок 2.2 – М-метод для ЗЛП

Процес роботи з програмою побудовано у вигляді діалогу. Перед кожним кроком користувачеві задається тестове запитання для перевірки його знання. На кожному кроці детально описуються дії, які потрібно виконати, за необхідності відображаються результати обчислень попередніх

кроків. При введенні даних з'являються підказки, які пояснюють, що вводиться в те чи інше поле.

«Розв'язування ЗЛП модифікованим симплекс-методом» – розроблений мовою програмування Java в середовищі розробки NetBeans IDE, він дає можливість засвоїти дану тему (рис. 2.3) [5].

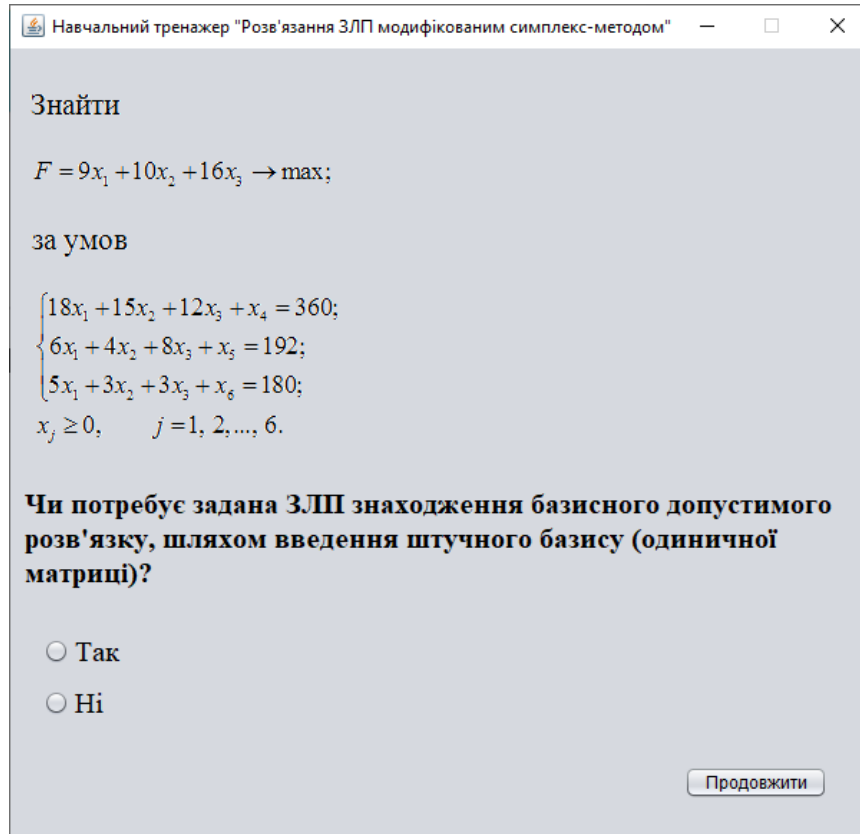


Рисунок 2.3 – Двоїстість в лінійному програмуванні

Розпочавши проходження тренажера на кожному кроці перед користувачем виводиться запитання та варіанти відповідей, також тренажер містить завдання з самостійним розв'язанням задачі.

Наступний дистанційний курс – «Теорія програмування»:

«Синтаксичний аналіз» – розроблений мовою програмування Java. В даному тренажері виводиться умова і потрібно поступово заповнити відповідні таблиці (рис. 2.4) [6].

Дано граматику
 1. $S \rightarrow F$, 2. $S \rightarrow (S + F)$, 3. $F \rightarrow a$
 та вхідний рядок: (a + a)

Тренажер розробила:
 Корсун Ольга, група І-411
 керівник к.ф.-м.н. Черненко О.О.

Який вигляд має множина FIRST(S)

{(a }
 {(a,+ }
 {(a,+ }
 {(a,+,\$ }

Відповісти

Множини FIRST і FOLLOW

	FIRST	FOLLOW
S		
F		

Таблиця аналізу

M	()	a	+	'\$'
S					
F					

Стек: [S,\$]
 Поточний вигляд рядка: (a + a)
 Результат:

Рисунок 2.4 – Синтаксичний аналіз

«Верифікація програм та алгоритмічно нерозв'язні проблеми» – розроблений мовою програмування Java. На відміну від решти тренажерів спочатку надається можливість вибору одного із завдань (рис. 2.5) [7].

Тренажер

Тренажер з теми

**«ВЕРИФІКАЦІЯ ПРОГРАМ ТА
 АЛГОРИТМІЧНО НЕРОЗВ'ЯЗНІ
 ПРОБЛЕМИ»**

дистанційного навчального курсу
 «Теорія програмування»

Завдання 1
 Завдання 2
 Завдання 3

Розпочати тренінг

Розробила: Бернацька Вікторія, керівник: Черненко О. О.

Рисунок 2.5 – Верифікація програм та алгоритмічно нерозв'язні проблеми

2.2. Позитивні аспекти та вади оглянутих робіт

Загальні основні вади оглянутих робіт:

- відсутня інформація про тренажер, яка тема розглядається;
- відсутній теоретичний матеріал, що пояснює деякі моменти в питаннях;
- на деяких кроках тренажера не зрозуміло що вимагають від користувача;
- елементи інтерфейсу можуть перекривати текст чи інший елемент.

Позитивні аспекти при реалізації тренажера:

- відразу зрозуміло тему тренажера;
- зручний інтерфейс програми при використанні;
- надається можливість переглянути матеріал з теми перед виконанням;
- доступ до довідки в будь-який момент;
- перехід між кроками тренажера;
- виведення відповідного повідомлення при помилці;
- вказується де саме було допущено помилку;
- поступово відбувається закріплення теми;
- присутнє відображення результату роботи тренажера.

2.3. Необхідність та актуальність теми роботи

Дистанційне навчання у ВНЗ не є різновидом або вдосконаленим варіантом заочного, оскільки це нова, самостійна, прогресивна форма навчання, яка володіє більшими потенційними можливостями. Сфера можливого застосування дистанційного навчання у ВНЗ досить широка: від

цілісної підготовки фахівців до окремих курсів та фрагментів дидактичного забезпечення під час різних видів занять. На основі аналізу світових тенденцій розвитку дистанційної освіти вважаємо за доцільне виокремити такі тенденції розвитку дистанційної освіти в Україні: інформаційно-технологічну, нормативно-правову, фінансово-економічну, інституційну, науково-методичну та організаційно - педагогічну.

Інформаційно-технологічна тенденція розвитку дистанційної освіти в Україні тісно пов'язана з інформатизацією, яка на сучасному історичному етапі має стратегічне значення для нашої країни. У наш час виробництво інформаційного продукту через його високу товарну вартість є важливим чинником економічного розвитку країни. Інформаційні технології, проникаючи в усі галузі діяльності людини, змінюють характер праці як у виробничій, так і у невиробничій сферах, впливають на структуру національної економіки, підвищують рівень інформування широких верств населення й таким чином сприяють демократизації суспільства. Ці тенденції, що в усьому світі визначаються як процес інформатизації, позначаються на житті суспільства. Порівняно з традиційними індустріальними методами їх застосування дає можливість забезпечити підвищення рівня матеріалізації інтелектуальної праці і якості виробів. Саме через це найбільш розвинуті країни світу отримують на міжнародних ринках значні переваги.

На відміну від зарубіжних моделей, українська дистанційна освіта більш наближена до нашого споживача і є більш демократична. Органічно поєднуючи в собі змішані технології відкритої освіти (кейс-технології, TV-технології, мережеві технології), українська дистанційна освіта стає найбільш доступна широким масам населення, роблячи можливим здобувати освіту не на все життя, а все життя.

Сучасне інформаційне суспільство висуває вимоги до системи освіти, основні з яких можна сформулювати так: вміння самостійно знаходити, накопичувати і переосмислювати наукові знання; вміння студентів самостійно орієнтуватися в сучасному інформаційному суспільстві [8].

Дистанційна освіта розвивається дуже швидко, і для України є перспективною формою вищої освіти. На Заході ця форма з'явилася вже досить давно і має велику популярність серед студентів через її економічні показники і навчальну ефективність. Дистанційну форму навчання ще називають «освітою на протязі всього життя» через те, що більшість тих, хто навчається – дорослі люди. Багато хто з них вже має вищу освіту, проте через необхідність підвищення кваліфікації або розширення сфери діяльності у багатьох виникає потреба швидко і якісно засвоїти нові знання і набути навички роботи. Саме тоді оптимальною формою може стати дистанційне навчання.

Відкрита освіта дає широке поле для наукових досліджень, що сприяють розвитку творчих ініціатив розробників і педагогів, переходячи з об'єкта вивчення в об'єкт прогнозування, конструювання та проектування [9].

Впровадження у навчальний процес віртуально-навчальних середовищ і надання студентам та викладачам необхідних методичних рекомендацій щодо використання віртуально-навчальних середовищ у фаховій підготовці забезпечить просування за критеріями та рівнями якості застосування технології дистанційного навчання у фаховій підготовці надасть змогу підвищити якість фахової підготовки.

Таким чином, на сьогоднішній день, виникає потреба розробки і запровадження у навчальний процес програм дистанційного навчання, що відповідають кращим світовим зразкам і забезпечують підготовку фахівців на високому професійному рівні. Використання мережі Internet дає можливість оперативного доступу до інформаційних ресурсів навчального закладу та можливість ефективної взаємодії «викладач-студент», як в on-line, так і в off-line режимах.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд матеріалу за темою роботи

Алгебра Жегалкіна - безліч булевих функцій, на яких визначені нульарна операція взяття одиниці 1, бінарна операція кон'юнкції \wedge , і бінарна операція суми по модулю два \oplus . Константа нуль вводиться як $1 \oplus 1 = 0$.

Операція заперечення вводиться співвідношенням $\neg x = x \oplus 1$. Операція диз'юнкції вводиться співвідношенням $x \vee y = x \wedge y \oplus x \oplus y$.

За допомогою алгебри Жегалкіна будь-яку диз'юнктивну нормальну форму можна перетворити на поліном Жегалкіна.

В зарубіжній літературі представлення полінома Жегалкіна зазвичай називається алгебраїчною нормальною формою (АНФ). Якщо у кожному члені поліному Жегалкіна кожна змінна входить один раз та поліном не містить однакових членів, то такий поліном Жегалкіна називається канонічним.

Теорема Жегалкіна — стверджує існування і унікальність будь-якої булевої функції у вигляді поліному Жегалкіна [10].

Основні властивості алгебри Жегалкіна:

Властивості операції кон'юнкції.

$(x_1 \wedge x_2) \wedge x_3 = x_1 \wedge (x_2 \wedge x_3)$ - асоціативність;

$x_1 \wedge x_2 = x_2 \wedge x_1$ - комутативність;

$x_1 \wedge 0 = 0$; $x_1 \wedge 1 = x_1$ - дії з константами;

$x \wedge x = x$ - ідемпотентність.

Властивості операції сума за модулем 2.

$x_1 \oplus x_2 = x_2 \oplus x_1$ - комутативність;

$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3$ - асоціативність;

$x \oplus x = 0$ - закон зведення подібних доданків;

$x_1 \oplus (x_2 \oplus x_3) = x_1 x_2 \oplus x_1 x_3$ - дистрибутивність кон'юнкції відносно

\oplus .

Решта операцій алгебри логіки виражаються через базис цієї алгебри в такий спосіб:

$$\bar{x} = 1 \oplus x;$$

$$x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1 x_2;$$

$$x_1 \rightarrow x_2 = 1 \oplus x_1 \oplus x_1 x_2;$$

$$x_1 \downarrow x_2 = 1 \oplus x_1 \oplus x_2 \oplus x_1 x_2;$$

$$x_1 \sim x_2 = 1 \oplus x_1 \oplus x_2;$$

$$x_1 \leftarrow x_2 = x_1 \oplus x_1 x_2;$$

$$x_1 | x_2 = 1 \oplus x_1 x_2 \quad [11].$$

Визначення. Поліномом Жегалкіна (поліномом по модулю 2) від n змінних $X_1, X_2 \dots X_n$ називається вираз виду

$$C_0 \oplus C_1 X_1 \oplus C_2 X_2 \oplus \dots \oplus C_n X_n \oplus C_{12} X_1 X_2 \oplus \dots \oplus C_{12 \dots n} X_1 X_2 \dots X_n,$$

де постійні C_k можуть набувати значень 0 чи 1.

Наприклад, $f = X \oplus YZ \oplus XYZ$ і $f_1 = 1 \oplus X \oplus Y \oplus Z$ - поліноми, причому друга є лінійною функцією.

Теорема. Кожна булева функція представляється у вигляді полінома Жегалкіна єдиним чином.

Наведемо основні методи побудови поліномів Жегалкіна від заданої функції [10].

1. Метод трикутника Паскаля.

Крок 1. Будуємо таблицю значень функції (рядки таблиці йдуть в порядку збільшення двоїстих кодів). Таблицю краще розташовувати в лівій частині сторінки.

Крок 2. Побудова трикутника. Для цього беремо вектор значень функції і виписуємо його навпроти першого рядка таблиці. Далі заповнюємо трикутник, складаючи попарно сусідні значення за модулем 2. Результат додавання записуємо нижче.

Крок 3. Побудова полінома Жегалкіна. Нас цікавить ліва сторона

трикутника (значення виділені жирним). Числа на лівій стороні (виділені жирним шрифтом) трикутника є коефіцієнтами поліному при монотонних кон'юнкціях, які відповідають наборам значень змінних. Тепер випишемо для наочності ці кон'юнкції. Кон'юнкції випишемо за двоїстими наборами в лівій частині таблиці за наступним принципом: якщо навпроти змінної x_i стоїть 1, то змінна входить в кон'юнкцію, в протилежному випадку – змінна відсутня в кон'юнкції. Набору (0,0,0) відповідає 1.

Якщо принцип отримання кон'юнкцій зрозумілий, то стовпчик з ним (краще) не виписувати, а одразу переходити о побудови поліному. Для побудови поліному необхідні кон'юнкції тільки з рядків з одиницями на лівій стороні трикутника. Це і є кон'юнкції, що входять до складу полінома Жегалкіна.

2. Метод невизначених коефіцієнтів.

Знайдемо поліном Жегалкіна для функції $f(x_1, x_2, x_3) = (x_1 x_2 \vee x_3) \rightarrow \overline{x_2}$, використовуючи метод невизначених коефіцієнтів.

Загальний вигляд поліному Жегалкіна для функції трьох змінних:

$$f(x_1, x_2, x_3) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_{12} x_1 x_2 + a_{13} x_1 x_3 + a_{23} x_2 x_3 + a_{123} x_1 x_2 x_3$$

Для цього спочатку необхідно побудувати таблицю істинності для булевої функції $f(x_1, x_2, x_3) = (x_1 x_2 \vee x_3) \rightarrow \overline{x_2}$.

Послідовно представимо набори значень змінних і знаходимо коефіцієнти: a_0, a_1, \dots, a_{123} .

$$f(0,0,1) = a_0 \oplus a_3 = 1 \oplus a_3 = 1, a_3 = 0;$$

$$f(0,1,0) = a_0 \oplus a_2 = 1 \oplus a_2 = 1, a_2 = 0;$$

$$f(0,1,1) = a_0 \oplus a_2 \oplus a_3 \oplus a_{23} = 1 \oplus 0 \oplus 0 \oplus a_{23} = 0, a_{23} = 1;$$

$$f(1,0,0) = a_0 \oplus a_1 = 1 \oplus a_1 = 1, a_1 = 0;$$

$$f(1,0,1) = a_0 \oplus a_1 \oplus a_3 \oplus a_{13} = 1 \oplus 0 \oplus 0 \oplus a_{13} = 1, a_{13} = 0;$$

$$f(1,1,0) = a_0 \oplus a_1 \oplus a_2 \oplus a_{12} = 1 \oplus 0 \oplus 0 \oplus a_{12} = 0, a_{12} = 1;$$

$$f(1,1,1) = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_{123} =$$

$$= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus a_{123} = 0, a_{123} = 1.$$

Підставляючи отримані коефіцієнти, отримаємо поліном Жегалкіна:

$$f(x_1, x_2, x_3) = 1 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1x_2x_3$$

За теоремою Поста, для того щоб система булевих функцій була повною необхідно й достатньо, щоб в ній існували:

1. Хоча б одна немонотонна функція.
2. Хоча б одна нелінійна функція.
3. Хоча б одна несамодвоїста функція.
4. Хоча б одна функція, що не зберігає нуль.
5. Хоча б одна функція, що не зберігає одиницю.

За теоремою Жегалкіна кожна булева функція єдиним чином представляється у вигляді поліному Жегалкіна. Теорема доводиться таким чином. Помітимо, що кількість різних булевих функцій від n змінних є $2^{(2^n)}$. При цьому, кон'юнкцій вигляду $x_1 \dots x_{ik}$ існує рівно 2^n , так як з n можливих елементів кожен або входить в кон'юнкцію, або ні.

В поліномі, в кожній такій кон'юнкції стоїть 0 або 1, тобто існує $2^{(2^n)}$ різних поліномів Жегалкіна від n змінних. Тепер потрібно лишень довести, що різні поліноми реалізують різні функції. Використаємо метод доведення від протилежного. Тоді, прирівнявши два різних поліноми і перенісши один із них в іншу частину рівності, отримаємо поліном, тотожно рівний нулю, і який має ненульові коефіцієнти. Тоді розглянемо доданок з одиничними коефіцієнтами з найменшою кількістю змінних (при повторенні змінних, беремо до уваги одну із них). Підставивши одиниці на місця цих змінних, і нулі на місця решти змінних, отримаємо, що на цьому наборі тільки один доданок набуває значення 1, тобто нульова функція на одному з наборів набуває значення 1.

Суперечність. Це означає, що кожна булева функція реалізується поліномом Жегалкіна єдиним унікальним чином [11].

3.2. Алгоритмізація задачі за темою роботи

Інформація в тренажері видається наступним чином: спочатку виводиться коротка довідка, потім практичне завдання.

При правильній відповіді користувач отримує повідомлення «Правильна відповідь», у іншому випадку користувач отримує повідомлення «Неправильна відповідь» та правильну відповідь.

Крок 1: Користувач отримує доступ до короткої довідки відносно алгебри та поліномів Жегалкіна. Коротка довідка:

«Алгебра Жегалкіна - алгебра булевих функцій утворених за допомогою:

- булевої константи 1
- булевих змінних
- булевих функцій:
- \oplus - сума за модулем 2
- кон'юнкція

Поліном Жегалкіна — довільна формула алгебри Жегалкіна, яка має вигляд суми кон'юнкцій булевих змінних. Поліном був запропонований в 1927 році Жегалкіним Іваном Івановичем, для зручного представлення булевих функцій алгебри логіки. В зарубіжній літературі представлення полінома Жегалкіна зазвичай називається алгебраїчною нормальною формою (АНФ). Якщо у кожний член поліному Жегалкіна кожна змінна входить один раз та поліном не містить однакових членів, то такий поліном Жегалкіна називається канонічним.»

Крок 2: Користувач отримує доступ до короткої довідки відносно операцій алгебри логіки. Коротка довідка:

«Властивості операції кон'юнкції:

$(x_1 \wedge x_2) \wedge x_3 = x_1 \wedge (x_2 \wedge x_3)$ - асоціативність;

$x_1 \wedge x_2 = x_2 \wedge x_1$ - комутативність;

$x_1 \wedge 0 = 0; x_1 \wedge 1 = x_1$ - дії з константами;

$x \wedge x = x$ - ідемпотентність.

Властивості операції сума за модулем 2:

$x_1 \oplus x_2 = x_2 \oplus x_1$ - комутативність;

$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3$ - асоціативність;

$x \oplus x = 0$ - закон зведення подібних доданків;

$x_1 \oplus (x_2 \oplus x_3) = x_1x_2 \oplus x_1x_3$ - дистрибутивність кон'юнкції відносно \oplus .

Решта операцій алгебри логіки виражаються через базис цієї алгебри в такий спосіб:

$$\bar{x} = 1 \oplus x;$$

$$x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1x_2;$$

$$x_1 \rightarrow x_2 = 1 \oplus x_1 \oplus x_1x_2;$$

$$x_1 \downarrow x_2 = 1 \oplus x_1 \oplus x_2 \oplus x_1x_2;$$

$$x_1 \sim x_2 = 1 \oplus x_1 \oplus x_2;$$

$$x_1 \leftarrow x_2 = x_1 \oplus x_1x_2;$$

$$x_1 | x_2 = 1 \oplus x_1x_2.$$

Крок 3 : Користувач отримує практичне завдання:

«Побудувати поліном Жегалкіна для $g = (x \vee y) \bar{z}$. Використаємо правило перетворення диз'юнкції: $g = (x \oplus y \oplus ??) \wedge (1 \oplus z)$ ».

Користувач отримує наступне завдання.

«Завдання: Заповніть пропуск в розв'язку»

Крок 4 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $xу$;
2. xx ;
3. yz »

Правильна відповідь 1. $xу$.

Крок 5 : Користувач отримує підказку до розв'язку.

«Розкриємо дужки: $g=(x \oplus y \oplus xy) \wedge (1 \oplus z) = x \oplus y \oplus xy \oplus xz \oplus ?? \oplus ???$.»

Користувач отримує наступне завдання.

«Завдання: Виберіть правильну відповідь»

Крок 6 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. xuz, xy ;
2. xx, xuz ;
3. yz, xuz ;

Правильна відповідь 3. yz, xuz .

Крок 7 : Користувач отримує фінальний розв'язок.

«Отримуємо поліном Жегалкіна для $g = (x \vee y) z = (x \oplus y \oplus xy) \wedge (1 \oplus z) = x \oplus y \oplus xy \oplus xz \oplus yz \oplus xuz$.»

Крок 8 : Користувач отримує практичне завдання:

«Перейти від формули $f = \bar{x}(x \vee y)$ до формули алгебри Жегалкіна для тієї ж функції».

Користувач отримує підказку до розв'язку.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = ???$ »

Крок 9 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $(x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy$;
2. $(x \oplus 1)x \oplus y \oplus xy$;
3. $1x \oplus xx \oplus 1y \oplus (1x \oplus xx)y$ »

Правильна відповідь 1. $(x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy$.

Крок 10 : Користувач отримує наступне завдання.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = (x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy = ???$ »

Крок 11 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y$;
2. $(x \oplus 1)x \oplus y \oplus xy$;
3. $1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y$.»

Правильна відповідь 3. $1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y$.

Крок 12 : Користувач отримує наступне завдання.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = (x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy =$
 $= 1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y = ???$ »

Крок 13 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y$;
2. $x \oplus y \oplus xy \oplus (x \oplus x)y$;
3. $x \oplus x \oplus y \oplus xy \oplus (x \oplus x)xy$.»

Правильна відповідь 1. $x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y$.

Крок 14 : Користувач отримує наступне завдання.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = (x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy =$
 $= 1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y = x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y = ???$ »

Крок 15 : Користувач обирає варіант відповіді.

Правильна відповідь: $0 \oplus y \oplus xy \oplus 0y$.

Крок 16 : Користувач отримує наступне завдання.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = (x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy =$
 $= 1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y = x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y =$
 $= 0 \oplus y \oplus xy \oplus 0y = ???$ »

Крок 17 : Користувач обирає варіант відповіді.

Правильна відповідь: $y \oplus xy$.

Крок 18 : Користувач отримує фінальний розв'язок.

«Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = (x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy =$

$$= 1x \oplus xx \oplus 1y \oplus xy \oplus (1x \oplus xx)y = x \oplus x \oplus y \oplus xy \oplus (x \oplus x)y = \\ = 0 \oplus y \oplus xy \oplus 0y = y \oplus xy \text{ »}$$

Крок 19 : Користувач отримує практичне завдання:

«Перетворити $\varphi = x \vee y \vee \bar{z}$ до формули алгебри Жегалкіна».

Користувач отримує підказку до розв'язку.

«Маємо $\varphi = x \vee y \vee (1 \oplus z) = ???$ »

Крок 20 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $x \oplus y \oplus xy \vee (1 \oplus z)$;
2. $(x \oplus y \oplus xy) \vee (1 \oplus z)$;
3. $(x \oplus y) \vee (1 \oplus z)$.»

Правильна відповідь 2. $(x \oplus y \oplus xy) \vee (1 \oplus z)$.

Крок 21 : Користувач отримує наступне завдання.

«Маємо $\varphi = x \vee y \vee (1 \oplus z) = (x \oplus y \oplus xy) \vee (1 \oplus z) = ???$ »

Крок 22 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y)(1 \oplus z)$;
2. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xy(1 \oplus z)$;
3. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y \oplus xy)(1 \oplus z)$.»

Правильна відповідь 3. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y \oplus xy)(1 \oplus z)$.

Крок 23 : Користувач отримує наступне завдання.

«Маємо $\varphi = x \vee y \vee (1 \oplus z) = (x \oplus y \oplus xy) \vee (1 \oplus z) =$

$= x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y \oplus xy)(1 \oplus z) = ???$ »

Крок 24 : Користувач обирає варіант відповіді з запронованих.

«Варіанти відповіді:

1. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xz \oplus yz$;
2. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus xz \oplus yz \oplus xyz$;
3. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xy \oplus xz \oplus yz \oplus xyz$.»

Правильна відповідь 3. $x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xy \oplus xz \oplus yz \oplus xyz$.

Крок 25 : Користувач отримує наступне завдання.

$$\begin{aligned} \text{«Маємо } \varphi &= x \vee y \vee (1 \oplus z) = (x \oplus y \oplus xy) \vee (1 \oplus z) = \\ &= x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y \oplus xy)(1 \oplus z) = \\ &= x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xy \oplus xz \oplus yz \oplus xyz = \text{«???»} \end{aligned}$$

Крок 26 : Користувач обирає варіант відповіді.

Правильна відповідь: $1 \oplus z \oplus xz \oplus yz \oplus xyz$.

Крок 27 : Користувач отримує фінальний розв'язок.

$$\begin{aligned} \text{«Маємо } \varphi &= x \vee y \vee (1 \oplus z) = (x \oplus y \oplus xy) \vee (1 \oplus z) = \\ &= x \oplus y \oplus xy \oplus 1 \oplus z \oplus (x \oplus y \oplus xy)(1 \oplus z) = \\ &= x \oplus y \oplus xy \oplus 1 \oplus z \oplus x \oplus y \oplus xy \oplus xz \oplus yz \oplus xyz = 1 \oplus z \oplus xz \oplus yz \oplus xyz \text{»} \end{aligned}$$

Крок 28: Користувач отримує доступ до короткої довідки відносно способів побудови поліномів Жегалкіна. Коротка довідка:

«Теорема Жегалкіна — стверджує існування і унікальність будь-якої булевої функції у вигляді поліному Жегалкіна. Формально поліном Жегалкіна можна представити у вигляді:

$$\begin{aligned} P(x_1, \dots, x_n) &= a \oplus a_1 \wedge x_1 \oplus a_2 \wedge x_2 \oplus \dots \oplus a_n x_n \oplus a_{12} \wedge x_1 \wedge x_2 \oplus a_{13} \wedge x_1 \wedge x_3 \oplus \\ &\oplus \dots \oplus a_{1\dots n} \wedge x_1 \dots \wedge x_n \end{aligned}$$

»

Крок 29: Користувач отримує доступ до короткої довідки відносно способів побудови поліномів Жегалкіна. Коротка довідка:

«Метод трикутника Паскаля.»

Крок 1. Будуємо таблицю значень функції (рядки таблиці йдуть в порядку збільшення двоїстих кодів). Таблицю краще розташовувати в лівій частині сторінки.

Крок 2. Побудова трикутника. Для цього беремо вектор значень функції і виписуємо його навпроти першого рядка таблиці. Далі заповнюємо трикутник, складаючи попарно сусідні значення за модулем 2. Результат додавання записуємо нижче.

Крок 3. Побудова полінома Жегалкіна. Нас цікавить ліва сторона трикутника (значення виділені жирним). Числа на лівій стороні (виділені жирним шрифтом) трикутника є коефіцієнтами поліному при монотонних кон'юнкціях, які відповідають наборам значень змінних. Тепер випишемо для наочності ці кон'юнкції. Кон'юнкції випишемо за двоїстими наборами в лівій частині таблиці за наступним принципом: якщо навпроти змінної x_i стоїть 1, то змінна входить в кон'юнкцію, в протилежному випадку – змінна відсутня в кон'юнкції. Набору (0,0,0) відповідає 1.

Якщо принцип отримання кон'юнкцій зрозумілий, то стовпчик з ним (краще) не випишувати, а одразу переходити до побудови поліному. Для побудови поліному необхідні кон'юнкції тільки з рядків з одиницями на лівій стороні трикутника. Це і є кон'юнкції, що входять до складу полінома Жегалкіна.»

Крок 30 : Користувач отримує практичне завдання:

«Приклад 1. Побудувати поліном Жегалкіна для функції f . Візьмемо функцію $f(x_1, x_2, x_3) = x_1 x_2 \vee x_2 x_3 \vee x_1 x_3 = (00010111)$.»

Користувач отримує підказку до розв'язку.

«Крок 1. Будуємо таблицю значень функції»

x_1	x_2	x_3	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Крок 31 : Користувач обирає варіант відповіді. Правильна відповідь:

x_1	x_2	x_3	f
0	0	0	0

0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Крок 32 : Користувач отримує наступне завдання:

«Крок 2. Побудова трикутника»

x_1	x_2	x_3	f	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

Крок 33 : Користувач обирає варіант відповіді. Правильна відповідь:

x_1	x_2	x_3	f	
0	0	0	0	0 0 0 1 0 1 1 1
0	0	1	0	0 0 1 1 1 0 0
0	1	0	0	0 1 0 0 1 0
0	1	1	1	1 1 0 1 1
1	0	0	0	0 1 1 0
1	0	1	1	1 0 1
1	1	0	1	1 1
1	1	1	1	0

Крок 34 : Користувач отримує наступне завдання:

«Крок 3. Побудова полінома Жегалкіна»

x_1	x_2	x_3	f		
0	0	0	0		0 0 0 1 0 1 1 1
0	0	1	0		0 0 1 1 1 0 0

0	1	0	0		0 1 0 0 1 0
0	1	1	1		1 1 0 1 1
1	0	0	0		0 1 1 0
1	0	1	1		1 0 1
1	1	0	1		1 1
1	1	1	1		0

Крок 35 : Користувач обирає варіант відповіді. Правильна відповідь:

x_1	x_2	x_3	f		
0	0	0	0	1	0 0 0 1 0 1 1 1
0	0	1	0	x_3	0 0 1 1 1 0 0
0	1	0	0	x_2	0 1 0 0 1 0
0	1	1	1	$x_2 x_3$	1 1 0 1 1
1	0	0	0	x_1	0 1 1 0
1	0	1	1	$x_1 x_3$	1 0 1
1	1	0	1	$x_1 x_2$	1 1
1	1	1	1	$x_1 x_2 x_3$	0

Крок 36 : Користувач отримує наступне завдання.

«Залишилося вписати сам поліном: $f(x_1, x_2, x_3) = \text{???}$ »

Крок 37 : Користувач обирає варіант відповіді.

Правильна відповідь: $x_1 x_2 + x_2 x_3 + x_1 x_3$.

Крок 38 : Користувач отримує фінальний розв'язок.

«Побудовано поліном Жегалкіна для функції f :

$$f(x_1, x_2, x_3) = x_1 x_2 + x_2 x_3 + x_1 x_3 \text{»}$$

Крок 39: Користувач отримує доступ до короткої довідки відносно способів побудови поліномів Жегалкіна. Коротка довідка:

«Метод невизначених коефіцієнтів.»

Приклад 2. Знайдемо поліном Жегалкіна для функції $f(x_1, x_2, x_3) = (x_1 x_2 \vee x_3) \rightarrow \overline{x_2}$, використовуючи метод невизначених коефіцієнтів.

Загальний вигляд поліному Жегалкіна для функції трьох змінних:

$$f(x_1, x_2, x_3) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_{12} x_1 x_2 + a_{13} x_1 x_3 + a_{23} x_2 x_3 + a_{123} x_1 x_2 x_3 \text{»}$$

Крок 40 : Користувач отримує практичне завдання:

«Для цього спочатку необхідно побудувати таблицю істинності для булевої функції $f(x_1, x_2, x_3) = (x_1 x_2 \vee x_3) \rightarrow \overline{x_2}$.»

x_1	x_2	x_3	$x_1 x_2$	$x_1 x_2 \vee x_3$	$\overline{x_2}$	f
0	0	0	0	0	1	
0	0	1	0	1	1	
0	1	0	0	0	0	
0	1	1	0	1	0	
1	0	0	0	0	1	
1	0	1	0	1	1	
1	1	0	1	1	0	
1	1	1	1	1	0	

Крок 41 : Користувач обирає варіант відповіді. Правильна відповідь:

x_1	x_2	x_3	$x_1 x_2$	$x_1 x_2 \vee x_3$	$\overline{x_2}$	f
0	0	0	0	0	1	1
0	0	1	0	1	1	1
0	1	0	0	0	0	1
0	1	1	0	1	0	0
1	0	0	0	0	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	1	1	0	0

Крок 42 : Користувач отримує наступне завдання.

«Послідовно представимо набори значень змінних і знаходимо коефіцієнти: a_0, a_1, \dots, a_{123} .

$$f(0,0,0) = a_0 = 1;$$

$$f(0,0,1) = a_0 \oplus a_3 = 1 \oplus a_3 = ??;$$

$$f(0,1,0) = a_0 \oplus a_2 = 1 \oplus a_2 = ??;$$

$$f(0,1,1) = a_0 \oplus a_2 \oplus a_3 \oplus a_{23} = 1 \oplus 0 \oplus 0 \oplus a_{23} = ??;$$

$$f(1,0,0) = a_0 \oplus a_1 = 1 \oplus a_1 = ??;$$

$$f(1,0,1) = a_0 \oplus a_1 \oplus a_3 \oplus a_{13} = 1 \oplus 0 \oplus 0 \oplus a_{13} = ??;$$

$$f(1,1,0) = a_0 \oplus a_1 \oplus a_2 \oplus a_{12} = 1 \oplus 0 \oplus 0 \oplus a_{12} = ??;$$

$$f(1,1,1) = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_{123} =$$

$$= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus a_{123} = ?? \rangle$$

Крок 43 : Користувач обирає варіант відповіді.

Правильна відповідь:

$$f(0,0,1) = a_0 \oplus a_3 = 1 \oplus a_3 = 1, a_3 = 0;$$

$$f(0,1,0) = a_0 \oplus a_2 = 1 \oplus a_2 = 1, a_2 = 0;$$

$$f(0,1,1) = a_0 \oplus a_2 \oplus a_3 \oplus a_{23} = 1 \oplus 0 \oplus 0 \oplus a_{23} = 0, a_{23} = 1;$$

$$f(1,0,0) = a_0 \oplus a_1 = 1 \oplus a_1 = 1, a_1 = 0;$$

$$f(1,0,1) = a_0 \oplus a_1 \oplus a_3 \oplus a_{13} = 1 \oplus 0 \oplus 0 \oplus a_{13} = 1, a_{13} = 0;$$

$$f(1,1,0) = a_0 \oplus a_1 \oplus a_2 \oplus a_{12} = 1 \oplus 0 \oplus 0 \oplus a_{12} = 0, a_{12} = 1;$$

$$f(1,1,1) = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_{123} =$$

$$= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus a_{123} = 0, a_{123} = 1.$$

Крок 44 : Користувач отримує наступне завдання.

«Підставляючи отримані коефіцієнти, отримаємо поліном Жегалкіна:

$$f(x_1, x_2, x_3) = ?? \rangle$$

Крок 45 : Користувач обирає варіант відповіді.

Правильна відповідь:

$$f(x_1, x_2, x_3) = 1 + x_1x_2 + x_2x_3 + x_1x_2x_3.$$

Крок 46 : Користувач отримує фінальний розв'язок.

«Побудовано поліном Жегалкіна для функції f :

$$f(x_1, x_2, x_3) = 1 + x_1x_2 + x_2x_3 + x_1x_2x_3 \rangle$$

Переміщення між частинами завдання реалізовані у вигляді кнопок «Далі» та «Назад».

Після закінчення роботи з тренажером з'явиться кнопка повтору, що перенаправить користувача на початок роботи з тренажером.

3.3. Розробка блок-схеми, яка підлягає програмуванню

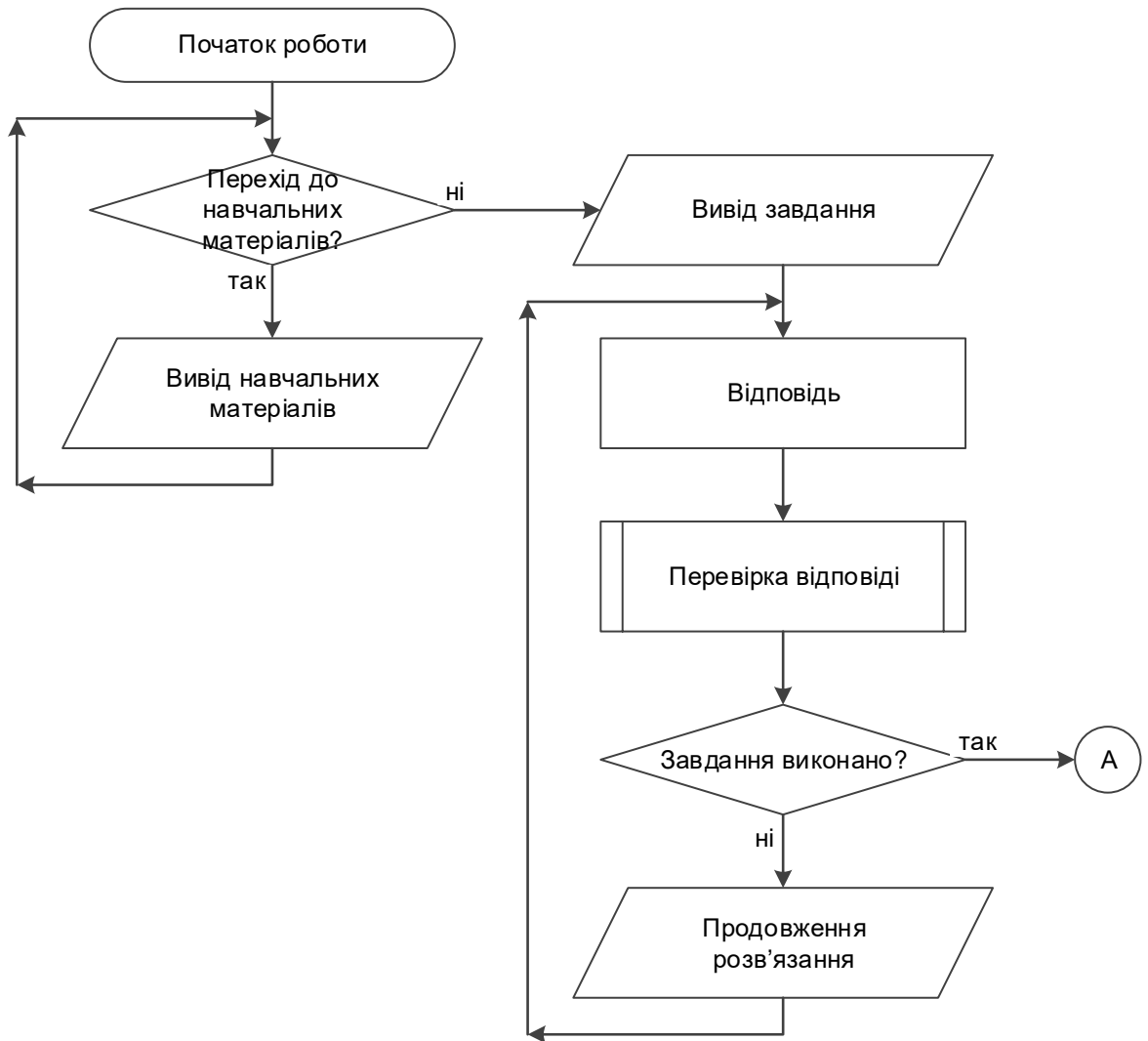


Рисунок 3.1 – Блок-схема роботи тренажера

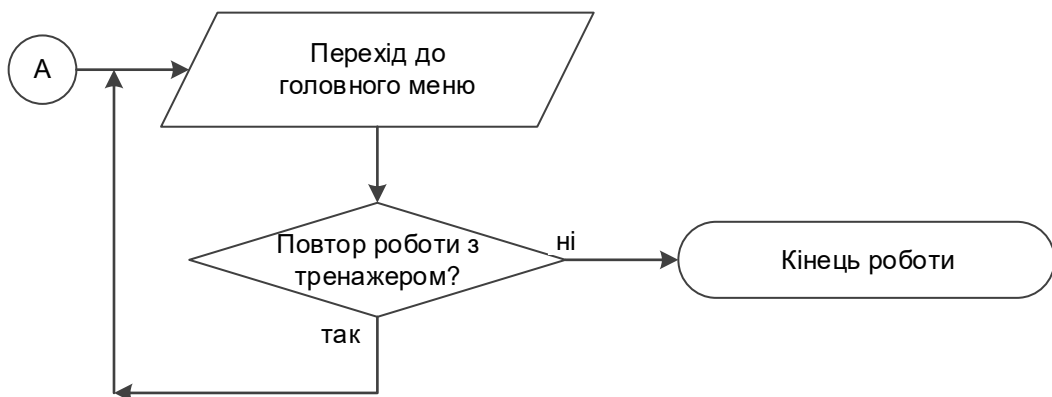


Рисунок 3.2 – Продовження блок-схеми роботи тренажера

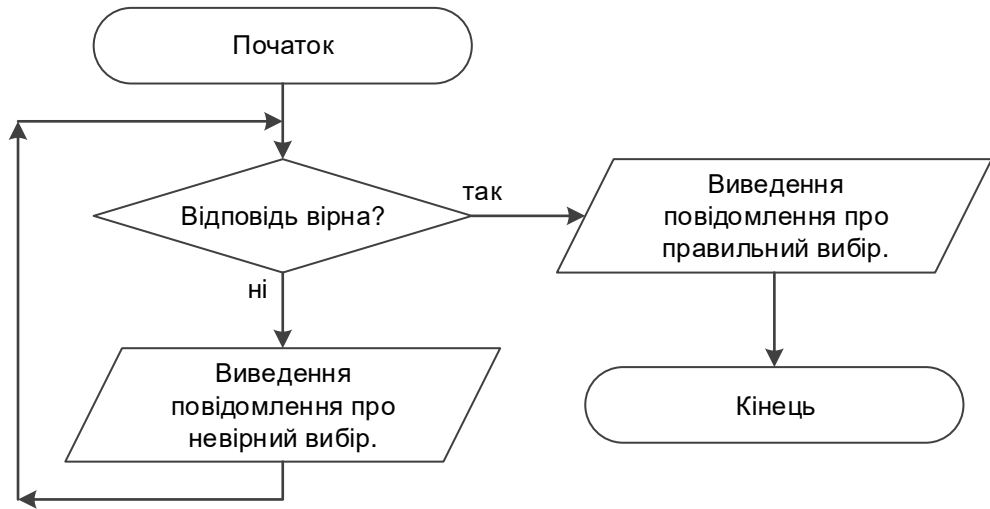


Рисунок 3.3 – Блок-схема перевірки відповіді

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників — мов C++, Object Pascal, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів.

C# має «препроцесорні директиви» (хоча насправді він не має препроцесора) на основі препроцесора C, це дає програмісту можливість визначити символи, але не макроси. Умовні директиви, такі як #if, #endif, чи #else також можливі. Директиви типу #region дають натяк редактору для згортання фрагментів коду.

Специфікація C# визначає мінімальний набір бібліотек типів і класів, на який має розраховувати компілятор. На практиці, C# найчастіше використовується з якоюсь реалізацією Common Language Infrastructure[en] (CLI), яка стандартизована як ECMA-335 Common Language Infrastructure (CLI).

Титульним компілятором C# є Microsoft Visual C#.

Існують інші компілятори C#, часто вони включають реалізації Common Language Infrastructure і бібліотеки класів .NET:

- Проєкт Microsoft Rotor (який тепер зветься Shared Source Common Language Infrastructure, ліцензований тільки для навчального і дослідницького використання) забезпечує реалізації CLR runtime і компілятор C#, і підмножину бібліотек фреймворка Common Language Infrastructure, відповідно до специфікації ECMA (до C# 2.0, і з підтримкою тільки Windows XP).

- Проєкт SharpDevelop від компанії icsharpcode, який використовується як альтернатива Visual Studio. Забезпечує повну реалізацію Common Language Infrastructure. Остання[коли?] стабільна версія IDE 4.4 (28 серпня 2013), тестова версія 5.0 (13 лютого 2014). Зовнішній вигляд IDE дуже нагадує Microsoft Visual C#, що робить комфортним перехід від одного середовища до іншого.

- Проєкт Mono, початий компанією Xamarin і продовжений її покупцем і наступником Novell, забезпечує відкритий компілятор C#, повну відкриту реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і близьку до повної реалізацію власницьких бібліотек класів Microsoft .NET до .NET 2.0, але не специфічних бібліотек .NET 3.0 і .NET 3.5, як для Mono 2.0.

- Проєкт DotGNU також надає відкритий компілятор C#, близьку до повної реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і підмножину деяких залишених власницьких бібліотек класів Microsoft .NET до .NET 2.0 (які не документовані або не включені у специфікації ECMA, але включені у стандартне визначення Microsoft .NET Framework).

- DotNetAnywhere Micro Framework Common Language Runtime націлений на вбудовані системи, і підтримує майже всі специфікації C# 2.0 [12].

Unity — багатоплатформовий інструмент для розробки відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна — це оглядач ресурсів проєкту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів.

Проєкт в Unity поділяється на сцени (рівні) — окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти-моделі (ландшафт, персонажі, предмети довкілля тощо), так і порожні ігрові об'єкти — ті, що не мають моделі, проте задають поведінку інших об'єктів (тригери подій, точки збереження прогресу тощо). Їх дозволяється розташовувати, обертати, масштабувати, застосовувати до них скрипти. В них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково наявний компонент Transform — він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою. Різні моделі можуть об'єднуватися в набори (ассети) для швидкого доступу до них. Наприклад, моделі споруд на спільну тему.

Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу

батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

У 2D іграх Unity переважно використовує спрайти. В 3D іграх Unity здебільшого використовує тривимірні моделі (меші), на які накладаються текстури (зумовлюють вигляд поверхні об'єктів), матеріали (зумовлюють як поверхня реагуватиме на різні фактори) та шейдери (невеликі скрипти, за яким вираховується зміна кольору кожного пікселя згідно заданих параметрів, як-от розсіяння відбитого світла). В обох видах застосовуються системи часток для відображення субстанцій, таких як рідини чи дим.

Unity підтримує стиснення текстур, міпмапінг і різні налаштування роздільності екрана для кожної платформи; забезпечує бамп-мапінг, мапінг відображень, паралакс-мапінг, затінення навколишнього світла у екранному просторі, динамічні тіні за картами тіней, рендер у текстуру та повноекранні ефекти обробки зображення, такі як зернистість, глибина чіткості, розмиття в русі, відблиски віртуальних лінз або ореол навколо джерел світла.

Рендеринг зображення відбувається через віртуальну камеру огляду. В робочій області редактора ігрова сцена може розміщуватися як завгодно, а при рендерингу — так, як її видно з камери. В сцені може бути декілька камер, які рухаються за персонажем чи за вказаною траєкторією. Вигляд з камери подається в двовимірно чи тривимірно (в перспективі або ортографічно). Фон сцени, видимий через камеру, типово зображає небо, утворене скайбоксом, але може презентувати й інше довкілля.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii. Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing.

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance. В ігровий проєкт Unity можна імпортувати об'єкти цих програм та виконувати налаштування за допомогою графічного інтерфейсу.

Для написання шейдерів використовується ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Шейдер може включати декілька варіантів реалізації, що дозволяє Unity визначати найкращий варіант для конкретної відеокарти. Unity також має вбудовану підтримку фізичного рушія Nvidia PhysX (колишнього Ageia), підтримку симуляції одягу в системі реального часу на довільній та прив'язаній полігональній сітці (починаючи з Unity 3.0), підтримку системи ray casts та шарів зіткнення.[13]

4.2. Опис процесу програмної реалізації

Інструменти Visual Studio для Unity - це безкоштовне розширення, яке забезпечує підтримку написання і налагодження C # і багато іншого.

Кроки для встановлення (якщо не встановлено Unity):

1. Запустити установник Visual Studio або запустити його, якщо він вже встановлений.
2. Клацніть Змінити (якщо встановлено) або Установка (для нової установки) для необхідної версії Visual Studio.
3. На вкладці робочі навантаження перейдіть до розділу гри і виберіть робоче навантаження Розробка ігор за допомогою Unity.

Коли Visual Studio стане зовнішнім редактором для Unity, при подвійному натисканні скрипта в редакторі Unity автоматично запускається або перемикається в Visual Studio і можна відкрити скрипт.

Крім того, можна відкрити Visual Studio без відкриття скрипта в редакторі вихідного коду, вибравши пункт активи > відкрити проект C # в Unity (рис. 4.1).

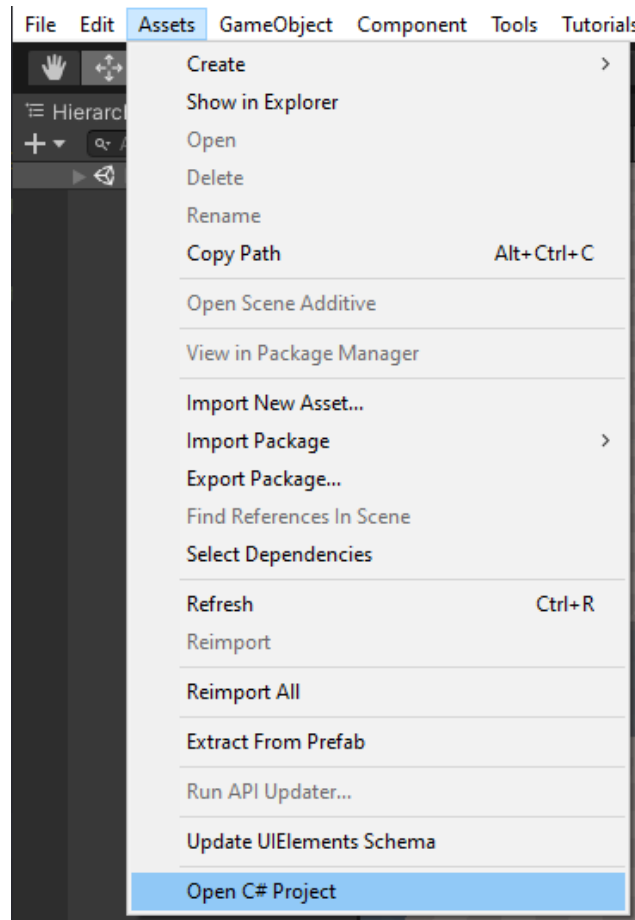


Рисунок 4.1 – Відкрити проект C # в Unity

Засіб автозавершення коду IntelliSense дозволяє легко реалізувати повідомлення API Unity в скриптах MonoBehaviour і допомагає у вивченні API Unity. Щоб використовувати IntelliSense для повідомлень Unity, зробіть наступне:

1. Помістіть курсор на новий рядок усередині тіла класу, похідного від MonoBehaviour.
2. Почніть вводити ім'я повідомлення Unity, наприклад OnTriggerEnter.
3. Після введення букв "ontri" відображається список пропозицій IntelliSense (рис. 4.2).

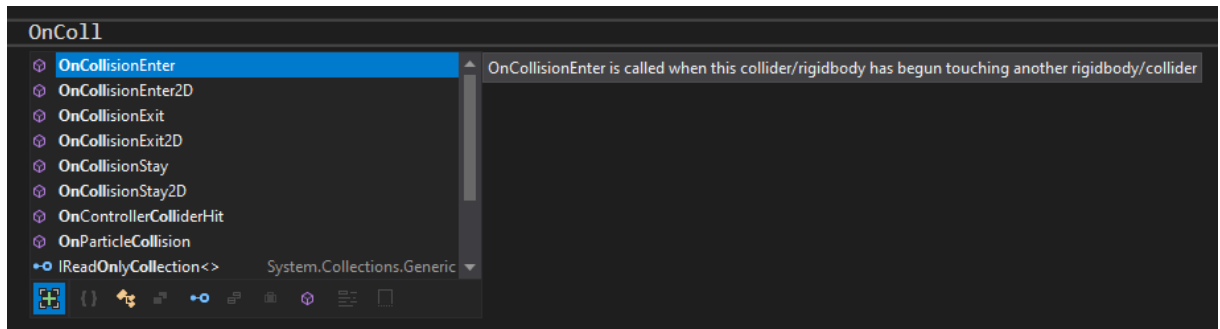


Рисунок 4.2 – Список пропозицій IntelliSense

4. Обраний елемент в списку можна змінити трьома способами:

- за допомогою клавіш зі стрілками ВГОРУ та ВНИЗ;
- клацаючи потрібний елемент;
- продовживши введення імені потрібного елемента.

5. IntelliSense може вставити вибране повідомлення Unity, включаючи всі необхідні параметри:

- при натисканні клавіші ТАБ;
- при натисканні клавіші ВВЕДЕННЯ;
- при подвійному натисканні обраного елемента.

Майстер MonoBehaviour можна використовувати для перегляду списку всіх методів API Unity і швидкого виконання порожнього визначення. Ця функція, особливо з увімкненим параметром Створити коментарі для методу, корисна при вивченні того, що є в Unity API.

Створення визначень порожніх методів MonoBehaviour за допомогою майстра MonoBehaviour.

1. У Visual Studio помістіть курсор туди, де необхідно вставити методи, а потім натисніть клавіші CTRL + SHIFT + M, щоб запустити майстер MonoBehaviour. У Visual Studio для Mac, натисніть cmd + SHIFT + M.

2. У вікні Створення методів скрипта встановіть прапорці поруч з ім'ям кожного методу, який потрібно додати.

3. Щоб вибрати необхідну версію, використовуйте список, що розкривається «Версія платформи».

4. Методи за замовчуванням вставляються в позицію курсора. В якості альтернативи можна вставити їх після будь-якого методу, який вже реалізований у класі. Для цього змініть значення списку «точка вставки» на потрібне розташування.

5. Якщо ви хочете, щоб майстер створював коментарі для обраних методів, встановіть прапорець «Створити коментарі до методу». Ці коментарі дозволяють краще зрозуміти, коли викликається метод і які його загальні функції.

6. Натисніть кнопку ОК, щоб вийти з майстра й вставити методи в код (рис. 4.3).

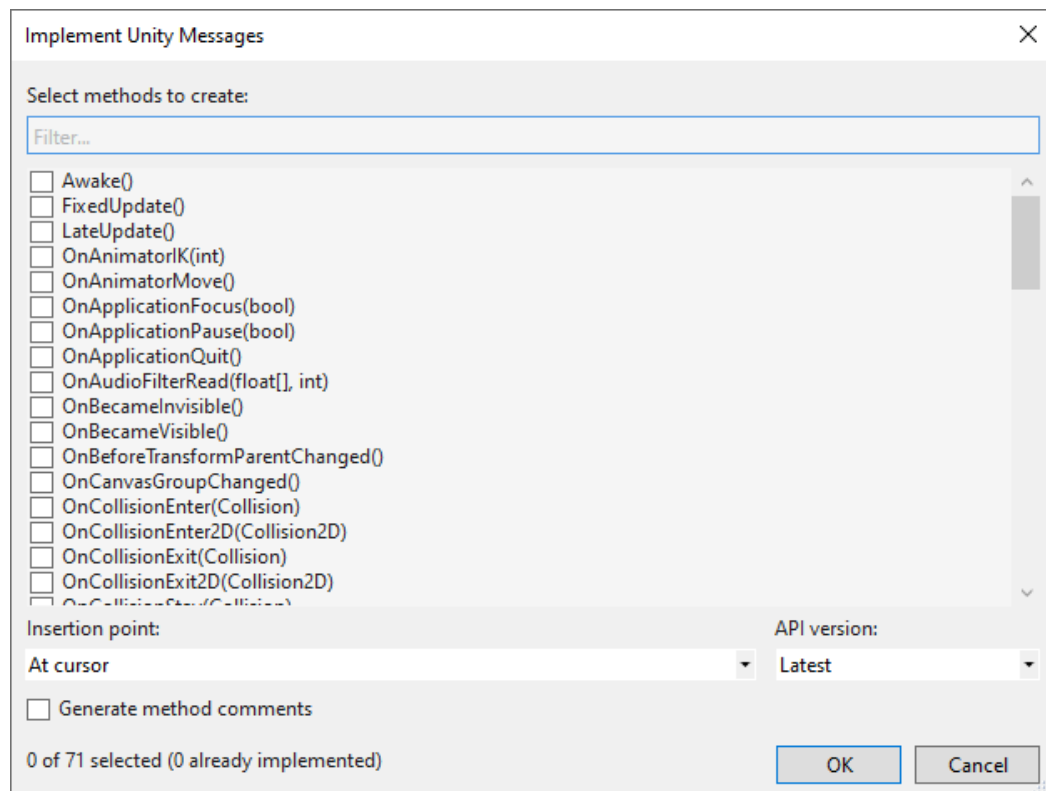


Рисунок 4.3 – Створення визначень порожніх методів MonoBehaviour

Всі переходи між навчальними матеріалами тренажеру оформлені за допомогою скрипту доданого до об'єкту головне меню в сцені Unity.

Код для кнопки «Далі» отримує активну сцену (`SceneManager.GetActiveScene()`), тобто слайд при роботі, та вимикає її (`SetActive(false)`), потім отримує наступну сцену та робить її активною.

Код кнопки «Далі»:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
    public void NextButton()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().SetActive(false
);
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +
1).SetActive(true);
    }
}
```

Роботу кнопки «Назад» вимикає поточну сцену та робить активною попередню.

Код кнопки «Назад»:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
    public void NextButton()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().SetActive(false
);
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
1).SetActive(true);
    }
}
```

Також реалізовано кнопку повернення до головного меню в будь-який момент роботи з тренажером.

Код кнопки повернення до головного меню:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
    public void MenuButton()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene()).SetActive(false
);
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex)
.SetActive(true);
    }
}
```

При роботі з завданнями практичного типу кнопка «Далі» з'являється тільки після вводу відповіді.

4.3. Опис програми, її інструкція

Спочатку вибираються налаштування відображення тренажера і потрібно натиснути «Play» (рис. 4.4). Для виходу «Quit».

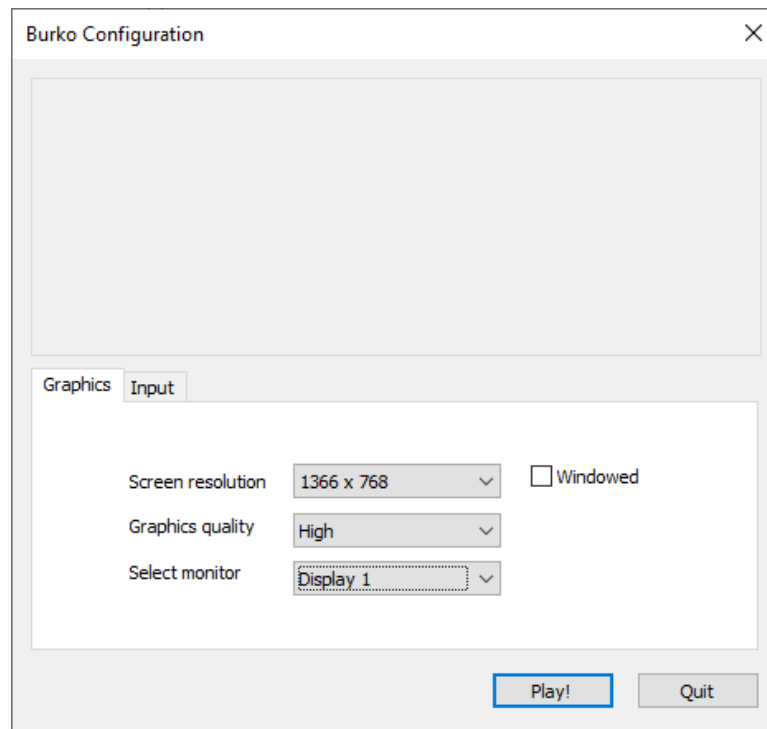


Рисунок 4.4 – Налаштування відображення тренажера

Після цього відображається головне вікно тренажера з основною інформацією (рис. 4.5):

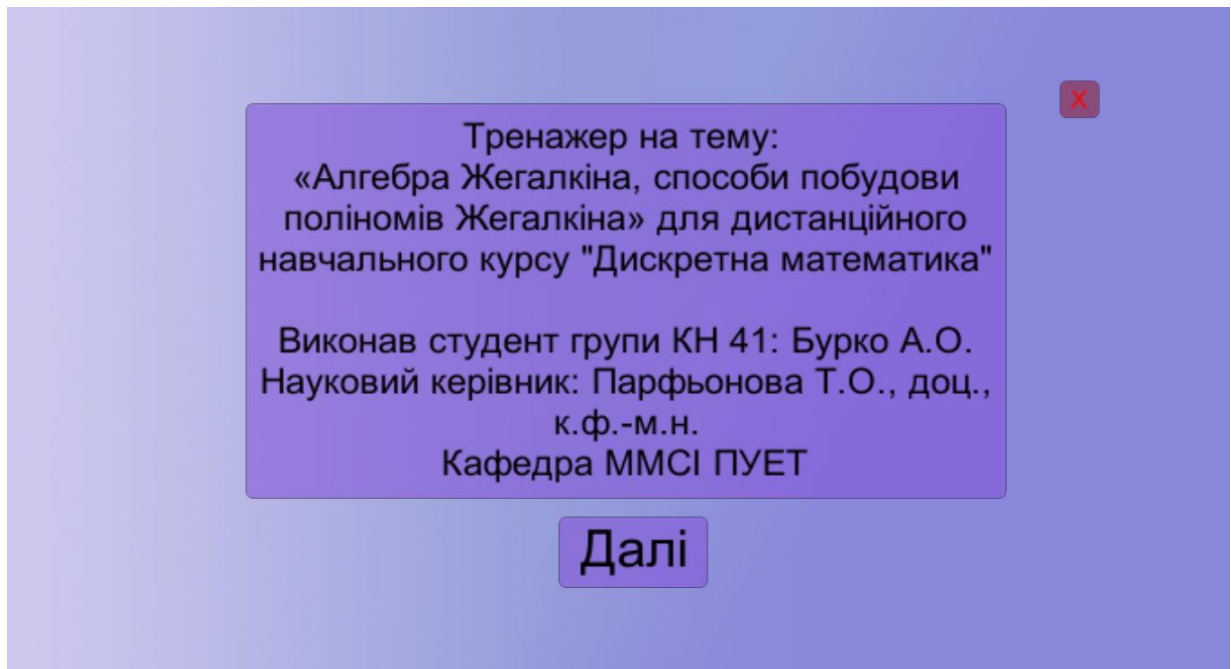


Рисунок 4.5 – Головне вікно

Наступним кроком виводиться головне меню, де можна переглянути матеріал по темі або виконувати завдання (рис. 4.6):

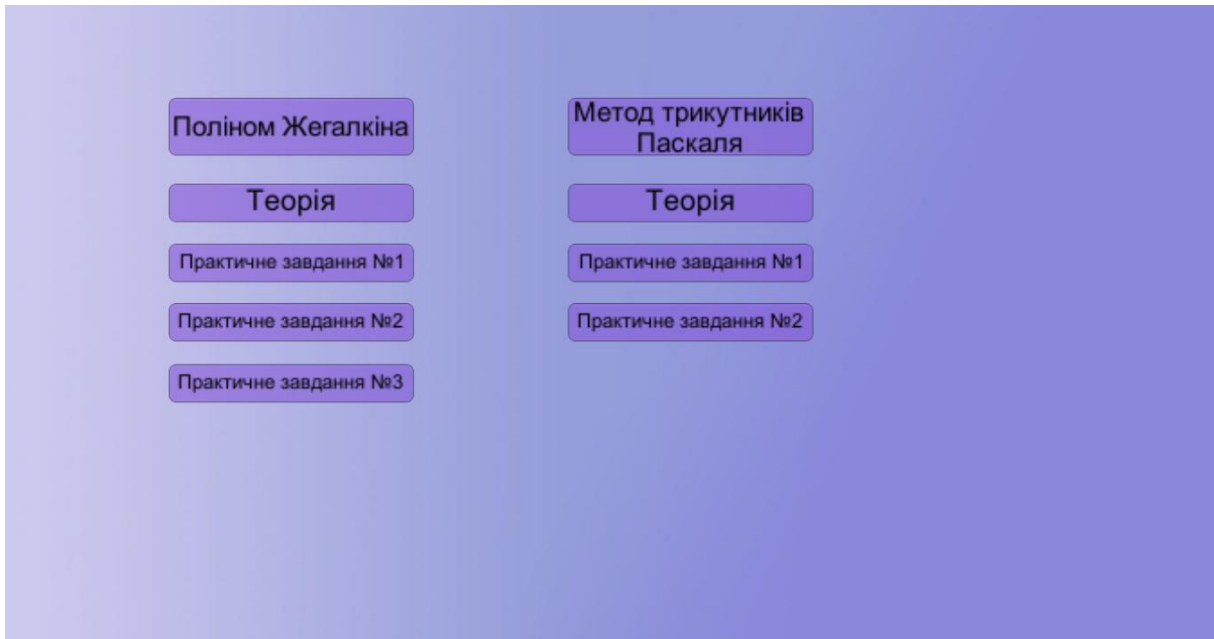


Рисунок 4.6 – Головне меню

Для перегляду теорії потрібно натиснути на відповідну кнопку (рис. 4.7, 4.8). При цьому навігація між слайдами здійснюється кнопками «Далі» і «Назад».

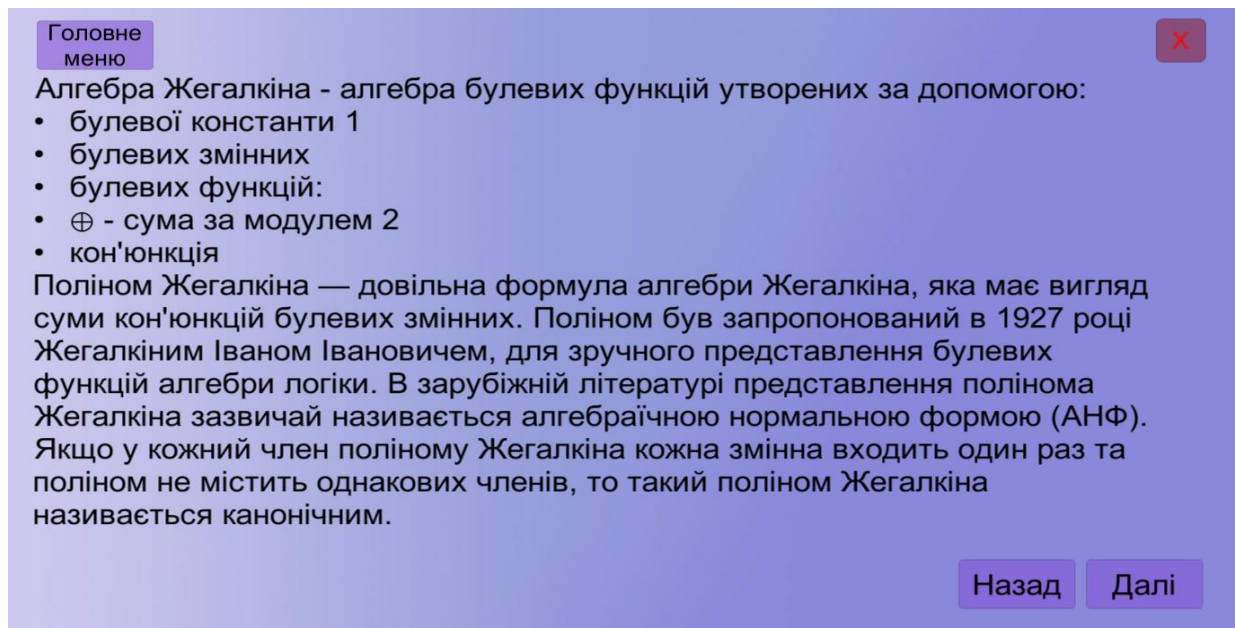


Рисунок 4.7 – Основна теорія з теми

Перші завдання являють собою вибір відповіді на питання по теорії (рис. 4.8).

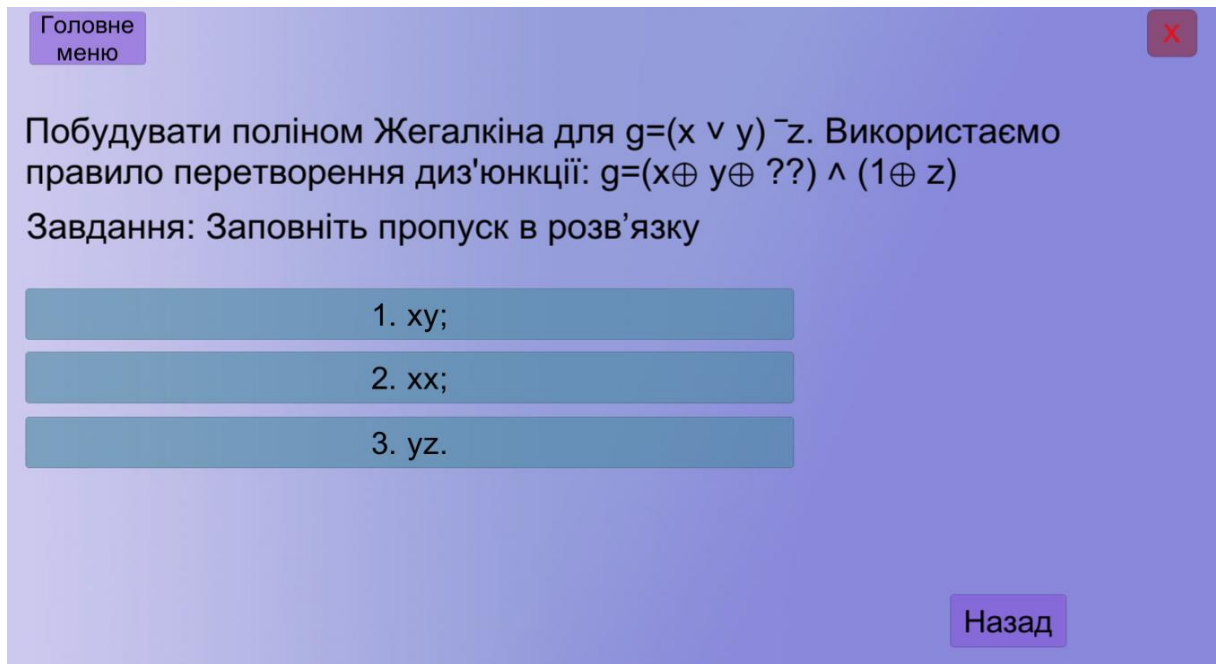


Рисунок 4.8 – Перше завдання

В наступних завданнях розв'язуються деякі приклади. Якщо після вибору відповіді вказується «Правильно», то можна перейти далі (рис. 4.9). Якщо «Неправильно» – потрібно вибрати іншу відповідь (рис. 4.10).

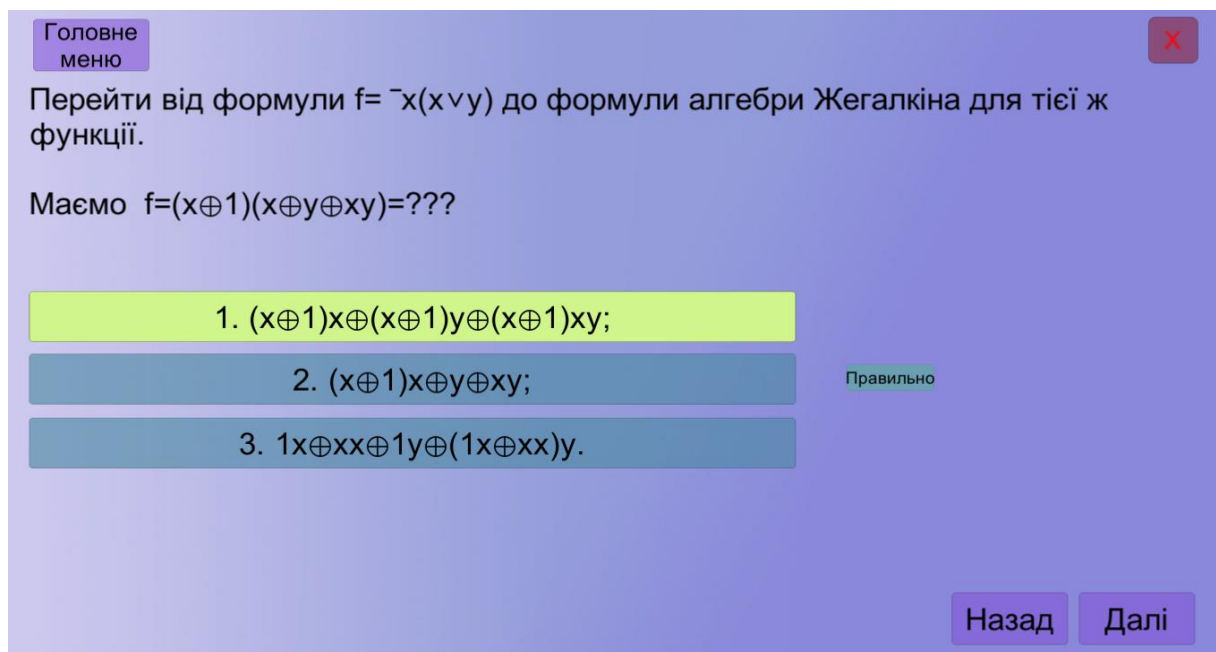


Рисунок 4.9 – Друге завдання (Правильно)

Головне меню ✖

Перейти від формули $f = \neg x(x \vee y)$ до формули алгебри Жегалкіна для тієї ж функції.

Маємо $f = (x \oplus 1)(x \oplus y \oplus xy) = ???$

1. $(x \oplus 1)x \oplus (x \oplus 1)y \oplus (x \oplus 1)xy$;

2. $(x \oplus 1)x \oplus y \oplus xy$;

3. $1x \oplus xx \oplus 1y \oplus (1x \oplus xx)y$.

Неправильно

Назад

Рисунок 4.10 – Друге завдання (Неправильно)

В завданнях по побудові полінома Жегалкіна необхідно виконувати покроково етапи розв'язування (рис. 4.11).

Головне меню ✖

Крок 1. Будуємо таблицю значень функції.
Натисніть на правильний варіант заповненої таблиці

1.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2.

x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

x_1	x_2	x_3	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Назад

Рисунок 4.11 – Четверте завдання

В кінці тренажера пропонується повторити роботу (рис. 4.12). В будь-який момент можна перейти до головного меню.

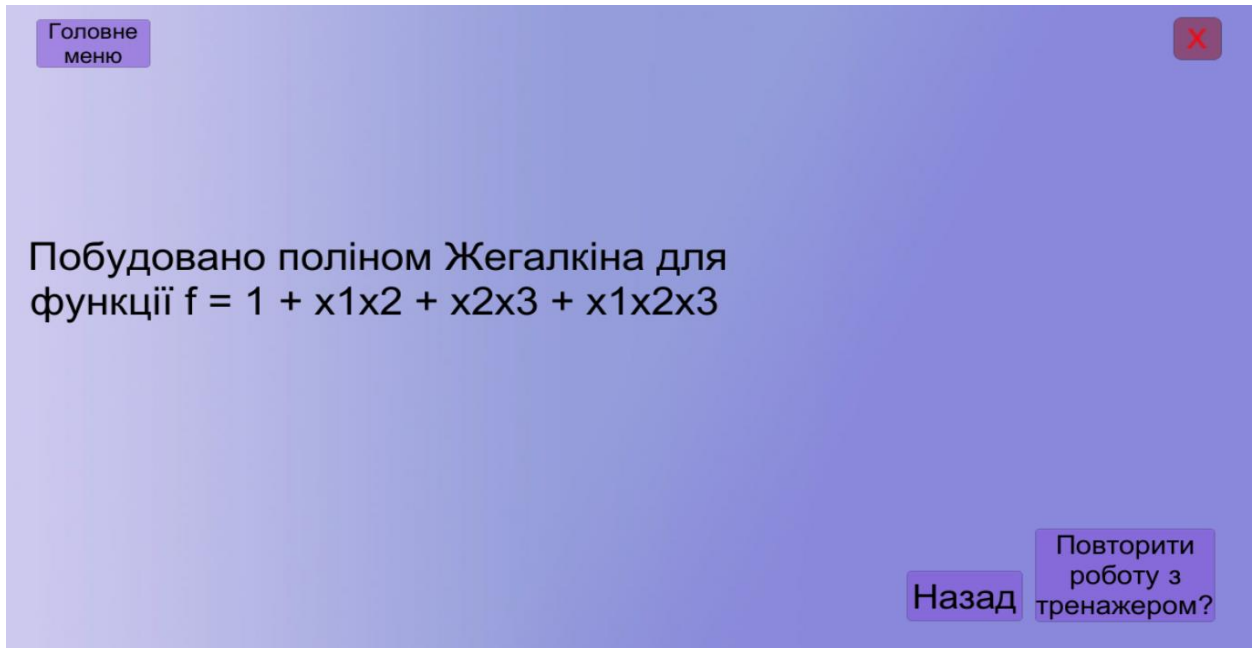


Рисунок 4.12 – Кінець тренажера

ВИСНОВКИ

Результатами роботи є:

1. Обрано програмне середовище – Visual Studio та мову програмування C#;
2. Програмний продукт розроблено на платформі Unity;
3. Розроблено алгоритм роботи тренажера з теми «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна» для дистанційного курсу «Дискретна математика».
4. Складено блок-схему до алгоритму.
5. Програмно реалізовано навчальну програму.
6. Виконано тестування програми.

Плюсами розробленого програмного забезпечення є:

1. Приємний очам дизайн та інтуїтивно зрозумілий інтерфейс програми.
2. Передбачено перевірку на правильність вводу даних та вивід відповідного повідомлення.
3. Надається можливість відкрити теоретичний матеріал по темі;
4. Можна перейти до виконання завдань по темі;
5. Вихід з тренажера або повернення на головне меню в будь-який момент.

Інформація в тренажері видається наступним чином: спочатку виводиться коротка довідка, потім практичне завдання.

При правильній відповіді користувач отримує повідомлення «Правильна відповідь», у іншому випадку користувач отримує повідомлення «Неправильна відповідь» та правильну відповідь.

Мету та завдання роботи виконано, створено тренажер для навчання студентів дистанційного навчального курсу «Дискретна математика» темі «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна».

Всі вимоги, описані в постановці задачі, були виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології» галузь знань - 12 «Інформаційні технології» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2011. – 71 с.
2. Цюрюпа В.С. Розробка програмного забезпечення тренажера з теми «Комбінаторні конфігурації» дисципліни «Елементи комбінаторної оптимізації» / В.С. Цюрюпа // Інформатика та системні науки(ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19-21 березня 2015року). – Полтава : ПУЕТ, 2015. – Режим доступу:
<http://dspace.uccu.org.ua/handle/123456789/2509>.
3. Крикля М. П. Розробка алгоритму тренажеру з теми «Графічний метод розв'язування задач лінійного програмування» / М. П. Крикля // Інформатика та системні науки(ІСН-2014): матеріали V Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 13-15 березня 2014року). – Полтава : ПУЕТ, 2015. – С. 163-165. – Режим доступу:
<http://dspace.puet.edu.ua/handle/123456789/1942>.
4. Педоренко С.В. Розробка тренажера з теми "М-метод" дистанційного навчального курсу "Методи оптимізації та дослідження операцій" / С.В. Педоренко, О.О. Ємець // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 16–18 берез. 2017 р.). – Полтава: ПУЕТ, 2017. – Режим доступу:
<http://dspace.puet.edu.ua/handle/123456789/5423>
5. Усольцев С.І. Розробка алгоритму, програми та дослідження тренажеру з теми "Модифікований симплекс-метод" дистанційного навчального курсу "Методи оптимізації та дослідження операцій" / С.І. Усольцев //

Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 309-310. – Режим доступу:

<http://dspace.puet.edu.ua/handle/123456789/2845>

6. Навчальний тренажер з теми «Синтаксичний аналіз» [Електронний ресурс] // Дистанційний курс «Теорія програмування». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=2009>
7. Навчальний тренажер з теми «Верифікація програм та алгоритмічно нерозв'язні проблеми» [Електронний ресурс] // Дистанційний курс «Теорія програмування». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=2009>
8. Про Основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки : Закон України від 9.01.2007 № 537-V [Електронний ресурс]. – Режим доступу : <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=537-16>.
9. Губанова А.О. Методика проведення навчальних занять з фізики у формі тьюторіала / А.О. Губанова // Наукові записки. Серія: Проблеми методики фізико-математичної і технологічної освіти. – Вип. 9. – Ч. 2. – Кіровоград: РВВ КДПУ ім. В. Вінниченка, 2016. – С. 100–107.
10. Дистанційний курс «Дискретна математика (частина 1)» [Електронний ресурс]. – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=2009>
11. Горбатов В.А. Основы дискретной математики: Учеб. пособие. – М.: Высшая школа, 1986. – 311 с.
12. C Sharp [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії.. – Режим доступу: https://uk.wikipedia.org/wiki/C_Sharp
13. Unity (рушій гри) [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії.. – Режим доступу: [https://uk.wikipedia.org/wiki/Unity_\(рушій_гри\)](https://uk.wikipedia.org/wiki/Unity_(рушій_гри))

14. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

ДОДАТОК А. КОД ПРОГРАМИ**ДОДАТОК А**

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;

    public InputField inputedTxt;
    public Text ShowInpTxt;

    public void Next()
    {
        Input.SetActive(false);
        AfterInput.SetActive(true);

        ShowInpTxt.text = inputedTxt.text;
    }
}

public class QuitButton : MonoBehaviour
{
    public void QuitGame()
    {
        Debug.Log ("QUIT!!");
        Application.Quit();
    }
}

void Start()
{
    Button btn = nxtButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}
```

```
void TaskOnClick()  
{  
    Theme1.SetActive(false);  
    Theme2.SetActive(true);  
  
    Debug.Log("You touched this button.");  
}  
  
void Start()  
{  
    Button btn = menuButton.GetComponent<Button>();  
    btn.onClick.AddListener(TaskOnClick);  
}  
  
void TaskOnClick()  
{  
    Theme10.SetActive(false);  
    MainMenu.SetActive(true);  
  
    Debug.Log("You touched this button.");  
}  
  
public void Next()  
{  
    Input.SetActive(false);  
    AfterInput.SetActive(true);  
  
    ShowInpTxt.text = inputTxt.text;  
}  
  
}
```