

Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

ДИПЛОМНА РОБОТА МАГІСТРА

на тему:

«Модифікація генератора завдань пошуку ТДНФ для тестової системи Moodle»

студента 2-го року навчання

спеціальність: 123 «Комп'ютерна інженерія»

Білика Богдана Анатолійовича

Науковий керівник:

асистент кафедри комп'ютерної інженерії

Слюсар Є. А.

До захисту допускаю: Завідувач кафедрою, к. ф.-м. н.,
доцент **Юрій Бойко**

Ухвалено на засіданні кафедри “___” _____ 2023 р., протокол №___

КИЇВ - 2023

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра містить 58 сторінок, 37 рисунків, 3 додатки, 4 таблиці, використано 14 інформаційних джерел.

ГЕНЕРАТОР, ДДНФ, ТДНФ, МДНФ, ЗАВДАННЯ, АЛГОРИТМ КВАЙНА-МАККЛАСКІ, МЕТОД ПЕТРИКА, АЛГОРИТМ СПРОЩЕННЯ З ВИКОРИСТАННЯ ОПЕРАЦІЙ ПОГЛИНАННЯ І СКЛЕЙКИ, MOODLE XML, C#, WPF, MVVM, мінтерм, шаблон проектування “Стратегія”.

Метою роботи є модифікація попередньо розробленого програмного забезпечення для розширення можливостей функціоналу, вирішення критичних проблем, пошук та виправлення проблем графічного інтерфейсу та логіки роботи застосунку.

Об’єктом роботи є процес генерації завдання для імпорту у середовище для тестування Moodle. Предметом роботи є модифікація розробленого раніше програмного забезпечення, яке дозволяє генерувати завдання та зберегти його в необхідному для подальшого імпорту форматі.

Методами дослідження є аналітичні: аналіз існуючих проблем та способів їх вирішення; та практичні: реалізація алгоритмів в кодї, розгортання та налаштування середовища тестування, перевірка правильності роботи завдання. Інструменти розроблення: мова програмування C#, мова розмітки XAML, середовище розробки Microsoft Visual Studio 2022. Програмний продукт може застосовуватись на персональних комп’ютерах з встановленою платформою .Net 4.8.

Робота містить в собі інформацію про функціонал та особливості роботи попередньої версії програмного забезпечення, аналіз та опис відомих проблем. Було визначено список необхідних змін. Даний список було реалізовано у новій версії генератора завдань. Роботу генератора завдань було перевірено в середовищі Moodle.

ЗМІСТ

РЕФЕРАТ	2
СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ФУНКЦІОНАЛЬНОСТІ ІСНУЮЧОГО РІШЕННЯ ДЛЯ ГЕНЕРАЦІЇ ЗАВДАНЬ	8
1.1 Архітектура попередньої версії застосунку	8
1.2 Принцип роботи та функціонал попередньої версії застосунку	9
1.3 Проблема неоднозначності перевірки завдання	10
1.4 Алгоритм мінімізації у наявному генераторі завдань	11
1.5 Розгляд нових алгоритмів для реалізації	12
РОЗДІЛ 2 РОЗРОБКА МОДИФІКАЦІЇ ГЕНЕРАТОРА ЗАВДАНЬ	14
2.1 Список вимог до існуючих рішень	14
2.2 Існуючі рішення: висновки	15
2.3 Алгоритм Квайна-МакКласкі	16
2.4 Метод Петрика	18
2.5 Вирішення проблеми неоднозначності перевірки завдання	19
2.6 Архітектура нової версії застосунку	23
2.7 Використання шаблону “Стратегія”	24
2.8 Зміни в графічному інтерфейсі	26
2.9 Підтримка категорій питань	27
2.10 Опис реалізації модифікованої версії генератора завдань	28
РОЗДІЛ 3 ТЕСТУВАННЯ МОДИФІКОВАНОЇ ВЕРСІЇ ГЕНЕРАТОРА ЗАВДАНЬ З МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ	29
3.1 Тестове середовище та умови тестування	29
3.2 Тестування правильності роботи алгоритму Квайна-МакКласкі	29
3.3 Тестування застосунку	32
ВИСНОВКИ	51
ПЕРЕЛІК ПОСИЛАНЬ	52

ДОДАТКИ

54

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ДНФ - диз'юнктивна нормальна форма

ДДНФ - досконала диз'юнктивна нормальна форма

ТДНФ - тупикова диз'юнктивна нормальна форма

МДНФ - мінімальна диз'юнктивна нормальна форма

Moodle - Modular Object-Oriented Dynamic Learning Environment, навчальна платформа для проведення тестових завдань.

XML - Extensible Markup Language, мова розмітки ієрархічно структурованих даних.

WPF - Windows Presentation Foundation, система для побудови графічних застосунків Windows.

ВСТУП

Підстави для виконання роботи. Для перевірки рівня знань студента з курсу дискретної математики викладачу необхідно проводити різні модульні завдання. Однією з тем проведення модульних завдань є мінімізації булевих функцій. Підготовка до проведення таких модулів займає чимало часу та зусиль. Окрім процесу підготовки до проведення модульної контрольної роботи, обов'язком викладача є і процес перевірки написаних робіт, що в свою чергу займає ще більше часу. До того ж, в залежності в кількості варіантів та різноманітності завдань, складність всього процесу зростає. Для вирішення цієї проблеми раніше було розроблено генератор тестових завдань для системи Moodle. Функціонал цього програмного забезпечення містить в собі підтримку алгоритму мінімізації ДДНФ з послідовним використанням операцій склейок і поглинання та збереження створеного завдання для імпорту до системи Moodle.

Мета й завдання роботи. Метою кваліфікаційної роботи є модифікація існуючого програмного забезпечення шляхом розширення переліку доступних алгоритмів через реалізацію мінімізації ДДНФ за допомогою інших алгоритмів спрощення. Також необхідно розглянути існуючі проблеми в існуючій версії застосунку та знайти до них рішення. Для реалізації наведеної мети поставлено наступні завдання:

- Дослідити принцип роботи різних алгоритмів мінімізації
- Реалізувати алгоритм вибраний алгоритм спрощення у генераторі завдань
- Розглянути проблему неоднозначності перевірки завдань, знайти та реалізувати підходяще рішення

Сфери застосування. Розроблене програмне забезпечення може бути, у подальшому, використане викладачами для підготовки та проведення модульних робіт з мінімізації булевих функцій. Застосунок може генерувати умову для завдання з мінімізації, провести мінімізацію за вибраним алгоритмом, сформувати завдання для тестової системи Moodle у необхідному форматі, яка вже автоматично перевірить правильність виконання роботи студентом.

Об'єкт та методи розробки. Об'єктом розробки є застосунок, здатний виконувати процес мінімізації булевих функцій за допомогою алгоритмів Квайна-МакКласкі та градієнтного спуску, формувати завдання у тестовому форматі та форматі типу коротка відповідь, зберігати сформовані завдання у файлі спеціального XML формату для подальшого імпорту до тестової системи Moodle.

В якості мови програмування була вибрана мова C#, адже оригінальний генератор тестових завдань був написаний на ній. Застосунок реалізований з використанням технології WPF. В якості операційної системи використана Windows 10. Ця система містить вмонтовану підтримку технології .Net Framework 4.8, необхідної для запуску WPF застосунка. Архітектура WPF застосунка використовує шаблони проектування MVVM та Стратегію для вибору типу алгоритму та алгоритму збереження завдань до файлу в залежності від вибраного формату запитання.

РОЗДІЛ 1 АНАЛІЗ ФУНКЦІОНАЛЬНОСТІ ІСНУЮЧОГО РІШЕННЯ ДЛЯ ГЕНЕРАЦІЇ ЗАВДАНЬ

1.1 Архітектура попередньої версії застосунку

Програма реалізована у вигляді WPF застосунку. При розробці попередньої версії застосунку використовувався шаблон проектування[1] MVVM[2]. Шаблон проектування - це програмне архітектурне рішення, яке допомагає вирішити задачі, які регулярно виникають при написанні коду.

При використанні шаблону MVVM застосунок поділяється на 3 незалежні одна від одної частини. Ці частини називаються View, Model, ViewModel.

View - це компонент програми, який відповідає за інтерфейс користувача. Зазвичай, у WPF застосунках View описаний за допомогою мови-розмітки xaml[3]. У генераторі тестових завдань є різні представлення, кожне з яких повністю реалізоване лише з використанням мови-розмітки xaml.

Model - це компонент програми, який відповідає за всю бізнес-логіку програми та зберігає всю інформацію. Всі обрахунки та всі алгоритми виконуються в цьому компоненті. В даному застосунку в моделі розташовані всі класи для зберігання інформації, класи для генерації та мінімізації ДДНФ[4]. Клас для зберігання згенерованих завдань до XML файлу у форматі Moodle xml також реалізований у компоненті Model.

Для взаємодії між цими двома незалежними компонентами у патерні MVVM використовується компонент-прошарок ViewModel. Цей компонент викликає методи, описані в моделі, та зберігає посилання на інформацію, яка буде відображена в графічному інтерфейсі. Інформації з компоненту View передається лише до компоненту ViewModel за

допомогою механізму, реалізованому у WPF, який називається Binding[5]. Для того, щоб викликати будь-який функціонал з Model, View використовує ще один механізм, який підтримується проектами WPF, який називається Command[6].

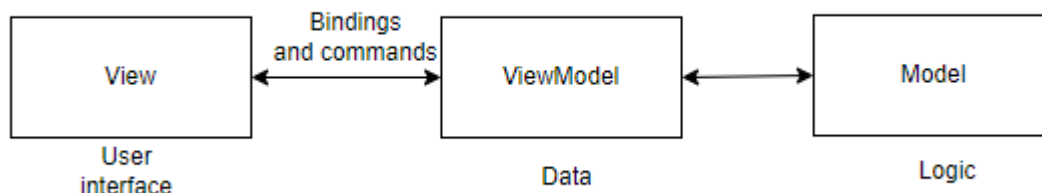


Рисунок 1.1 Узагальнений принцип роботи шаблону MVVM

1.2 Принцип роботи та функціонал попередньої версії застосунку

Програма дозволяє автоматично згенерувати ДДНФ для булевої функції з вибраною кількістю змінних та конститuent 1. Наступним кроком є мінімізація ДДНФ за допомогою алгоритму з послідовним використанням операцій склейок і поглинання, додає згенеровану пару “Запитання - відповідь” до пулу запитань.

Відповідь зберігається у визначеному форматі відповідей Moodle “Short-Answer”[7]. Цей тип запитань дозволяє користувачу, який проходить тест у системі Moodle ввести відповідь у форматі строки. При перевірці цієї відповіді система Moodle парсить посимвольно введену відповідь та звірює її з правильною.

Пул запитань - це список різних пар “Запитання - відповідь”, який використовується для того, щоб користувач міг генерувати декілька варіантів для модулю.

Після того, як була згенерована потрібна кількість завдань, отриманий пул зберігається в окремий xml файл, який у подальшому імпортується до системи Moodle.

Також у застосунку передбачена можливість редагування текстового повідомлення для запитання, яке відображається в завданні перед умовою. В цьому повідомленні користувач може зазначити всю необхідну інформацію для проходження модулю, наприклад назву алгоритма мінімізації, за яким потрібно виконувати завдання, правила оформлення відповіді, кількість балів за правильно виконане завдання, тощо.

Окрім цього, програма реалізує спеціальний модуль, який дозволяє переглянути згенероване завдання у тому вигляді, в якому це завдання буде представлене в тестовій системі Moodle. Цей модуль відкриває нове вікно, яке називається “Вікно перегляду”.

Загалом, весь функціонал можна зобразити наступним чином, якщо взяти до уваги те, що програма підтримує шаблон проектування MVVM:

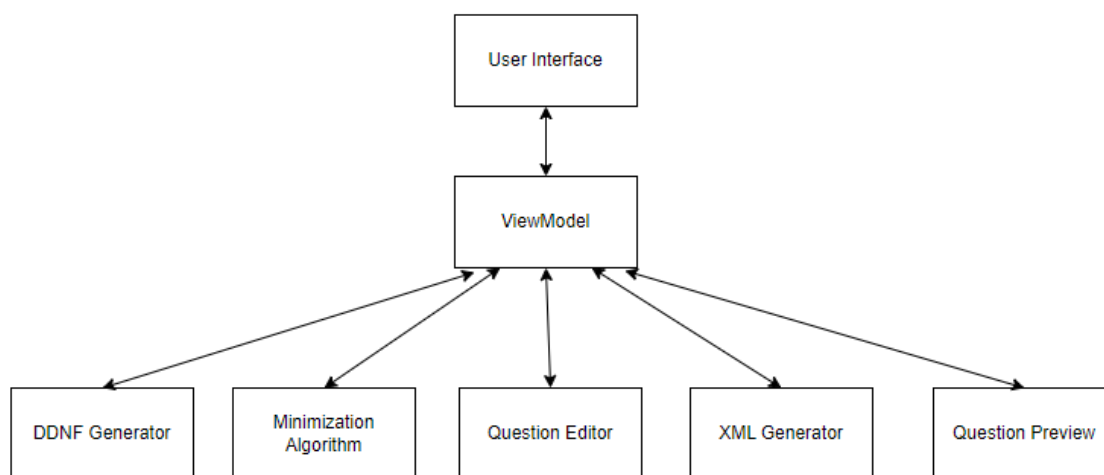


Рисунок 1.2 Архітектура попередньої версії застосунку

1.3 Проблема неоднозначності перевірки завдання

Як вже було зазначено раніше, тип запитання у наявного програмного забезпечення - це вбудований до систему Moodle тип запитання “Short Answer”. Проблема такого типу запитань полягає в тому, що введена відповідь повинна співпадати з правильною з точністю до

символу. В свою чергу це може привести до того випадку, коли студент введе відповідь, в якій елементарні кон'юнкції, або ж конституенти 1 будуть розташовані в іншому порядку, відмінному від того, який вказаний в правильній відповіді. З точки зору дискретної математики така відповідь буде вважатись правильною, адже порядок елементарних кон'юнкцій, або ж конститuent 1 не впливає на значення ДНФ та будь-якої її форми. До того ж через особливості роботи програми, відповідь повинна була містити багато пробілів, якими символи відокремлювалися один від одного, для того, щоб алгоритм міг правильно розпізнати введену інформацію та правильно мінімізувати введену чи згенеровану ДДНФ.

Ця проблема може потенційно створити ситуації, при яких студент буде апелювати до викладача, наполягаючи на те, що з точки зору законів дискретної математики відповідь, введена студентом є правильною. В свою чергу, розгляд таких апеляцій забере додатковий час у викладача, який він економить, користуючись генератором завдань. А отже, така ситуація протиречить самій концепції програми, яка була створена для того, щоб економити час викладачу.

1.4 Алгоритм мінімізації у наявному генераторі завдань

У попередній версії генератора завдань з мінімізації булевих функцій використовувався алгоритм з послідовним використанням операцій склейок і поглинання. Цей алгоритм був реалізований в програмі саме тому, що на той момент в курсі дискретної математики він розглядався як основний аналітичний спосіб мінімізації булевих функцій.

Проте, це алгоритм має декілька особливостей, серед яких те, що результат виконання алгоритму напряму залежить від того, в якій послідовності були записані конституенти одиниці та елементарні кон'юнкції в них при створенні ДДНФ для початку мінімізації. Це

призводить до того, що одна й та сама булева функція, записана в різних формах, після мінімізації за цим алгоритмом видасть різні ТДНФ. Ці ТДНФ будуть формою запису булевої функції, яка виражається за допомогою ДДНФ, проте не всі отримані ТДНФ будуть МДНФ, що у свою чергу не зовсім відповідає концепції мінімізації булевих функцій. Адже якщо після мінімізації була отримана ТДНФ, яка не є МДНФ то це значить що у подальшому на запис цієї булевої функції може бути витрачено більше пам'яті, або ж будова мікросхеми може бути надто ускладненою через те, що доведеться використовувати більше елементів для її побудови.

Саме тому в новому поколінні генератора завдань з мінімізації булевих функцій було вирішено додати підтримку алгоритму мінімізації, який завжди дає однозначний результат. До того ж, підтримка декількох алгоритмів зробить функціонал програми більш гнучким та надасть змогу пристосуватися до змін у навчальній програмі, чи до використання цієї програми у інших навчальних закладах, які не розділяють навчальну програму Київського національного університету імені Тараса Шевченка.

1.5 Розгляд нових алгоритмів для реалізації

1) Алгоритм Квайна-МакКласкі

Серед існуючих алгоритмів мінімізації одним з найбільш розповсюджених є алгоритм Квайна-МакКласкі[8]. Цей алгоритм дає однозначний результат, який завжди є МДНФ, незалежно від послідовності введених мінтернів чи послідовності елементарних кон'юнкцій в них. Даний алгоритм є заміною карт Карно, проте з використанням таблиць для процесу мінімізації. Окрім того, під цей алгоритм містить в собі ще один, адже при виконанні цього алгоритму виникає необхідність мінімізації таблиці простих імплікант, що зручно виконувати за допомогою Методу Петрика[10]. Недоліком цього алгоритму є зростання часу, який

витрачається на мінімізації зі зростанням кількості змінних у булевій функції, проте розуміючи, що дана програма буде використовуватись в навчальних цілях, вірогідність того, що кількість змінних у булевій функції буде більше п'яти дуже низька.

2) Метод Блейка-Порецького[9]

Даний метод дає можливість отримати скорочену ДНФ булевої функції за довільної ДНФ. Принцип методу полягає в тому, щоб до ДНФ були застосовані всі можливі операції неповної склейки, а потім всі можливі операції поглинання. Знаходження мінімальної ДНФ досягається шляхом рекурсивного застосування методу Блейка-Порецького до кожної з отриманих скорочених ДНФ до тих пір, поки не буде отримана ТДНФ. Проблема цього методу полягає в тому, що кількість операцій узагальненого склеювання зростає не лише від кількості змінних в булевій функції, але і від кількості мінтермів, записаних в ДДНФ.

Проаналізувавши 3 популярні аналітичні методи мінімізації булевих функцій, було вирішено що найбільш підходящим алгоритмом для розширення можливостей функціоналу є алгоритм Квайна-МакКласкі. Метод Петрика може видати декілька результатів, що є неприйнятним для завдання тестової системи Moodle, проте всередині алгоритму Квайна-МакКласкі є певні умови до результату, який отримується після спрощення таблиці простих імплікант за методом Петрика. Метод Блейка-Порецького може стати занадто складним при певних умовах, що посилює складність завдання без сильної необхідності.

РОЗДІЛ 2 РОЗРОБКА МОДИФІКАЦІЇ ГЕНЕРАТОРА ЗАВДАНЬ

2.1 Список вимог до існуючих рішень

Перед тим, щоб аналізувати всі можливі існуючі рішення, які можна було б використовувати в роботі, необхідно сформулювати список вимог до них. Проблема неоднозначності перевірки відповіді не може бути вирішена з використанням вже існуючого програмного забезпечення.

1) Безкоштовність.

Оскільки генератор завдань пошуку ТДНФ не є комерційним програмним забезпеченням та розповсюджується абсолютно безкоштовно, то і використане програмне забезпечення для розширення функціоналу повинно бути безкоштовним, адже розробка такого проекту не має якогось стороннього фінансування.

2) Підтримка алгоритму Квайна-МакКласкі

Оскільки модифікація генератора завдань полягає ще і у додаванні підтримки алгоритму Квайна-МакКласкі, то розглянуте програмне забезпечення повинне підтримувати алгоритм Квайна-МакКласкі.

3) Можливість генерації відповідей у форматах, призначених для імпорту у систему Moodle

Оскільки кінцева мета використання генератора завдань є економія часу, то для цього згенеровані відповіді повинні бути певним чином імпортовані до системи Moodle. Існуюча версія генератора завдань підтримує формат Moodle Xml, тому такий формат навіть буде найбільш пріоритетним.

4) Сумісність з операційною системою Windows

Оскільки генератор завдань з пошуку ТДНФ це WPF проект, тобто застосунок, який можна запустити лише на операційних системах сімейства Windows, то і існуюче рішення повинно бути сумісне з

сімейством операційних систем Windows. В свою чергу, ця умова зобов'язує існуюче рішення або ж бути написаним виключно під систему Windows, або ж бути онлайн ресурсом.

5) Правильний формат умови

Окрім відповіді, отриманої в результаті мінімізації, умова завдання, тобто ДДНФ повинна бути записана в такому вигляді, щоб користувач розумів, що перед ним саме ДДНФ, а не набір чисел, який представляє собою набір індексів мінтермів булевої функції. Мінтерми повинні бути записані у такому форматі, де видно весь набір всіх елементарних кон'юнкцій,

б) Використання програмного забезпечення повинно бути легальним

Правила користування існуючого рішення та ліцензія, якщо вони наявні не повинні конфліктувати з існуючими умовами та реаліями використання генератора завдань

2.2 Існуючі рішення: висновки

Пошук в мережі інтернет не дав жодного прикладу програмного забезпечення, яке б відповідало всім умовам, зазначеним до існуючих рішень. Підтримка лише частини умов не дає змогу використовувати програмне забезпечення від інших розробників у цьому проекті, адже кожна з умов є критичною для правильності роботи програми та відповідності концепції програмного забезпечення, яке дозволило б зекономити час викладачу.

Реалізація алгоритму в коді Квайна-МакКласкі є неодноразово реалізованою задачею, проте через наявний список умов алгоритм необхідно реалізувати власноруч, без використання будь-яких існуючих рішень.

Перечисливши список вимог та дійшов висновку, що проблема неоднозначності перевірки завдання не може бути вирішена з використанням іншого програмного забезпечення можна дійти до висновку, що всі поставлені задачі будуть реалізовані в коді.

2.3 Алгоритм Квайна-МакКласкі

Алгоритм Квайна-МакКласкі дозволяє отримати мінімальну ДНФ, отримуючи на вхід ДДНФ. Послідовність дій при виконанні цього алгоритму наступна

- 1) Усі мінтерми функції впорядковуються у порядку зростання одиниць у мінтермі.
- 2) Список розбивається на підмножини(групи) мінтермів однакової ваги. Кількість одиниць в кожній мінтермі групи є індексом групи i .
- 3) Проводяться всі можливі операції склейок між елементами сусідніх груп. Елементи, які не приймають участі в склейках є простими імплікантами та заносяться до окремого списку простих імплікант.
- 4) Якщо після операцій склейок отриманий список не порожній, алгоритм повертається на крок 2. У тому випадку, якщо список порожній, або такий, де не можна проводити більше операцію склейки, всі наявні елементи отриманого списку заносяться до списку простих імплікант.
- 5) Після того, як список простих імплікант сформований, з нього формується таблиця простих імплікант, яка мінімізується за допомогою метода Петрика. Ця мінімізація необхідна у тому випадку, коли покриття булевої функції, отримане внаслідок спрощення, можна скоротити до більш простої форми запису.

Внаслідок виконання алгоритму, після мінімізації таблиці простих імплікант, утворюється ДНФ. ДНФ, утворена з таблиці простих імплікант,

отриманого після виконання алгоритму Квайна-МакКласкі та методу Петрика є мінімальною.

Блок схема алгоритму Квайна-МакКласкі виглядає таким чином:

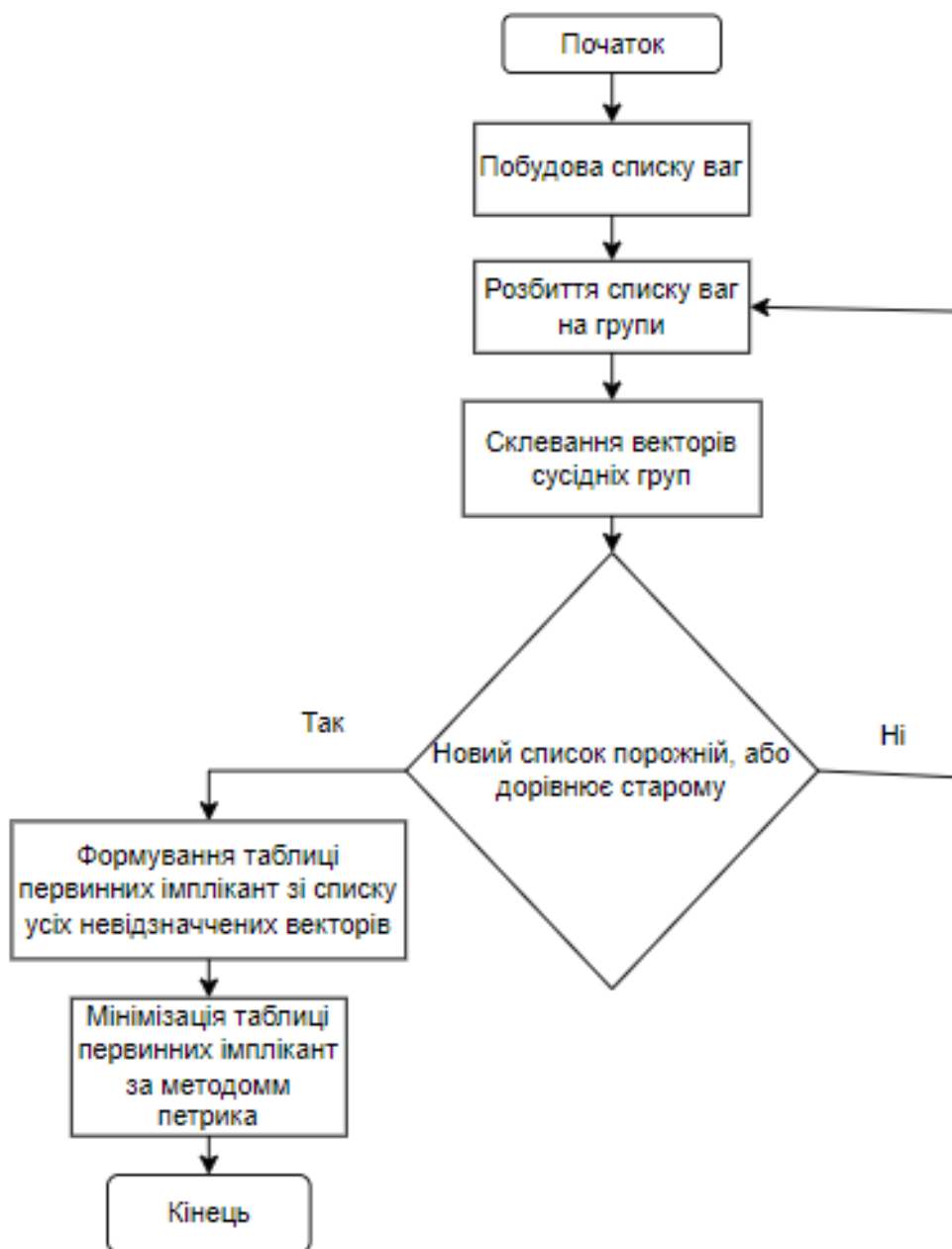


Рисунок 2.3 Блок схема алгоритму Квайна-МакКласкі

2.4 Метод Петрика

Метод Петрика - це алгоритм для пошуку всіх МДНФ, використовуючи таблицю простих імплікант.

Метод Петрика є частиною алгоритму Квайна-МакКласкі, адже при виконанні цього алгоритму виникає таблиця простих імплікант, яку потенційно можна спростити, проте може використовуватись і окремо від нього.

Послідовність кроків для виконання методу Петрика полягає наступним чином:

- 1) Мінімізувати таблицю простих імплікант. Для цього необхідно викреслити з неї рядки, які відповідають необхідним, або центральним імплікантам, та терми, що їм відповідають, тобто викреслити стовпці(терми), які позначені в таблиці лише 1 раз
- 2) Позначити кожну з імплікант довільною змінною
- 3) Створити булеву функцію, яка буде істиною лишу у тому випадку, коли вона покриває всі стовпці таблиці простих імплікант.
- 4) Спростити цю булеву функцію, використовуючи операції операції поглинання та склеювання.
- 5) Вибрати диз'юнкцію з найменшим числом змінних всередині після мінімізації. Кожна з утворених диз'юнкції представляє собою одне з рішень первинної булевої функції, тобто тої, з якої було сформовано таблицю простих імплікант
- б) Розшифрувати отриману диз'юнкцію за допомогою мінтермів, з яких її було сформовано. Якщо таких диз'юнкцій декілька, то вибрати ту, яка розшифровується з мінімальною кількістю мінтермів.

Блок схема методу Петрика виглядає наступним чином:

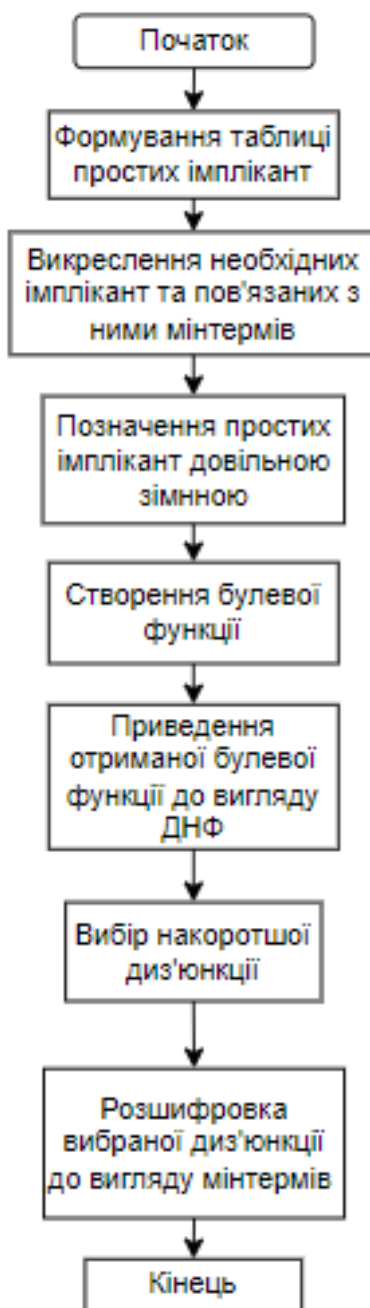


Рисунок 2.4 Метод Петрика

2.5 Вирішення проблеми неоднозначності перевірки завдання

Проблема неоднозначності перевірки відповіді полягає в тому, що студент може отримати можливість апелювати щодо оцінки. При цьому може правильно вирішити завдання, проте неправильно його оформити.

Така модульна робота може викликати додаткові втрати часу у викладача, а основна задача генератора тестових завдань його економити.

Для вирішення цієї проблеми було прийнято рішення реалізувати підтримку в генераторі тестових завдань такого типу запитань, де коректність вводу користувача не може бути оскаржена в подальшому.

Найбільш підходящим типом запитань для такого випадку буде запитання тестового формату[11], тобто запитання, де користувачу пропонують вибрати правильну відповідь серед наведених.

Рішення підтримувати завдання у тестовому форматі породжує нову проблему: необхідність генерувати не лише правильний варіант відповіді, але декілька точно неправильних. До того ж неправильна відповідь повинна виглядати так само, як і правильна, тобто бути схожою на ТДНФ, а у найкращому випадку бути ТДНФ.

Якщо проводить якісь операції над правильною відповіддю, то можна скоро дійти до висновку, що будь-яка зміна символу в правильній відповіді може потенційно створити декілька ситуацій, які скомпрометують новий неправильний варіант. До таких випадків можна віднести ситуацію, де після модифікації правильної ТДНФ студент бачить, що в неправильній ТДНФ можна провести операцію склейки чи поглинання, або ж після такої модифікації студент може побачити, якісь артефакти, помилки в оформленні варіанту відповіді. В такому випадку це полегшить студентіві завдання, адже вибірка потенційно правильних відповідей буде зменшуватись.

Вирішення цієї проблеми полягає в тому, що генератор тестових завдань буде змінювати не відповідь, отриману в результаті мінімізації, а умову, яка їй передує, тобто ДДНФ.

Знаючи, що різні булеві функції мають різні ДДНФ, можна дійти до висновку, що незначна зміна ДДНФ та проведення мінімізації над нею дасть ще одну ТДНФ, відмінну від початкової, тої що була отримана

внаслідок мінімізації початкової ДДНФ. Така відповідь буде сприйматись, як справжня, адже це буде справжньою ТДНФ, лише для булевої функції, відмінної від тієї, що задана в умові.

Алгоритм створення тестового завдання з генерацією неправильних відповідей виглядає таким чином

1) Початок

На вхід до цього алгоритму приходять ДДНФ та мінімізована правильна ТДНФ.

2) Отримання списку всіх змінних

Генератор завдань булевих функцій підтримує ввід ДДНФ з клавіатури, а отже змінні булевих функцій можуть бути будь-якими літерами латинського алфавіту або кирилиці. Для того, що згенерувати ДДНФ, подібну до тієї, що прийшла на вхід необхідно дізнатись список всіх змінних булевої функції

3) Модифікація ДДНФ

Модифікація ДДНФ відбувається шляхом інвертування випадково обраної змінної в випадково вибраному мінтермі. В такому випадку, нова модифікована ДДНФ представляє вже зовсім іншу булеву функцію.

4) Мінімізація нової ДДНФ за вибраним алгоритмом

Алгоритм повинен бути таким самим, як і той, яким було мінімізовано початкову ДДНФ

5) Додавання результату до списку відповідей

Отримана ТДНФ додається до списку неправильних відповідей

б) Перевірка кількості неправильних відповідей

У тому випадку, якщо кількість неправильних ТДНФ у списку менша 3, то алгоритм повертається на крок 2. У іншому випадку, алгоритм завершує виконання.

Загальна блок-схема алгоритма створення неправильних ТДНФ для використання у завданні тестового типу виглядає наступним чином:

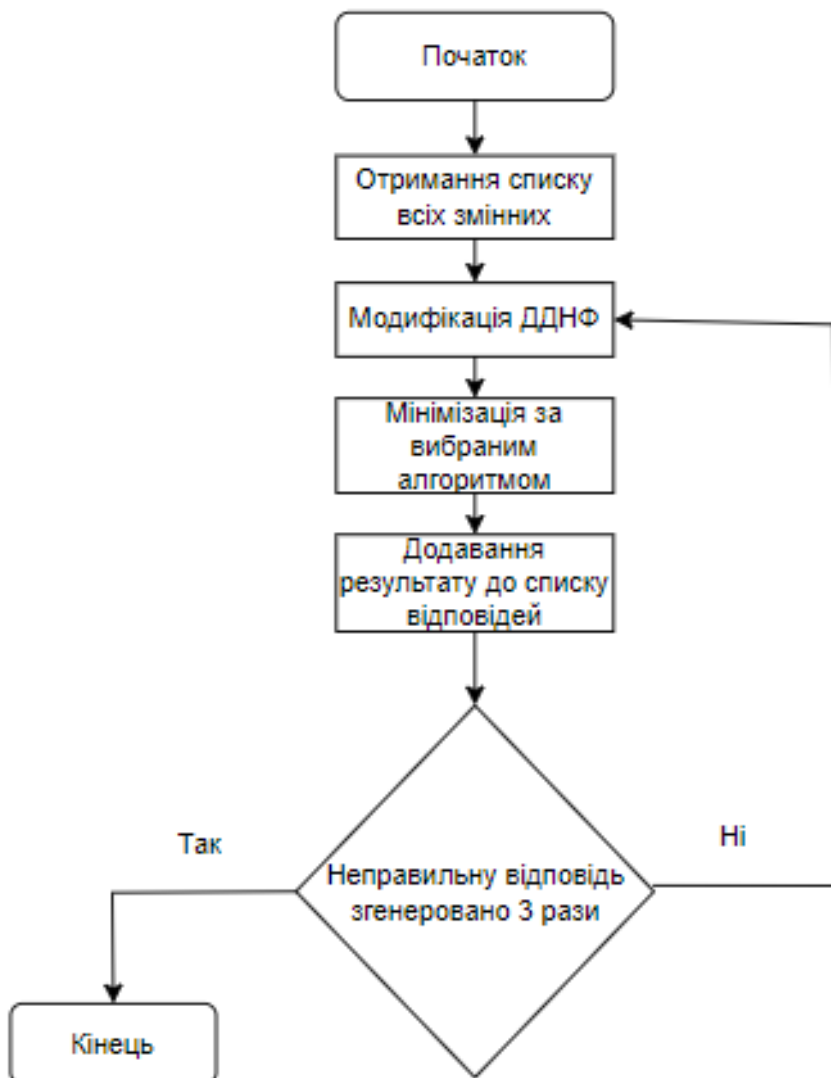


Рисунок 2.5 Блок-схема алгоритму створення неправильних ТДНФ для подальшого використання у завданнях тестового типу

Окрім того, що запитання тестового типу потребують генерацію неправильних відповідей до них, такі запитання ще потребують зовсім іншого алгоритму збереження таких питань у файл формату Moodle Xml, адже кожен тип запитань вимагає особливого підходу до збереження у Xml формат. Це призвело до того, що реалізації алгоритму збереження пулу запитань до Xml файлу відрізняються, в залежності від того, який саме тип запитання користувач генератора завдань намагається зберегти.

2.6 Архітектура нової версії застосунку

Архітектура нової версії застосунку мало чим відрізняється від архітектури існуючої версії, адже завдяки принципу модульності є можливість змінити окремі компоненти програми, не сильно впливаючи на роботи всіх інших. Основні зміни - це можливість підтримки різних алгоритмів мінімізації, можливість підтримки збереження різних файлів формату Moodle Xml для збереження різних типів запитань. Також з'явився новий модуль, який відповідає за генерацію додаткових ТДНФ для створення неправильних відповідей у тестових завданнях.

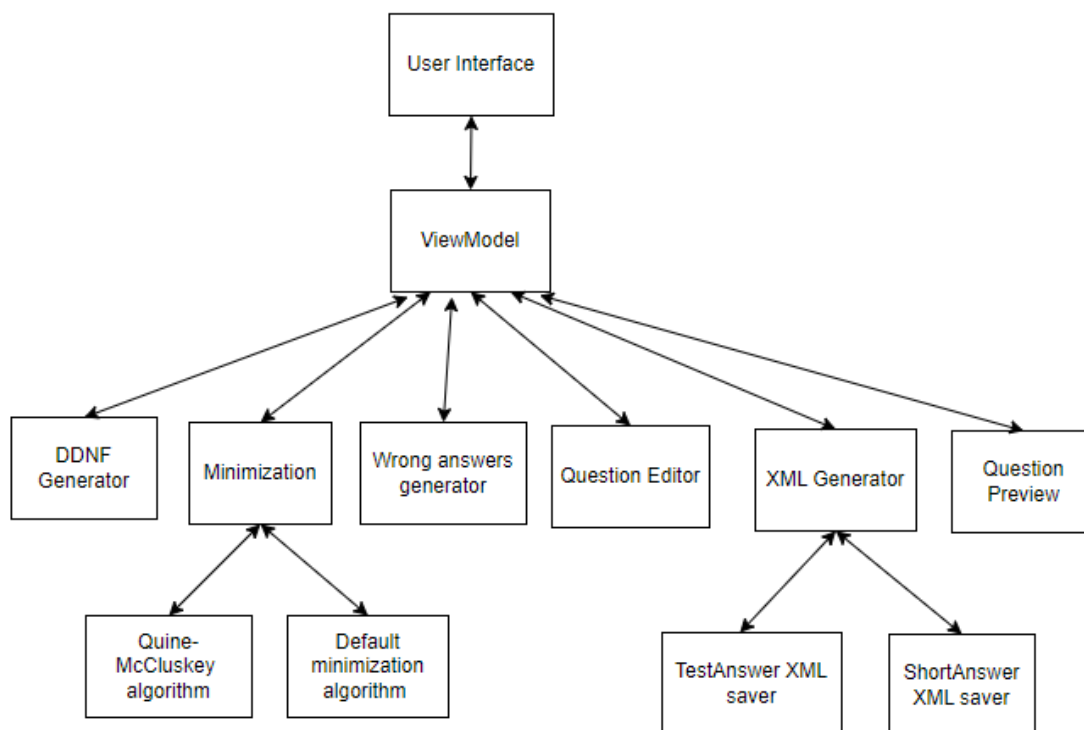


Рисунок 2.6 Архітектура нової версії застосунку

Схематичне зображення архітектури нової версії застосунку дозволяє побачити, що 2 модулі, які відповідають за мінімізацію та генерацію файлів формату Moodle Xml мають по 2 реалізації. Отже, можна

дійти висновку, що в архітектурі програми присутня конструкція, яка повторюється декілька разів, та може бути узагальнена якимось іншим правилом. Цю конструкцію реалізовано з використанням шаблону проектування “Стратегія”[12].

2.7 Використання шаблону “Стратегія”

Як вже згадувалось раніше, шаблони проектування - це архітектурні конструкції в коді, які дозволяють вирішувати проблеми, які часто виникають при розробці програмного забезпечення.

При модифікації генератора завдань виникла ситуація, коли одна частина програми, яка відповідає за певний функціонал, може підтримувати 2 і більше алгоритмів, які реалізують цей функціонал. Ця ситуація ідеально підходить під опис шаблону проектування “Стратегія”.

Шаблон “Стратегія” можна зобразити на UML схемі таким чином:

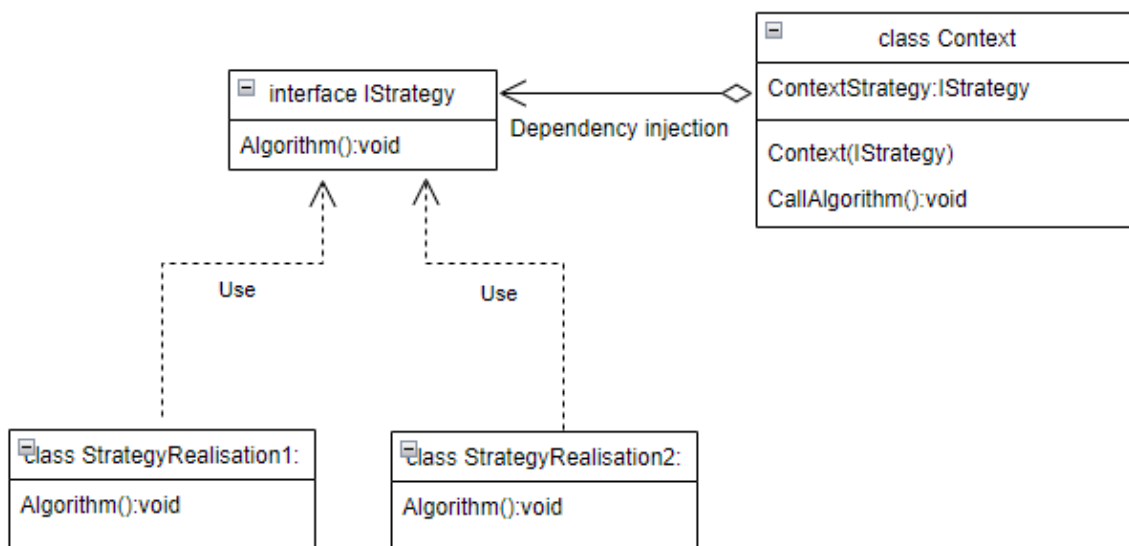


Рисунок 2.7.1 UML діаграма шаблону “Стратегія”

В генераторі завдань є одразу 2 модулі, які реалізують шаблон проектування “Стратегія”. Першим таким модулем є модуль, що відповідає

за обрахунок ТДНФ введеної ДДНФ булевої функції. Оскільки в програмі реалізовано 2 різні алгоритми, які обидва приймають на вхід строку, що представляє ДДНФ, парсять цю строку та мінімізують відповідно до правил конкретного алгоритму, то мінімізацію булевих функцій було представлено у вигляді шаблону “Стратегія”. Лістинг реалізації класу AlgorithmsContext наведено у додатку А.

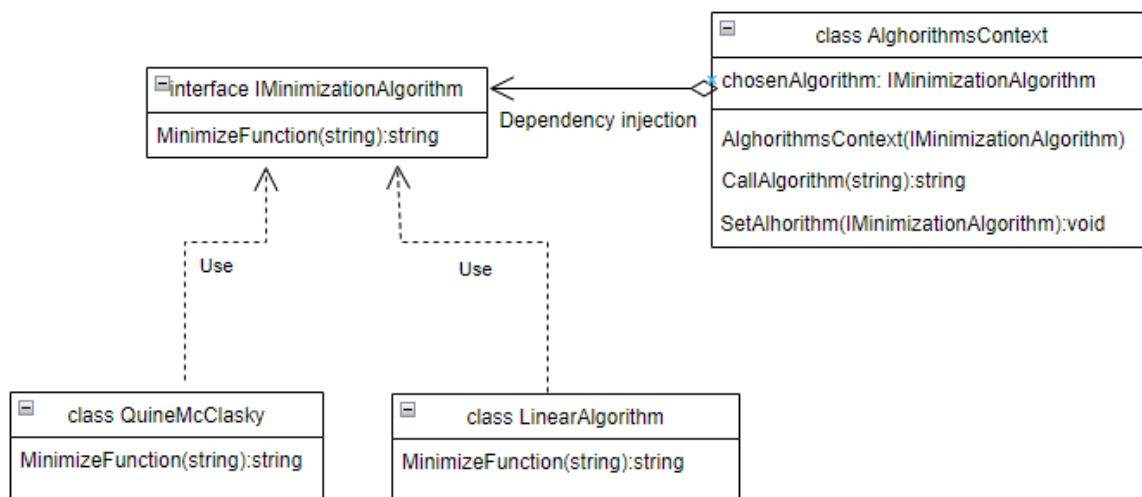


Рисунок 2.7.2 UML діаграма реалізації шаблону “Стратегія” для модулю мінімізації булевих функцій

Другим прикладом імплементації шаблону проектування в модифікації генератора завдань є модуль, який відповідає за генерацію файлів у форматі Moodle Xml. Основна проблема полягала в тому, що різні типи питань потребують різної структури файлу формату Moodle Xml. Тому в цьому модуль також була реалізований шаблон “Стратегія”, аналогічно до модулю мінімізації. Лістинг коду, який відповідає реалізації класу XmlSaverContext наведена в додатку Б

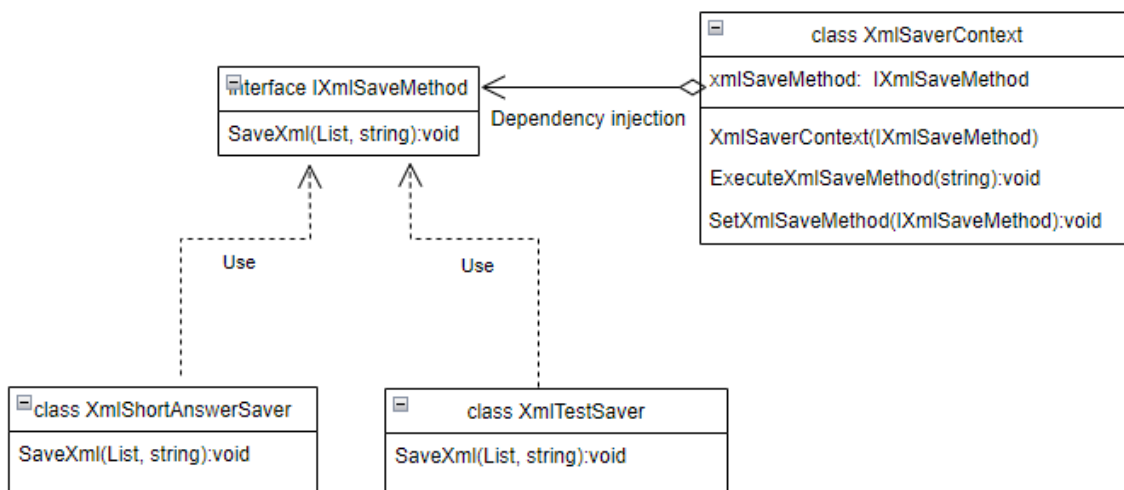


Рисунок 2.7.3 UML діаграма реалізації шаблону “Стратегія” для модулю запису запитань до файлів

2.8 Зміни в графічному інтерфейсі

Одним з головних недоліків наявної версії програми було позначення операції диз’юнкції. Однією з загальноприйнятих форм позначення операції диз’юнкції є знак: v

Даний символ відсутній на звичайній клавіатурі сучасного користувача. Також такий символ не може бути згенерований автоматично, при генерації довільної ДДНФ, адже генератор завдань підтримує ще і ввід з клавіатури. Тому в попередній версії інтерфейсу користувача для позначення операції диз’юнкції використовувалась латинська велика літера V.

Використання такої літери, як символу диз’юнкції може збентежити користувача, адже зазвичай аргументи булевої функції теж позначаються літерами латинського алфавіту. До того ж, це унеможлиблює використання цієї літери як аргумента булевої функції, хоча при вводі з клавіатури підтримка латинських літер верхнього регістру зберігається.

Для вирішення цієї проблеми у модифікованій версії генератора тестових завдань для позначення операції диз'юнкції використовується символ +. Це обґрунтовується тим, що таку операцію ще часто називають “логічною сумою”. При цьому позначення операції кон'юнкції програмою ігнорується, адже таку операцію ще називають “логічним добутком”, а деякі математики ігнорують написання знаку добутку в формулах.

Для операції заперечення використовується символ !, адже такий символ зустрічається в різних мовах програмування як символ логічного заперечення.

Загалом при змінах в інтерфейсі користувача згенерована ДДНФ виглядає таким чином:

ДДНФ:
$$\bar{x} y z + \bar{x} !y z + x !y !z + \bar{x} y !z + \bar{x} !y !z$$

Рисунок 2.8 Оновлені позначення логічних операцій

2.9 Підтримка категорій питань

Система Moodle дозволяє групувати питання, які були створені в курсі, чи імпортовані до нього. Такі групи називаються категоріями[13]. Для того, щоб була можливість правильно відфільтрувати завдання в курсі, після їх імпортування, генератор тестових завдань підтримує категорії запитань, автоматично додаючи ці категорії при створенні файлу формату Moodle Xml.

Генератор тестових завдань підтримує всього 2 категорії в залежності від алгоритму мінімізації, щоб адміністратору курсу було легше створювати модульне завдання. Оскільки генератор завдань підтримує лише одночасне збереження завдань одного типу запитань, тобто лише тестові завдання, або лише завдання з короткою формою відповіді, то при генерації завдання в категорії вказується лише назва

алгоритму мінімізації. Підтримка категорій за типом запитання не підтримується, адже в системі Moodle в банку запитань можна і так побачити тип кожного запитання та легко відрізнити їх один від одного. Як приклад, у додатку В наведено лістинг коду, який реалізує генерацію завдань у тестовому форматі.

```
<question type="category">
  <category>
    <text>$course$/top/QuineMcClasky</text>
  </category>
</question>
```

Рисунок 2.9.1 Категорія запитань для запитань на мінімізацію за допомогою алгоритму Квайна-МакКласкі

```
<question type="category">
  <category>
    <text>$course$/top/LinearMinimization</text>
  </category>
</question>
```

Рисунок 2.9.2 Категорія запитань для запитань на мінімізацію за допомогою алгоритму з використанням операцій склейок і поглинання

2.10 Опис реалізації модифікованої версії генератора завдань

Модифікований генератор тестових завдань містить в собі 3 інтерфейси, 26 класів, серед яких 5 класів відповідають за компонент представлення та містять в собі окрім автоматично згенерованого коду, розмітку, написану на мові XAML. 3 класи утворюють компонент ViewModel, наслідуючи абстрактний клас, який реалізує інтерфейс INotifyPropertyChanged, для того, щоб зміни у ViewModel були відображені в представленні. Застосунок реалізує двічі шаблон “Стратегію”: для обчислення мінімізованої ДНФ та для створення XML файлів.

Загалом реалізація застосунку складає більше 2500 рядків коду.

РОЗДІЛ 3 ТЕСТУВАННЯ МОДИФІКОВАНОЇ ВЕРСІЇ ГЕНЕРАТОРА ЗАВДАНЬ З МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ

3.1 Тестове середовище та умови тестування

Операційною системою, придатною для тестування було обрано Windows 10, адже ця система підтримувалась попередньою версією програми. До того ж, застосунок написаний з використанням платформи .Net Framework 4.8, а ця платформа підтримується операційною системою Windows 10. Для перевірки правильності роботи необхідно створити завдання у тестовій системі Moodle з повноцінним використанням генератора тестових завдань з мінімізації булевих функцій. В якості емулятора середовища Moodle вибрано відкриту безкоштовну демо версію платформи Moodle Sandbox[14] версії 4.1. Згенеровані завдання повинні містити в собі поєднання нового та старого функціоналу, щоб переконатися, що нововведення не завадили роботі існуючих модулів.

3.2 Тестування правильності роботи алгоритму Квайна-МакКласкі

Для того, щоб впевнитись в правильності роботи реалізованого в генераторі завдань алгоритму Квайна-МакКласкі необхідно одну й ту саму булеву функцію мінімізувати вручну, покроково виконуючи кожну дію алгоритму, та мінімізувати цю саму функцію, використовуючи генератор тестових завдань.

Нехай, задано булеву функцію з трьох змінних x , y , z , яка представлена у вигляді ДДНФ:

$$\square \square \square \vee \square \square \square \vee \square \square \square \vee \square \square \square \vee \square \square \square$$

В лівій колонці - порядковий номер мінтерму в таблиці істинності цієї булевої функції. В останній колонці - умовні позначення, де знак \rightarrow означає що імпліканту можна спростити а знак \square означає що мінтерм спростити у подальшому неможливо. Символ $*$ у таблиці, означає що дана зміну мінтерму була спрощена в процесі виконання алгоритму, а отже, її значення не впливає на результат булевої функції в цьому мінтермі.

Список мінтермів до початку спрощення:

N	x	y	z	Результат
0	0	0	0	\rightarrow
2	0	1	0	\rightarrow
3	0	1	1	\rightarrow
4	1	0	0	\rightarrow
5	1	0	1	\rightarrow

Перше коло спрощення:

В першій колонці сказано, номери яких мінтермів приймали участь в спрощенні.

N	x	y	z	Результат
0,2	0	*	0	\square
0,4	*	1	0	\square
2,3	0	1	*	\square
4,5	1	0	*	\square

Всі мінтерми спростити далі неможливо, а отже необхідно переходить до другого етапу виконання алгоритму - до побудови та мінімізації таблиці простих імплікант.

Заповнюючи таблицю простих імплікант необхідно відмити для кожного спрощеного мінтерму стовпчик з номерами індексів мінтермів, з яких було утворено цей мінтерм.

N	x	y	z	0	2	3	4	5
0,2	0	*	0	<input type="checkbox"/>	<input type="checkbox"/>			
0,4	*	0	0	<input type="checkbox"/>			<input type="checkbox"/>	
2,3	0	1	*		<input type="checkbox"/>	<input type="checkbox"/>		
4,5	1	0	*				<input type="checkbox"/>	<input type="checkbox"/>


Після заповнення таблиці простих імплікант можна помітити, що в стовпцях під номерами 3 і 5 лише по одному знаку . Це значить, що мінтерми (2,3) та (4,5) є центральними імплікантами а отже ці рядки можна з таблиці викреслити. також з таблиці можна викресли стовпці 2, 3, 4 і 5, адже вони вже покрити центральними імплікантами. Після викреслення зайвих рядків та стовпців таблиця простих імплікант набуває наступного вигляду:

N	x	y	z	0
0,2	0	*	0	<input type="checkbox"/>
0,4	*	0	0	<input type="checkbox"/>

З таблиці зрозуміло, що обидва мінтерми взаємозамінні, і оскільки кількість змінних в них рівна, по 2 в кожній, то з таблиці вилуємо перший мінтерм (0,2). А отже МДНФ булевої функції складається з мінтермів (2,3) (4,5) і (0,2). Розшифровуючи цю форму запису за допомогою аргументів булевої функції, з яких складається кожен мінтерм отримуємо наступну МДНФ:

$$\underline{x} \underline{y} \vee \underline{x} \underline{z} \vee \underline{x} \underline{y} \underline{z}$$

Тепер переіримо цю ж саму ДДНФ за допомогою генератора завдань, вибравши при цьому алгоритм мінімізації Квайна-МакКласкі

 Генератор завдань

Визначити повідомлення: Кількість змінних:

ДДНФ:

Алгоритм: Тип відповіді:

Результат:

Рисунок 3.2 Результат мінімізації

З отриманого результату видно, що відповіді співпадають, адже порядок запису імплікант не впливає на значення булевої функції, а отже генератор завдань правильно реалізує алгоритм Квайна-МакКласкі.

3.3 Тестування застосунку

Перевірка повного функціоналу застосунку потребує об'ємний тест, який покриває всі можливі варіанти використання застосунку. Це стосується і генерації різних типів запитань і правильна роботі інтерфейсу

користувача і правильно логіка роботи програми, відповідно до впровадженої архітектури.

Для того, щоб почати перевірки роботи, необхідно запустити застосунок.

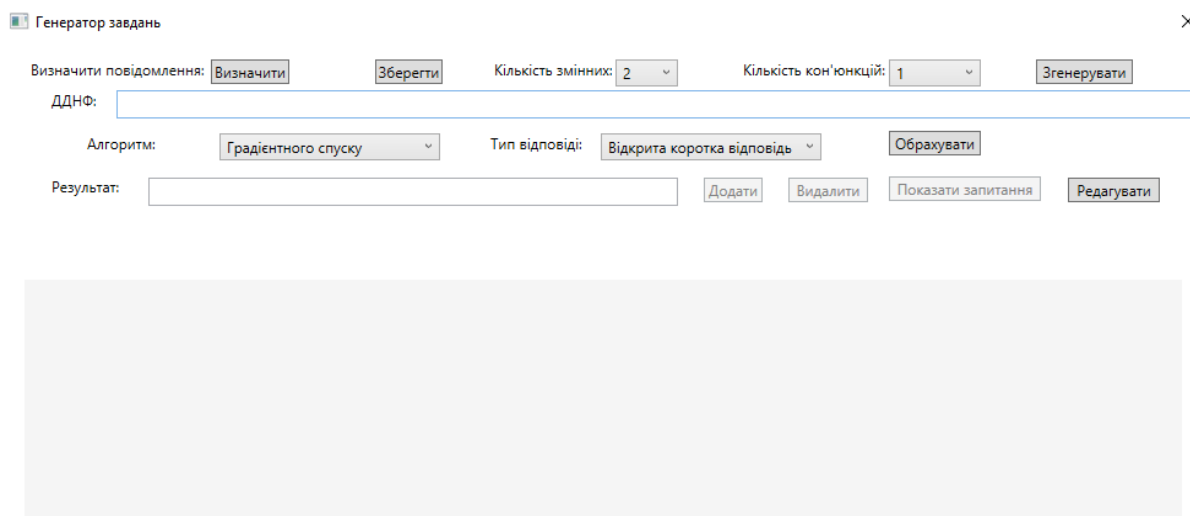


Рисунок 3.3.1 Головне вікно генератора завдань

Для того, щоб мінімізувати функцію, необхідно ввести ДДНФ, яка буде представляти умову завдання. Генератор тестових завдань підтримує ввід ДДНФ з клавіатури, проте для того, щоб пришвидшити процес вводу, застосунок може сам згенерувати ДДНФ з використанням псевдовипадкової генерації. Для цього необхідно вибрати кількість параметрів булевої функції та кількість мінтермів у відповідних комбінованих списках “Кількість змінних” та “Кількість кон’юнкцій”. Нехай перша булева функція буде визначатися як булева функція з чотирьох аргументів та семи мінтермів. Після вибору параметрів, що визначають булеву функцію, необхідно натиснути кнопку “Згенерувати”. Результат автоматичної генерації ДДНФ з’явиться у текстовому полі “ДДНФ”.

Рисунок 3.3.2 Згенерована ДДНФ

Наступним кроком буде вибір алгоритму мінімізації та типу запитання. Нехай алгоритмом спрощення буде алгоритм Квайна-МакКласкі та обраним типом запитання буде тестове завдання. Для цього необхідно вибрати відповідні значення в відповідних комбінованих списках “Алгоритм” та “Тип відповіді”. Для обчислення МДНФ необхідно натиснути кнопку “Обрахувати”. Після цього, результат вибраного алгоритму мінімізації з’явиться у полі “Результат”. Це поле доступне тільки для перегляду, його неможливо модифікувати з клавіатури.

Генератор завдань ×

Визначити повідомлення: Кількість змінних: Кількість кон'юнкцій:

ДДНФ:

Алгоритм: Тип відповіді:

Результат:

Рисунок 3.3.3 Мінімізована булева функція

Після того, як результат мінімізації був отриманий, отриманий результат можна занести до списку запитань, який буде збережений у вихідний файл. В залежності від обраного раніше типу запитань, вигляд списку буде відрізнятися. Наприклад, при виборі тестового завдання, у списку запитань будуть відображені: умова (ДДНФ), список варіантів відповідей з уже позначеним правильним варіантом при цьому змінити правильний варіант неможливо. Варіантів відповідей завжди 4. Для того, щоб додати останнє згенероване завдання до пулу запитань необхідно натиснути кнопку “Додати”.

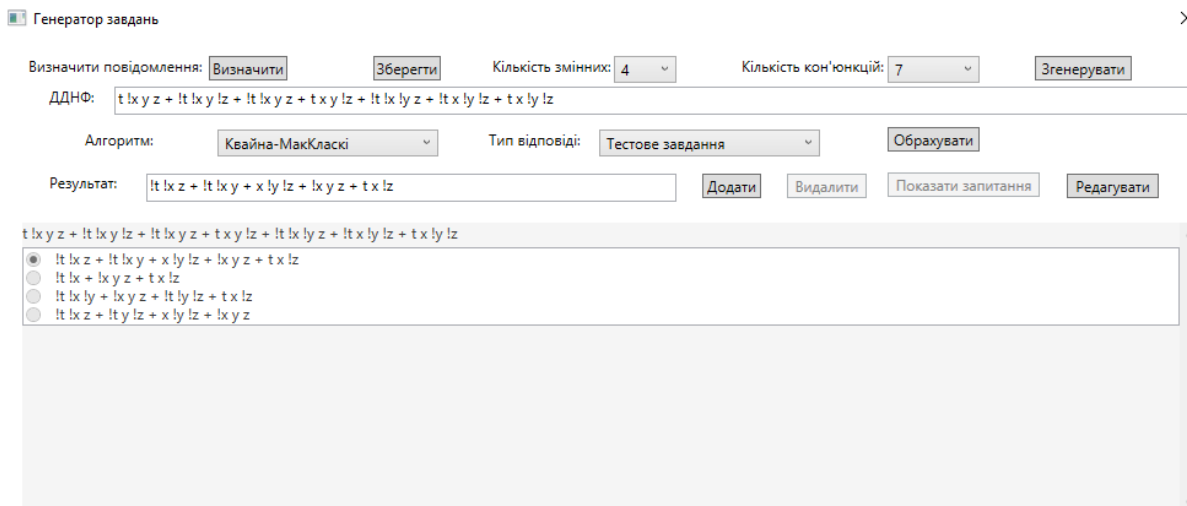


Рисунок 3.3.4 Додане завдання до списку запитань.

Для того, щоб завдання було більш інформативним, необхідно додати до нього певний опис. Для цього необхідно вибрати завдання, натиснувши на нього у списку завдань, та натиснути кнопку “Визначити”. Після цього з’явиться нове вікно з великим полем для вводу тексту. Введене повідомлення прикріплюється абсолютно до всіх згенерованих завдань, адже це дасть користувачу можливість зекономити час на тому, щоб не вводити один і той самий текст для різних варіантів.

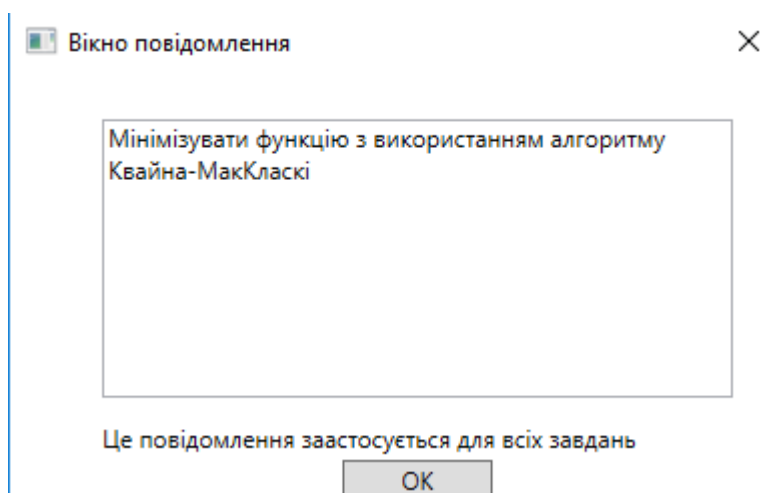


Рисунок 3.3.5 Вікно для введення додаткових повідомлень

Після того, як повідомлення було введено, Отримане завдання можна переглянути у тому вигляді, в якому воно предстане перед студентом у тестовій системі Moodle після імпорту. Зробити це необхідно для того, щоб була можливість перевірити правильність оформлення завдання перед остаточною генерацією файлу. Для цього необхідно натиснути кнопку “Показати запитання”. Відкрите нове вікно надає користувачеві функціонал попереднього перегляду для вибраного запитання.

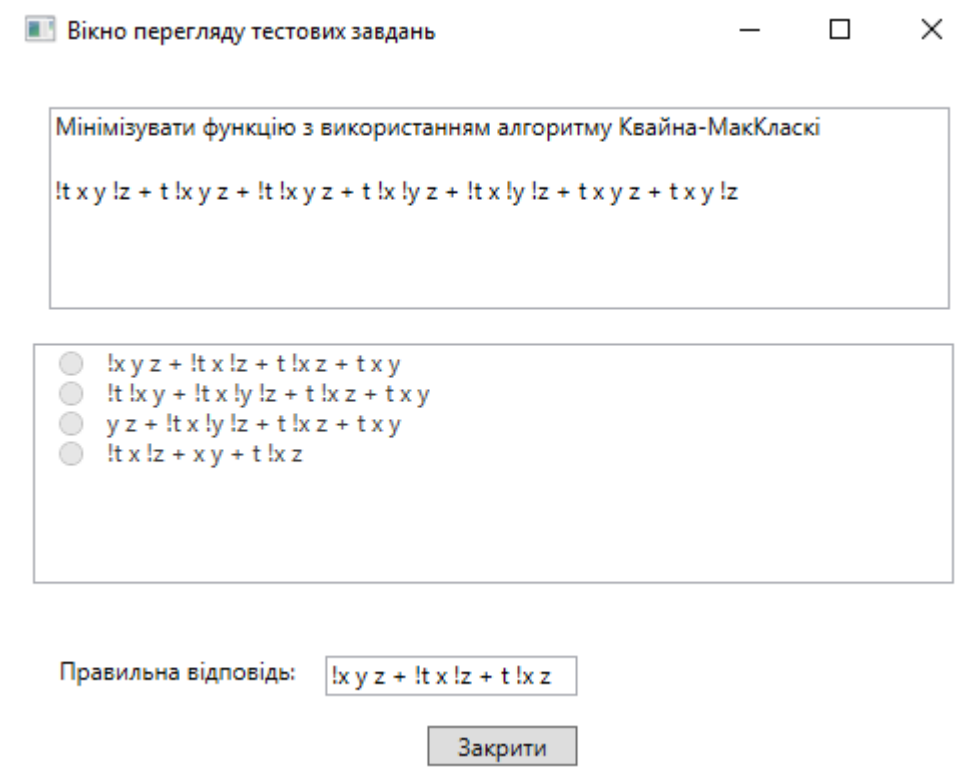


Рисунок 3.3.6 Вікно попереднього перегляду

Додамо ще раз одне і те саме завдання. Для цього необхідно натиснути кнопку “Додати” знову. Після цього в списку завдань буде 2 завдання з однаковою умовою та однаковим набором варіантів.

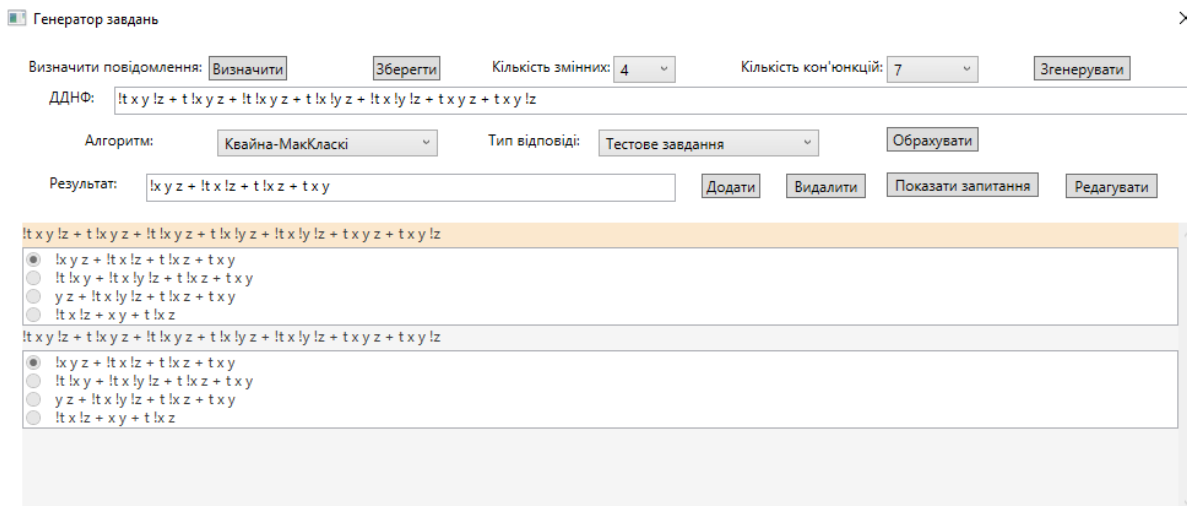


Рисунок 3.3.7 Згенеровано два однакові завдання

Після цього, виберемо друге завдання, натиснувши на нього та натиснемо кнопку “Редагувати”. У новому відкритому вікні, є можливість редагувати неправильні варіанти відповідей для вибраного запитання. Можливості редагувати правильну відповідь не існує. Можливість редагування тестових відповідей підтримується лише для запитань тестового виду, адже запитання з короткою відповіддю неможливо редагувати: там немає нічого, окрім умови та відповіді.

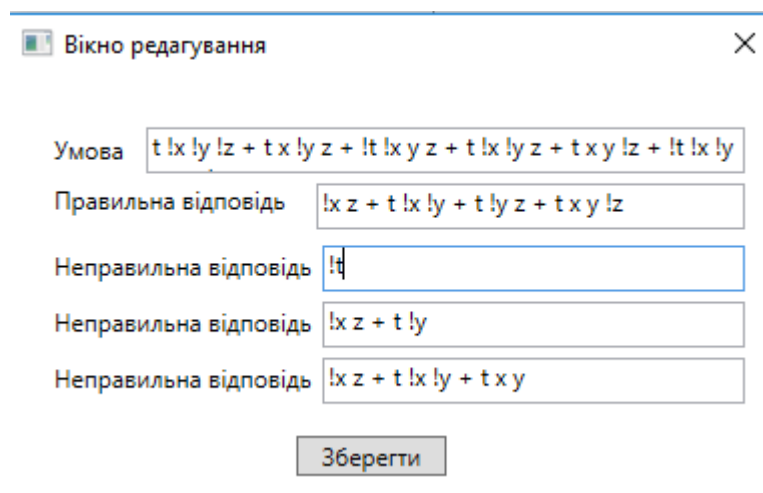


Рисунок 3.3.8 Вікно редагування

Змінимо одну з неправильних відповідей та збережемо завдання, натиснувши після цього кнопку “Зберегти”. Після цього вікно редагування закривається, а у список запитань зможе відобразити зміни, проведені під час редагування. Після цього видно, що другий варіант в однакових запитаннях відрізняється один від одного, адже для одного з завдань він був відредагований.

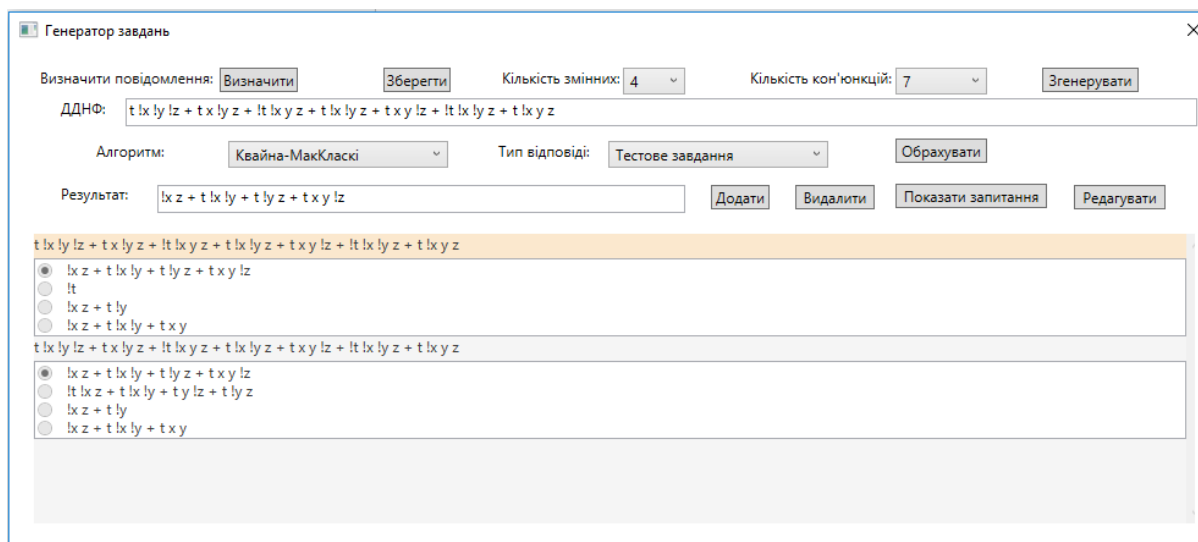


Рисунок 3.3.9 Зміни у списку запитань після редагування

Оскільки одне і те саме запитання можна випадково додати до списку запитань двічі, у користувача повинна бути можливість видалення непотрібних завдань зі списку. Для цього необхідно натиснути кнопку “Видалити”. Вибране завдання видалиться зі списку.

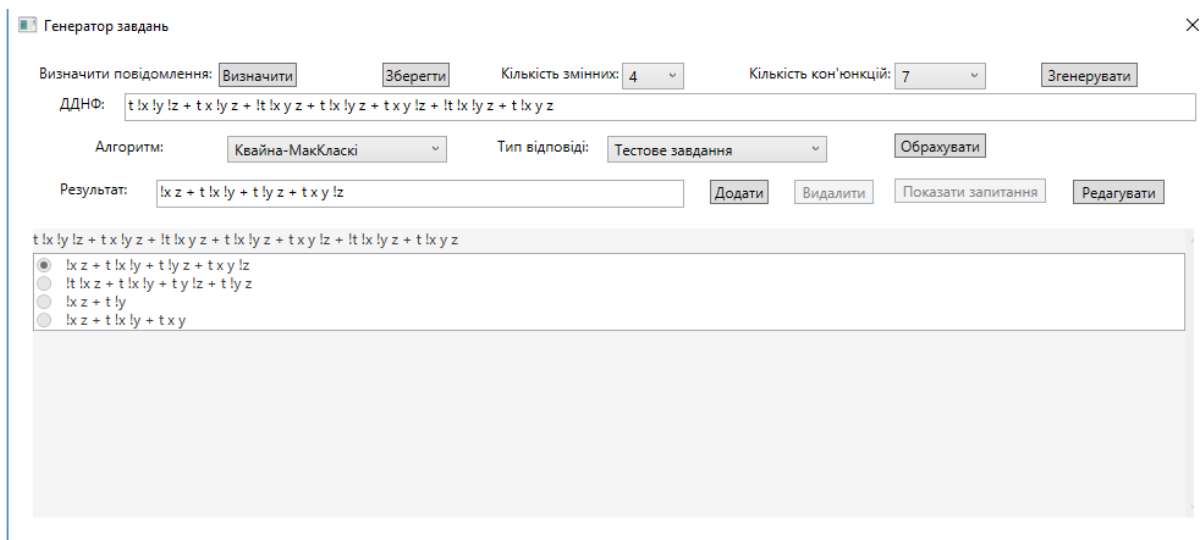


Рисунок 3.3.10 Результат видалення завдання

Додамо ще декілька завдань, проте на цей раз з різними умовами, генеруючи для них окремі ДДНФ та спрощуючи них. В результаті отримуємо декілька завдань з різними умовами та відповідями. Такий функціонал дозволить підтримувати декілька варіантів модульної роботи.

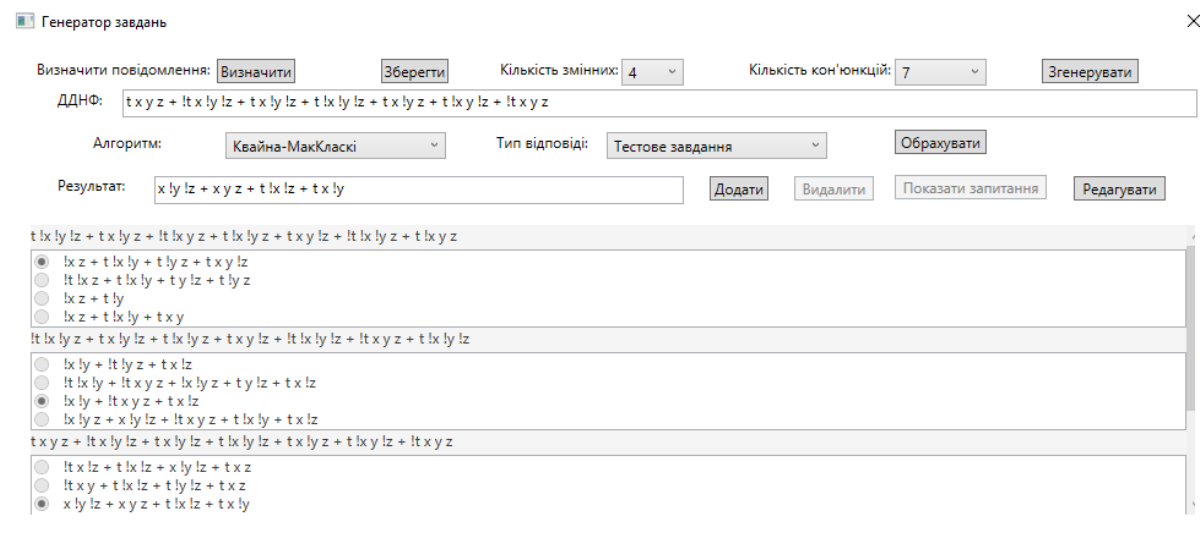


Рисунок 3.3.11 Декілька варіантів згенерованих завдань

Після того, як необхідна користувачеві кількість завдань була згенерована, необхідно зберегти завдання до файлу, який у подальшому

буде імпортований в систему Moodle. Для цього необхідно натиснути кнопку “Зберегти”. У відкритому діалоговому вікні необхідно вказати шлях та назву файлу. При цьому формат файлу вже автоматично вибраний програмою.

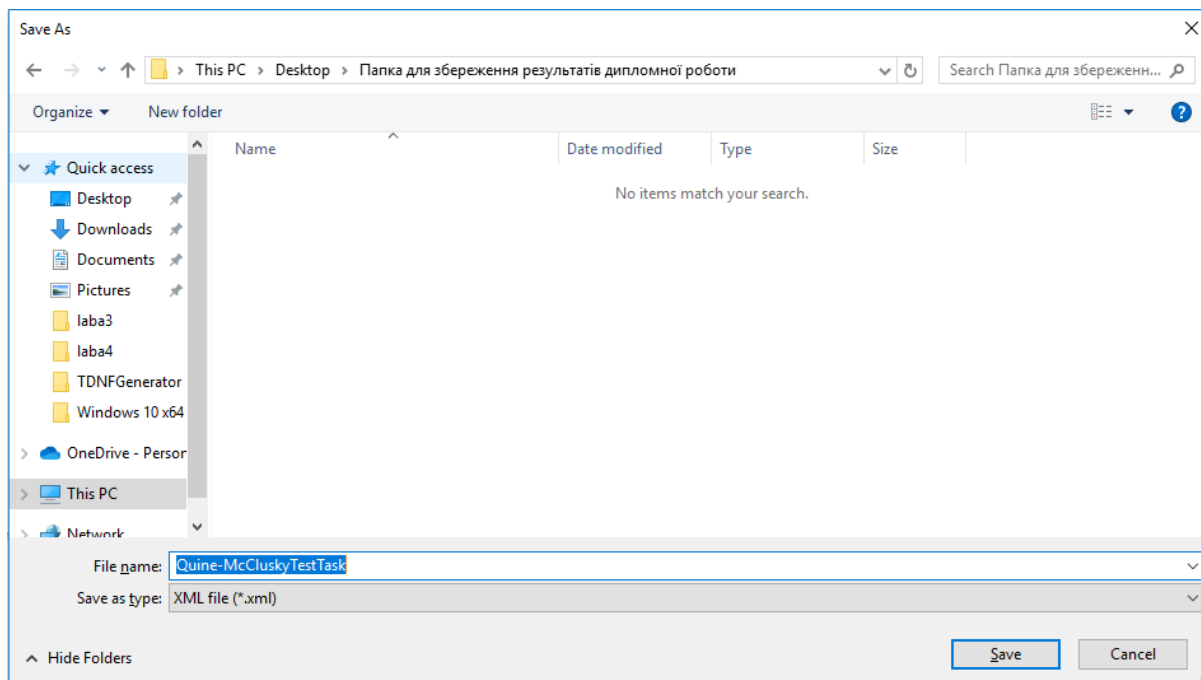
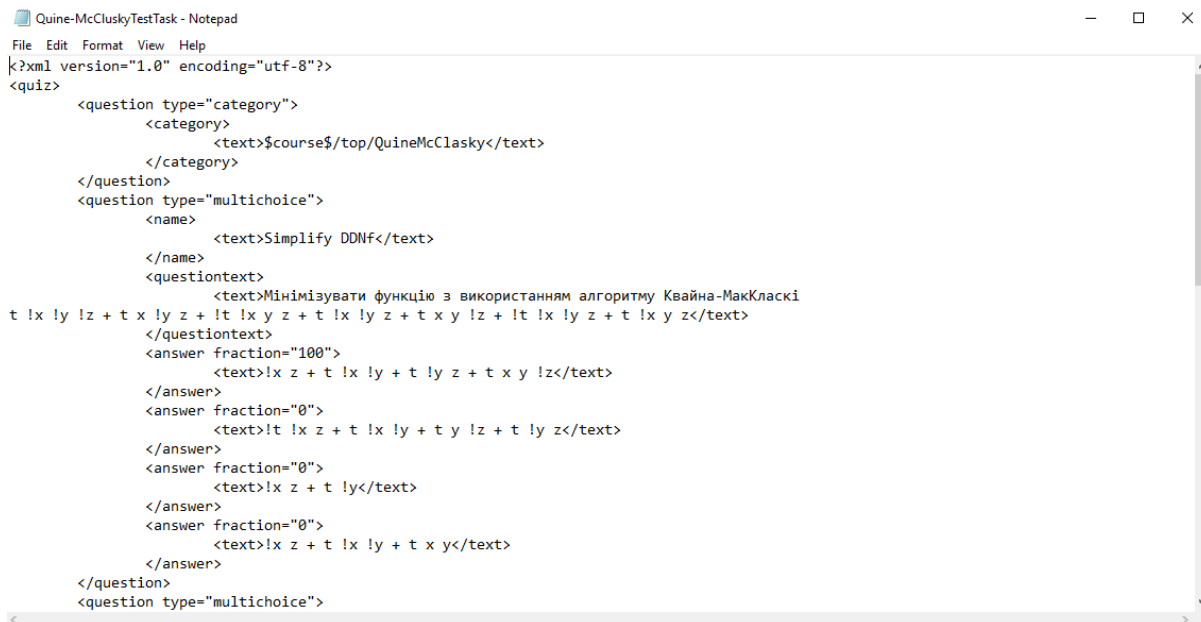


Рисунок 3.3.12 Збереження файлу

В збереженому файлі можна перевірити, як саме було збережено інформацію. Для цього необхідно відкрити файл в будь-якому текстовому редакторі. Відкривши файл, можна побачити, що у файлі присутні декілька запитань, та встановлена категорія питання.



```

Quine-McCluskyTestTask - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<quiz>
  <question type="category">
    <category>
      <text>$course$/top/QuineMcClusky</text>
    </category>
  </question>
  <question type="multichoice">
    <name>
      <text>Simplify DDNF</text>
    </name>
    <questiontext>
      <text>Мінімізувати функцію з використанням алгоритму Квайна-МакКласкі
t !x !y !z + t x !y z + !t !x y z + t !x !y z + t x y !z + !t !x !y z + t !x y z</text>
    </questiontext>
    <answer fraction="100">
      <text>!x z + t !x !y + t !y z + t x y !z</text>
    </answer>
    <answer fraction="0">
      <text>!t !x z + t !x !y + t y !z + t !y z</text>
    </answer>
    <answer fraction="0">
      <text>!x z + t !y</text>
    </answer>
    <answer fraction="0">
      <text>!x z + t !x !y + t x y</text>
    </answer>
  </question>
  <question type="multichoice">

```

Рисунок 3.3.13 Вміст збереженого файлу для імпорту

Після того, як файл файлу був згенерований, необхідно налаштувати тестову систему Moodle. Для цього необхідно авторизуватись в системі Moodle Sandbox, використовуючи обліковий запис Адміністратора ресурсу. Після авторизації необхідно створити власний курс, в якому будуть розміщені завдання.

Minimization Course

Course Settings Participants Grades Reports More ▾

General

Collapse all



Announcements

Рисунок 3.3.14 Створений новий курс в системі Moodle

Далі, необхідно знайти вкладку “Question bank” та змінити вибраний елемент комбінованого списку на “Import”. Це дозволить перейти на сторінку, яка дозволяє почати процес імпорту запитань. Перед початком процесу імпорту важливо зазначити, що формат імпортованого файлу відповідає формату “Moodle XML”.

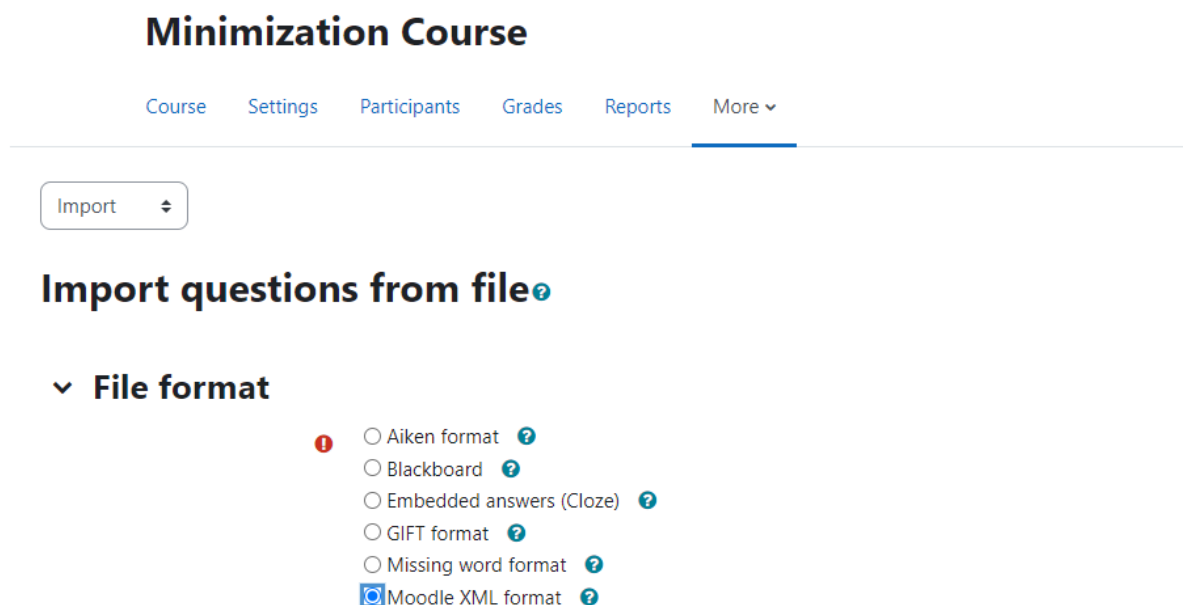


Рисунок 3.3.15 Сторінка імпорту завдань

Після того, як параметри імпорту були визначені необхідно вибрати файл з локальної системи та завантажити його до системи Moodle. Після завантаження, система Moodle автоматично парсить та перевіряє імпортований файл на відповідність формату. У випадку, якщо формат правильний, система Moodle проінформує про це, позначивши кожне правильне імпортоване завдання зеленим кольором.

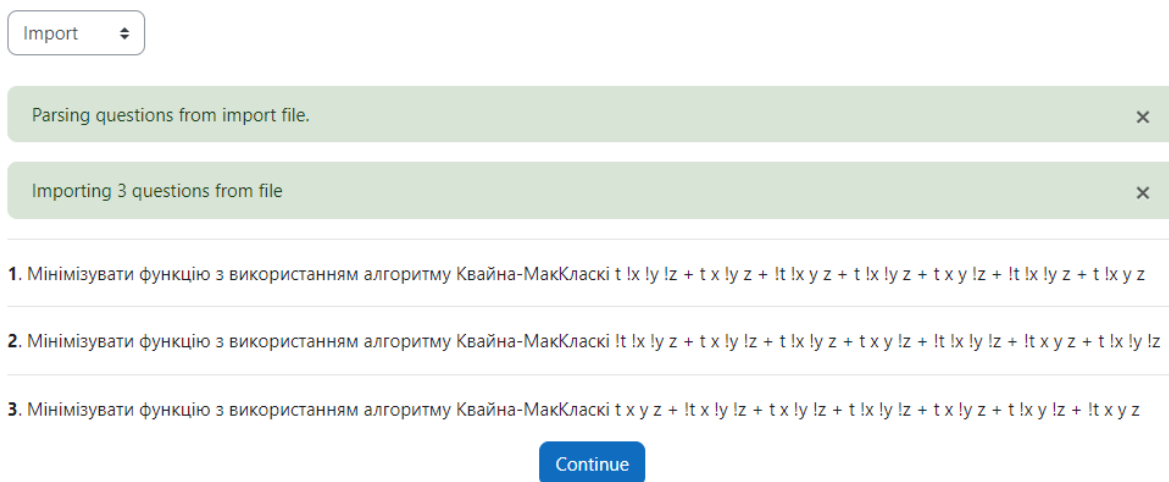


Рисунок 3.3.16 Імпортовані завдання пройшли перевірку

Після того, як кнопка “Continue” була натиснута, знову відкриється вікно банку запитань. На цей раз в банку запитань буде присутня нова категорія запитань, яка була визначена при генерації завдання. Назва категорії відповідає назві алгоритму мінімізації.

Question bank

Select a category:

QuineMcClasky (3) ▾

No tag filters applied

Рисунок 3.3.17 Нова категорія завдань після імпорту файла

Нижче на цій сторінці буде відображено список імпортованих запитань. Кожному завданню можна змінити статус, та дозволити, або заборонити його подальше використання при створенні модулів. Запитання відображаються лише для вибраної категорії, тому якщо якийсь запитання було створене для іншої категорії, воно не буде відображене

Question name / ID number	Actions	Status	Version	Created by First name / Last name / Date	Comments	Needs checking?	Facility index	Discriminative efficiency	Usage	Last used	Modified by First name / Last name / Date
Simplify DDNf	Edit	Ready	v1	Admin User 9 April 2023, 1:37 PM	0	-	N/A	N/A	0	Never	Admin User 9 April 2023, 1:37 PM
Simplify DDNf	Edit	Ready	v1	Admin User 9 April 2023, 1:37 PM	0	-	N/A	N/A	0	Never	Admin User 9 April 2023, 1:37 PM
Simplify DDNf	Edit	Ready	v1	Admin User 9 April 2023, 1:37 PM	0	-	N/A	N/A	0	Never	Admin User 9 April 2023, 1:37 PM

With selected ^

Рисунок 3.3.18 Список імпортованих завдань з мінімізації з використанням алгоритму Квайна-МакКласкі

Після того, як банк запитань було поповнено імпортованими завданнями, необхідно створити модуль в самому курсі, щоб налаштувати опитування. Для цього необхідно включити режим редагування, натиснувши на перемикач “Edit mode” у верхньому правому кутку сторінки. Після ввімкнення режиму редагування користувача відкриється можливість створювати різні активності в курсі, серед них створення модулів. Створимо активність, а в якості вмісту активності, виберемо модуль (Quiz). При створенні модулю, тестова система Moodle вимагає серед обов'язкових налаштувань лиге назву завдання.

Updating: Quiz

[Expand all](#)

General

Name



Minimization Quiz

Рисунок 3.3.19 Назва модулю

Після того, як завдання було створене, необхідно додати одне випадкове запитання зі списку імпортованих. Для цього необхідно зайти на вкладку “Questions”, вибрати елемент випадяючого списку “Add random question”, при цьому можна зазначити, що завдання буде обиратись з певної категорії завдань. Після цього, система Moodle буде обирати випадкове запитання з обраної категорії кожний раз, коли студент починає проходити модуль.

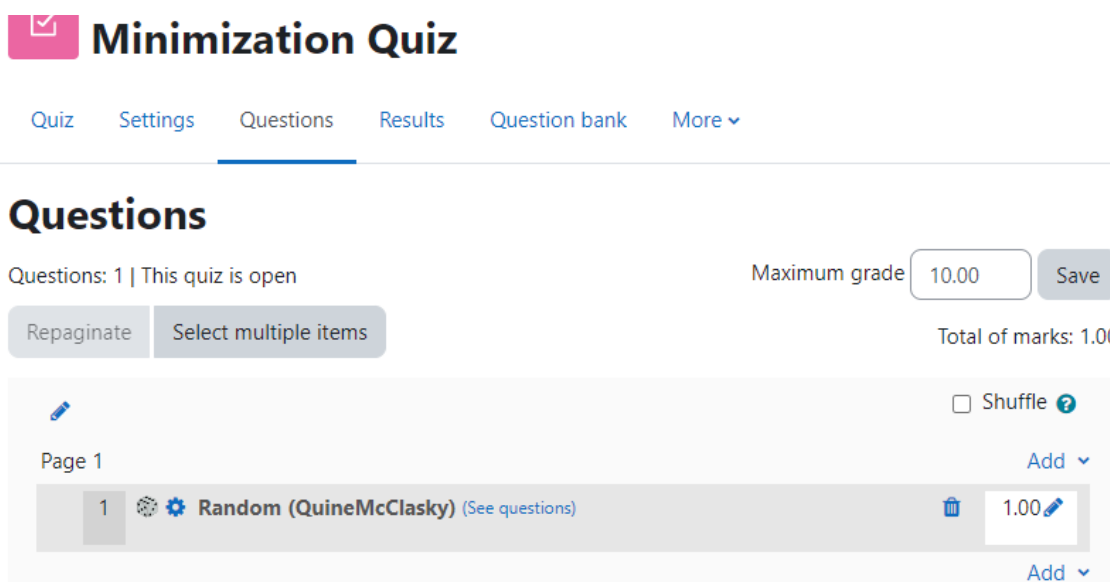


Рисунок 3.3.20 Випадково обране завдання з визначеної категорії

Модульне завдання повністю згенероване, тому тепер необхідно перевірити, чи правильно вона працює. Для цього, необхідно переміститися на вкладку “Quiz” та натиснути кнопку “Preview Quiz”. Це дозволить користувачу запусити тестове завдання. Після тестового запуску модульного завдання можна побачити, що завдання підтримує тестовий формат та містить у собі визначене раніше додаткове повідомлення.

Question 1
Not yet answered
Marked out of 1.00
[Flag question](#)
[Edit question](#)

Мінімізувати функцію з використанням алгоритму Квайна-МакКласкі $lt\ lx\ ly\ z + tx\ !y\ lz + t\ !x\ !y\ z + tx\ y\ !z + !t\ !x\ !y\ !z + !t\ x\ y\ z + t\ !x\ !y\ !z$

Select one:

a. $lx\ !y + !t\ !y\ z + tx\ !z$

b. $!t\ !x\ !y + !t\ x\ y\ z + !x\ !y\ z + t\ y\ !z + tx\ !z$

c. $lx\ !y + !t\ x\ y\ z + tx\ !z$

d. $!x\ !y\ z + x\ !y\ !z + !t\ x\ y\ z + t\ !x\ !y + tx\ !z$

[Clear my choice](#)

[Finish attempt ...](#)

Рисунок 3.3.21 Тестове завдання в системі Moodle

Для того, щоб бути впевненим в тому, що завдання правильно згенероване, необхідно перевірити правильність перевірки. Для цього виберемо правильну відповідь та сдамо завдання, натиснувши кнопку “Finish attempt”. Після цього в відкритому вікні побачимо, що вибрана правильна відповідь вважається системою як валідна, а отже тестові завдання були коректно впроваджені.

Started on	Sunday, 9 April 2023, 2:57 PM
State	Finished
Completed on	Sunday, 9 April 2023, 2:58 PM
Time taken	47 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1
Correct
Mark 1.00 out of 1.00
[Flag question](#)
[Edit question](#)

Мінімізувати функцію з використанням алгоритму Квайна-МакКласкі $lt\ lx\ ly\ z + tx\ !y\ lz + t\ !x\ !y\ z + tx\ y\ !z + !t\ !x\ !y\ !z + !t\ x\ y\ z + t\ !x\ !y\ !z$

Select one:

a. $lx\ !y + !t\ !y\ z + tx\ !z$

b. $!t\ !x\ !y + !t\ x\ y\ z + !x\ !y\ z + t\ y\ !z + tx\ !z$

c. $lx\ !y + !t\ x\ y\ z + tx\ !z$ ✓

d. $!x\ !y\ z + x\ !y\ !z + !t\ x\ y\ z + t\ !x\ !y + tx\ !z$

The correct answer is: $lx\ !y + !t\ x\ y\ z + tx\ !z$

Рисунок 3.3.22 Тестове завдання правильно вирішене

Для того, щоб тестування програми було повноцінним, необхідно ще протестувати функціонал модулів, які були розроблені до початку

модифікації генератора завдань. Щоб це перевірити, згенеруємо ще декілька завдань з мінімізації, використовуючи алгоритм лінійної з використанням операцій склейок і поглинання, де типом запитання буде коротка відповідь. При цьому в повідомленні зазначимо, що мінімізації необхідно проводити за іншим алгоритмом.

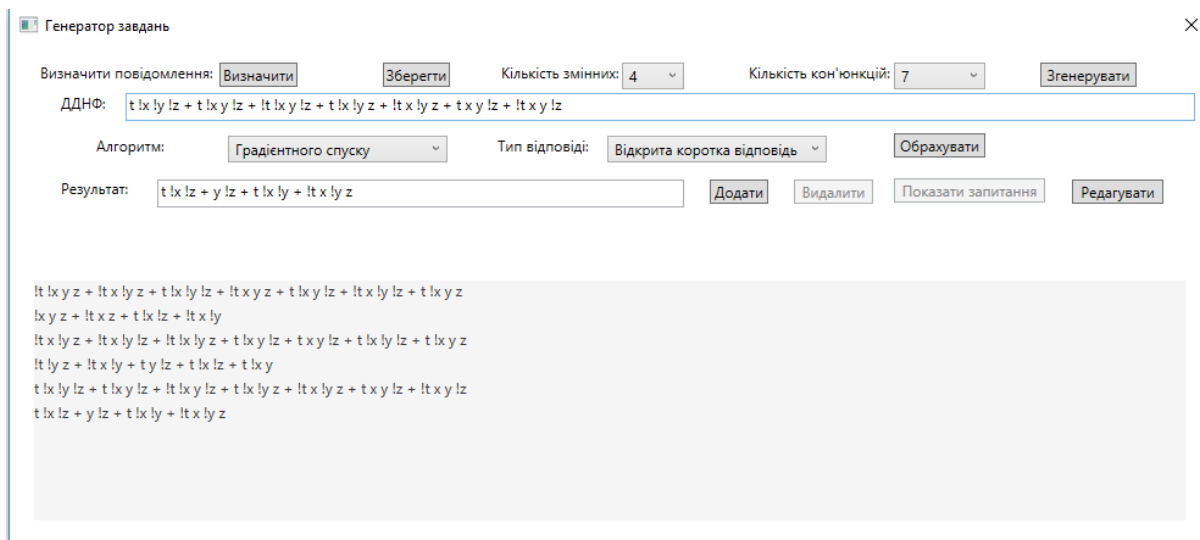


Рисунок 3.3.23 Запитання з короткою відповіддю

Збережемо ці запитання в окремий файл, та імпортуємо його до системи Moodle. Після імпорту файлу можна впевнитись в тому, що документ був правильно пропаршений системою Moodle.

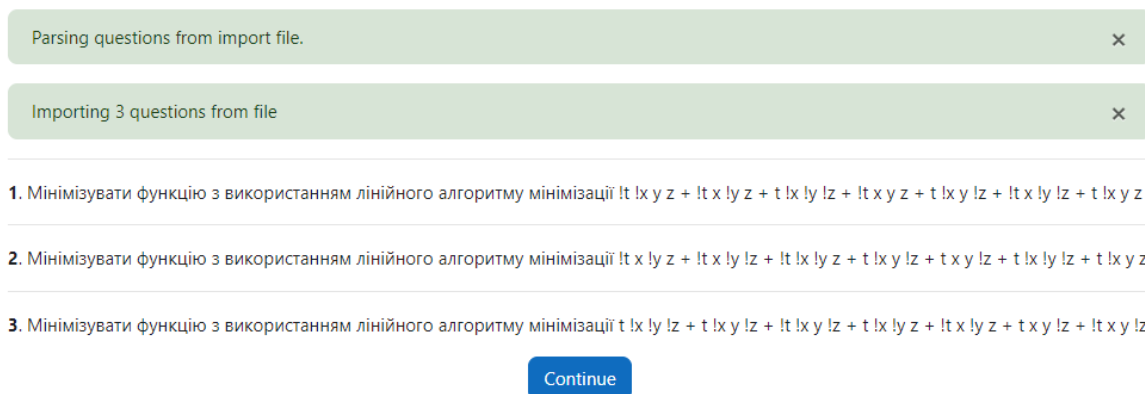


Рисунок 3.3.24 Імпорт запитань з короткою відповіддю

Після імпорту завдань важливо переконатись, що завдання були вставлені в правильну категорію, відмінну від попередньої, адже алгоритм мінімізації відрізняється.

Question bank

Select a category: LinearMinimization (3) ▾

Рисунок 3.3.25 Нова категорія завдань

Аналогічно до попереднього завдання, створюємо в системі Moodle ще один тест, в якому вибираємо завдання вже з категорії запитань на використання алгоритму лінійної мінімізації.

Questions

Questions: 1 | This quiz is open

Maximum grade 10.00 Save

Repaginate Select multiple items

Total of marks: 1.00

✎ Page 1 Shuffle ?

1	⚙️ Random (LinearMinimization) (See questions)	🗑️ 1.00 ✎
---	--	-----------

Add ▾

Рисунок 3.3.26 Завдання додані до модулю

Останнім кроком перевірки буде запуск модулю, в якому імпортовані завдання на мінімізацію булевих функцій з використанням лінійного алгоритму.

QUIZ Linear Minimization Quiz

Quiz Settings Questions Results Question bank More ▾

Back

Question 1
Not yet answered
Marked out of 1.00
Flag question
Edit question

Мінімізувати функцію з використанням лінійного алгоритму мінімізації $!t x !y z + !t x !y !z + !t !x !y z + t !x !y !z + t x y !z + t !x !y !z + t !x y z$

Answer:

Рисунок 3.3.27 Завдання типу Short Answer

Після вводу правильної відповіді переконуємось, що система Moodle правильно ідентифікувала введену ТДНФ.

Started on	Sunday, 9 April 2023, 5:28 PM
State	Finished
Completed on	Sunday, 9 April 2023, 5:31 PM
Time taken	3 mins 18 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question
Edit question

Мінімізувати функцію з використанням лінійного алгоритму мінімізації $!t x !y z + !t x !y !z + !t !x !y z + t !x !y !z + t x y !z + t !x !y !z + t !x y z$

Answer: ✓

The correct answer is: $!t !y z + !t x !y + t y !z + t !x !z + t !x y$

Рисунок 3.3.28 Завдання правильно перевірене

ВИСНОВКИ

В даній роботі було розглянуто принципи роботи та проблеми попередньої версії генератора завдань з мінімізації булевих функцій для тестової системи Moodle. Проаналізовано існуючі програмні рішення на основі створеного списку вимог. Список вимог не дозволяє використовувати існуючі рішення в генераторі завдань, тому було вирішено змінювати функціонал застосунку, доповнюючи його новими модулями та виправляючи існуючі помилки.

В ході виконання роботи було реалізовано алгоритм Квайна-МакКласкі, який містить в собі метод Петрика. Правильність роботи алгоритму Квайна-МакКласкі було перевірено. В ході перевірки жодних проблем не було знайдено. Було розглянуто проблему неоднозначності перевірки завдань при генерації завдань типу “Коротка відповідь”. Дану проблему було вирішено за допомогою підтримки генерації завдань тестового формату. В ході реалізації змін в коді використовувались такі практики написання коду як використання шаблонів проектування, модульне програмування. Створені модулі були протестовані в системі Moodle в умовах, максимально наближених до реальних. Тестування показало, що нововведення працюють згідно до поставлених задач.

ПЕРЕЛІК ПОСИЛАНЬ

1. Патерни проектування [Електронний ресурс]. URL: <https://refactoring.guru/uk/design-patterns>
2. Патерн MVVM [Електронний ресурс]. URL: <https://metanit.com/sharp/wpf/22.1.php>
3. Огляд xaml [Електронний ресурс]. URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/xaml/?view=netdesktop-7.0>
4. Нормальні та довершені нормальні форми логічних функцій [Електронний ресурс]. URL: <https://epa.kpi.ua/wp-content/Buryan/%D0%90%D0%A2%D0%9F%D0%A3-1.%20%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%202.pdf>
5. Data binding overview [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/data/?view=netdesktop-7.0>
6. Commanding overview [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/advanced/commanding-overview?view=netframeworkdesktop-4.8>
7. Short-Answer question type [Електронний ресурс]. URL: https://docs.moodle.org/401/en/Short-Answer_question_type
8. All about Quine-McClusky [Електронний ресурс]. URL: <https://web.archive.org/web/20090107120802/http://www.embedded.com/columns/programmerstoolbox/29111968>
9. Метод Блейка-Порецького [Електронний ресурс]. URL: https://web.posibnyky.vntu.edu.ua/fitki/6kondratenko_komp_praktikum_matlog/33..htm
10. Метод Петрика [Електронний ресурс]. URL:

http://ptca.narod.ru/lec/lec4_6.html

11. Multiple Choice question type [Електронний ресурс]. URL:

https://docs.moodle.org/401/en/Multiple_Choice_question_type

12. Шаблони поведінки [Електронний ресурс]. URL:

<https://metanit.com/sharp/patterns/3.1.php>

13. Course categories [Електронний ресурс]. URL:

https://docs.moodle.org/401/en/Course_categories

14. Sandbox [Електронний ресурс]. URL:

https://moodle.org/plugins/local_sandbox

ДОДАТКИ

ДОДАТОК А

```
using TDNFGenerator.Model.Interfaces;

namespace TDNFGenerator.Model.Algorithms
{
    public class AlgorithmsContext
    {
        public IMinimizationAlgorithm chosenAlgorithm { get; private set; }

        public AlgorithmsContext(IMinimizationAlgorithm minimization)
        {
            chosenAlgorithm = minimization;
        }

        public string ExecuteMinimization(string DDNF)
        {
            return chosenAlgorithm.MinimizeFunction(DDNF);
        }

        public void SetAlgorithm(IMinimizationAlgorithm
minimizationAlgorithm)
        {
            chosenAlgorithm = minimizationAlgorithm;
        }
    }
}
```

ДОДАТОК Б

```
using System.Collections.ObjectModel;
using TDNFGenerator.Model.Interfaces;

namespace TDNFGenerator.Model.XmlSavers
{
    public class XmlSaverContext
    {
        private IXmlSaveMethod xmlSaveMethod;

        public XmlSaverContext(IXmlSaveMethod saveMethod)
        {
            xmlSaveMethod = saveMethod;
        }

        public void ExecuteXmlSaveMethod(ObservableCollection<ITask> input,
Algorithms.AlgorithmsContext minimizationAlgorithm, string path)
        {
            xmlSaveMethod.SaveXml(input,
minimizationAlgorithm.chosenAlghorithm, path);
        }

        public void SetXmlSaveMethod(IXmlSaveMethod saveMethod)
        {
            xmlSaveMethod = saveMethod;
        }
    }
}
```

ДОДАТОК В

```

using System.Collections.ObjectModel;
using System.Xml;
using TDNFGenerator.Model.Interfaces;

namespace TDNFGenerator.Model.XmlSavers
{
    public class XMLTestSaver: IXmlSaveMethod
    {
        public void SaveXml(ObservableCollection<ITask> input,
IMinimizationAlgorithm minimizationAlgorithm, string path)
        {
            XmlWriterSettings xmlWriterSettings = new XmlWriterSettings()
            {
                Indent = true,
                IndentChars = "\t",
                NewLineOnAttributes = false
            };
            using (XmlWriter writer = XmlWriter.Create(path, xmlWriterSettings))
            {

                writer.WriteStartElement("quiz");
                writer.WriteStartElement("question");
                writer.WriteAttributeString("type", "category");
                writer.WriteStartElement("category");
                writer.WriteStartElement("text");
                if (minimizationAlgorithm is TDNF)
                    writer.WriteString("$course$/top/LinearMinimization");
                else

```

```
    writer.WriteString("$course$/top/QuineMcClasky");
writer.WriteEndElement();
writer.WriteEndElement();
writer.WriteEndElement();

foreach (var task in input)
{
    var testTask = task as SingleTestTask;
    writer.WriteStartElement("question");
    writer.WriteAttributeString("type", "multichoice");
    writer.WriteStartElement("name");
    writer.WriteStartElement("text");
    writer.WriteString("Simplify DDNf");
    writer.WriteEndElement();
    writer.WriteEndElement();

    writer.WriteStartElement("questiontext");
    writer.WriteStartElement("text");
    writer.WriteString(task.Text + "\n" + task.Question);
    writer.WriteEndElement();
    writer.WriteEndElement();

    foreach (var answer in testTask.AllTestAnswers)
    {
        if (answer.Validity)
        {
            writer.WriteStartElement("answer");
            writer.WriteAttributeString("fraction", "100");
            writer.WriteStartElement("text");
```

```
        writer.WriteString(answer.Content);
        writer.WriteEndElement();
        writer.WriteEndElement();
    }
    else
    {
        writer.WriteStartElement("answer");
        writer.WriteAttributeString("fraction", "0");
        writer.WriteStartElement("text");
        writer.WriteString(answer.Content);
        writer.WriteEndElement();
        writer.WriteEndElement();
    }
}
writer.WriteEndElement();
}
writer.Flush();
}
}
```