ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____Олена ОЛЬХОВСЬКА
(підпис)

«_____»___202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ СИСТЕМИ ПРАКТИЧНИХ ЗНАНЬ З ТЕМИ «ЕВКЛІДОВІ КОМБІНАТОРНІ МНОЖИНИ ПЕРЕСТАВЛЕНЬ» З ДИСЦИПЛІНИ «ЕЛЕМЕНТИ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ»»

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

Виконавець роботи Сірик Владислав Сергійович

_____«____» _____202_ р.
(підпис)

Науковий керівник к.ф.-м.н., доцент Чілікіна Тетяна Василівна

_____«____»_____202_ р.
(підпис)

Рецензент

ПОЛТАВА 2023

РЕФЕРАТ

Записка: 59 с., 7 рис., 0 таблиць, 1 додаток, 8 джерел.

ЕВКЛІДОВІ КОМБІНАТОРНІ МНОЖИНИ, ТРЕНАЖЕР, JAVA, НАВЧАЛЬНА СИСТЕМА, АЛГОРИТМІЗАЦІЯ.

Об’єкт розробки – процес освіти з комбінаторної оптимізації.

Мета роботи – створення програмного тренажера для навчання роботі з евклідовими комбінаторними множинами перестановлень.

Методи дослідження – методи програмування на Java, використання алгоритмічних підходів в комбінаторниці, середовище розробки Apache NetBeans, тестування програмних продуктів.

У роботі проведено аналіз існуючих тренажерів для навчання комбінаторної оптимізації, визначено ключові вимоги до системи, що сприяє ефективному навчанню. Виділені основні переваги та недоліки сучасних тренажерів, що були враховані при розробці нового рішення.

Розроблено структуру тренажера, що включає модулі теоретичного навчання, практичних вправ та оцінювання знань. Велика увага приділена інтерактивності навчального процесу: інтегровані інтерактивні завдання, підказки та детальні пояснення для кращого засвоєння матеріалу.

Програмна реалізація тренажера забезпечує динамічну взаємодію з користувачем, адаптивність завдань до рівня знань студента та систематичне відстеження прогресу. Використання середовища Apache NetBeans дозволило ефективно інтегрувати різноманітні модулі системи та забезпечити їх надійність та швидкість роботи.

Тестування тренажера серед студентів виявило значне підвищення рівня їх підготовки та зрозуміння матеріалу порівняно з традиційними методами навчання, що підтверджує високу ефективність розробленого продукту та актуальність подальшого його розвитку і впровадження у навчальні курси.

Зміст

ВСТУП

У сучасному світі, де технології розвиваються стрімкими темпами, з’являється все більше можливостей оптимізувати навчальні процеси та зробити їх більш доступними та ефективними. Не так давно, зіткнувшись з пандемією COVID-19, багато навчальних закладів та організацій були змушені перейти на дистанційну роботу, що спонукало до активного використання цифрових технологій. Онлайн-платформи, віртуальні класи та освітні тренажери стали ключовими інструментами в цьому процесі.

Спеціалізовані освітні тренажери відіграють особливу роль, оскільки вони не тільки допомагають у засвоєнні теоретичних знань, але й надають можливість закріпити ці знання на практиці. Вони є незамінними, зокрема, в областях, де потрібно розв’язувати складні практичні задачі, такі як комбінаторна оптимізація.

Однією з ключових тем у цій дисципліні є алгоритмізація та програмна реалізація навчальної системи практичних знань з теми «Евклідові комбінаторні множини переставлень». Ця тема важлива, адже розуміння комбінаторних множин переставлень і способів їх оптимізації може бути використано у багатьох сферах наукових досліджень та практичних застосувань.

Об’єктом дослідження є процеси комбінаторної оптимізації та їх програмна реалізація у вигляді тренажера.

Предметом дослідження є програмний продукт — тренажер для вивчення та застосування евклідових комбінаторних множин переставлень, розроблений на мові програмування Java.

Завдання цієї роботи полягає у розробці та оцінці ефективності тренажера, який дозволяє студентам не просто вивчати теоретичні аспекти комбінаторної оптимізації, але й активно застосовувати отримані знання для розв'язання практичних задач.

Методи дослідження включають розробку алгоритмів, програмування, тестування і впровадження інтерактивних елементів навчання.

Для реалізації проекту використовується середовище програмування NetBeans IDE та мова Java, що дозволяє інтегрувати необхідні обчислювальні можливості та графічний інтерфейс.

Розробка такого тренажера актуальна і важлива для підготовки кваліфікованих фахівців, здатних ефективно працювати в області комбінаторної оптимізації, і стане вагомим внеском у систему освіти математичних та інженерних дисциплін.

1. ПОСТАНОВКА ЗАДАЧІ

Для успішного виконання дипломної роботи на тему "Алгоритмізація та програмна реалізація навчальної системи практичних знань з теми «Евклідові комбінаторні множини переставлень»" необхідно здійснити наступні кроки:

- Вивчити теоретичний матеріал по темі евклідових комбінаторних множин переставлень та глибоко розібратися у специфіках цих структур.
- Ознайомитись з існуючими електронними тренажерами в області комбінаторної оптимізації, щоб зрозуміти методи і підходи до викладання та тренування.
- Вибрати приклади і сценарії використання евклідових комбінаторних множин переставлень, які будуть інтегровані в тренажер.
- Розробити детальний алгоритм роботи тренажера.

Тренажер повинен не тільки надавати користувачам різні типи завдань для практичних вправ, але й дозволяти їм ознайомитися з теоретичними аспектами до їх виконання. Крім того, інтерфейс тренажера має бути інтуїтивно зрозумілим і доступним для користувачів різного рівня підготовки, що вимагає наступних елементів:

- Стартова сторінка, яка забезпечує вступ та навігацію по програмі.
- Розділ для ознайомлення з теоретичним матеріалом, що допоможе користувачам підготуватися до вирішення практичних завдань.
- Модуль з завданнями та вправами, де користувачі можуть застосовувати свої знання.
- Варіанти відповідей або поле для введення рішень, залежно від типу завдання.
- Система зворотного зв'язку, яка повідомляє про помилки, надає підказки або показує приклади розв'язання схожих задач.
- Функція завершення роботи з тренажером у будь-який час, що дозволяє гнучко планувати навчальний процес.

Завданням цього проекту є створення такого тренажера, який зробить процес вивчення евклідових комбінаторних множин переставлень не тільки ефективним, але

й захоплюючим, забезпечуючи глибоке розуміння теми та підготовку студентів до розв'язання складних практичних задач.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Переваги використання тренажерів для вивчення комбінаторних оптимізацій

Використання навчальних тренажерів для вивчення комбінаторних оптимізацій пропонує ряд значущих переваг, які сприяють збагаченню освітнього процесу та підвищенню якості засвоєння матеріалу студентами. Такі інструменти стають незамінними в сучасних умовах навчання, оскільки вони відповідають потребам динамічної освітньої моделі і підтримують активне навчання через практичний досвід.

Доступність і гнучкість: Навчальні тренажери можна використовувати в будь-який час і в будь-якому місці, де є доступ до інтернету. Це дозволяє студентам планувати свій навчальний процес відповідно до особистих обставин та ритму життя, сприяючи більш ефективному вивченню матеріалу.

Інтерактивність: Тренажери з комбінаторної оптимізації часто включають інтерактивні завдання, які можуть бути налаштовані на індивідуальні потреби учня. Інтерактивність підвищує залученість та мотивацію студентів, допомагаючи їм краще засвоювати складні концепції через безпосередній досвід.

Миттєвий зворотний зв'язок: Системи тренажерів часто надають миттєвий зворотний зв'язок щодо виконання завдань, що дозволяє студентам швидко виявляти та виправляти помилки. Це сприяє глибшому розумінню матеріалу та допомагає уникнути затягування помилок у майбутньому.

Персоналізація навчання: Тренажери можуть бути адаптовані до рівня знань кожного студента, що дозволяє викладачам та учням зосередитися на слабких сторонах, забезпечуючи більш цілеспрямоване та ефективне навчання.

Розвиток критичного мислення: Розв'язання задач через тренажери з комбінаторної оптимізації сприяє розвитку критичного та аналітичного мислення. Студенти навчаються аналізувати задачі, розробляти стратегії рішення та оцінювати результати своїх рішень.

Покращення результатів навчання: Використання тренажерів може підвищити

результати навчання, надаючи студентам можливість практикуватися частіше і глибше розуміти матеріал. Це особливо важливо для складних дисциплін, як комбінаторна оптимізація, де практичне застосування теоретичних знань може значно покращити розуміння.

Таким чином, використання тренажерів у вивченні комбінаторних оптимізацій відіграє ключову роль у підвищенні ефективності навчання, адаптації навчального процесу до потреб сучасних студентів та підготовці їх до розв'язання реальних оптимізаційних задач.

2.2 Огляд існуючих навчальних тренажерів з комбінаторної оптимізації

Аналіз існуючих навчальних тренажерів з комбінаторної оптимізації важливий для розуміння поточного стану технологій та визначення напрямків для подальшого розвитку і поліпшення. В даний час, з розвитком цифрових технологій, освітні платформи і тренажери стають все більш популярними, особливо у сферах, де необхідно велике кількість обчислень і аналізу даних, як у комбінаторній оптимізації.

Mathway та Wolfram Alpha: Ці онлайн-інструменти включають можливості для вирішення широкого спектру математичних задач, включаючи комбінаторні проблеми. Хоча вони не є спеціалізованими тренажерами вузько для комбінаторної оптимізації, їх здатність до обчислення і надання детальних роз’яснень робить їх корисними для освітніх цілей.

Coursera і Khan Academy: Ці платформи пропонують курси з комбінаторної оптимізації, які включають відеолекції, інтерактивні вправи та тести. Вони допомагають студентам зрозуміти основні концепції та методики, що використовуються у комбінаторній оптимізації, хоча і можуть не надавати глибоку індивідуалізацію процесу навчання.

CodeSignal та LeetCode: Ці платформи зосереджені на практичному застосуванні алгоритмічних навичок та оптимізації, пропонуючи задачі, що можуть включати елементи комбінаторної оптимізації. Вони корисні для розвитку навичок вирішення проблем та підготовки до технічних співбесід.

Спеціалізовані університетські ресурси: Багато технічних університетів розробляють власні тренажери для викладання специфічних курсів з комбінаторної оптимізації. Ці ресурси часто включають спеціалізовані програмні інструменти та завдання, які максимально адаптовані до курсу та його специфічних вимог.

На основі аналізу існуючих тренажерів можна визначити ключові елементи для успішної реалізації навчальної програми з комбінаторної оптимізації:

- Інтеграція теорії та практики: Ефективний тренажер повинен забезпечувати баланс між теоретичними викладками та практичними завданнями.
- Інтерактивність та залученість: Наявність інтерактивних елементів та симуляцій підвищує мотивацію студентів та сприяє глибшому засвоєнню матеріалу.
- Миттєва зворотній зв'язок: Системи, які надають студентам негайний зворотній зв'язок на їх дії, дозволяють швидше виявляти та виправляти помилки.
- Гнучкість та доступність: Важливо, щоб тренажер був доступний студентам з різним рівнем підготовки та мав гнучкі налаштування в залежності від індивідуальних потреб.

Розуміння цих аспектів існуючих тренажерів з комбінаторної оптимізації дозволяє оптимізувати та вдосконалити підходи до створення нових освітніх ресурсів, спрямованих на підготовку студентів до вирішення складних і практичних завдань у цій галузі.

2.3 Значення евклідових комбінаторних множин переставлень у комбінаторній оптимізації

Евклідові комбінаторні множини переставлень є важливим інструментом у комбінаторній оптимізації, зокрема, вони дають змогу ефективно моделювати та розв'язувати задачі, які мають велику кількість можливих рішень. Ці множини переставлень дозволяють виконувати глибокий аналіз структурних властивостей оптимізаційних задач, а також забезпечують механізми для пошуку оптимальних або наближених рішень за допомогою ефективних алгоритмів.

Математичне та алгоритмічне значення: Евклідові множини переставлень представляють усі можливі способи організації даних, що мають ключове значення в теорії графів, теорії ігор, а також в задачах маршрутизації та розкладу. Комбінаторні переставлення є основою для алгоритмів сортування, вибору, шифрування та багатьох інших операцій, які можуть бути оптимізовані для підвищення продуктивності в комп'ютерних системах та мережах.

Застосування в оптимізаційних алгоритмах: Евклідові комбінаторні множини переставлень використовуються для формулювання та розв'язання NP-повних задач, таких як задача комівояжера, задачі розкладу робіт, логістичні оптимізації тощо. Через їх здатність представляти різні конфігурації та комбінації, вони забезпечують потужний інструментарій для розробки алгоритмів, які шукають оптимальні шляхи або розподіли ресурсів.

Навчальні аспекти: Використання евклідових комбінаторних множин переставлень в освітніх тренажерах і симуляціях дозволяє студентам з комбінаторики, алгоритміки та оптимізації візуалізувати складні концепції та покращувати розуміння математичних та алгоритмічних структур. Такі тренажери стимулюють критичне мислення та аналітичні здібності, надаючи студентам інструменти для розв'язання реальних задач.

Інноваційний потенціал: Вивчення евклідових комбінаторних множин переставлень відкриває нові можливості для інновацій у різних галузях, включаючи розробку програмного забезпечення, системи штучного інтелекту та машинне навчання. Вони допомагають удосконалювати алгоритми, забезпечуючи більш ефективне використання даних і ресурсів у складних обчислювальних системах.

Таким чином, евклідові комбінаторні множини переставлень мають значний вплив на розвиток сучасної комбінаторної оптимізації, вносячи вагомий вклад у теоретичні та практичні аспекти розв'язання складних задач.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Теоретичні основи евклідових комбінаторних множин переставлень

Евклідові комбінаторні множини переставлень формують теоретичну основу для вивчення комбінаторної оптимізації, надаючи механізми для аналізу та розв'язання задач, де кількість можливих рішень є значною. Ці множини мають глибокий математичний зміст та застосування в різних аспектах наукових досліджень та інженерії.

Визначення: Евклідова комбінаторна множина переставлень — це структурована сукупність всіх можливих унікальних перестановок заданого набору елементів. У контексті математики і комбінаторики, перестановка визначається як впорядкований розподіл елементів, де порядок розташування має значення. Для множини з n унікальних елементів існує $n!$ (факторіал від n) можливих перестановок, кожна з яких є унікальною конфігурацією цих елементів.

Математичне моделювання: Евклідові комбінаторні множини можуть бути представлені і аналізовані через використання геометричних і алгебраїчних понять. Одним з підходів є представлення кожної перестановки як точки у високовимірному просторі, де кожна вісь відповідає позиції одного з елементів у перестановці.

- Простір перестановок: Якщо ми розглядаємо множину з nn елементів, то кожна перестановка може бути представлена як точка у nn-вимірному просторі. Кожна вісь у цьому просторі представляє один з елементів, а її значення — позицію цього елемента у перестановці.
- Гіперплощини і перетини: Математичні властивості перестановок можна досліджувати через аналіз гіперплощин, які поділяють простір на різні регіони, відповідно до властивостей перестановок, таких як інверсії (коли елементи розташовані в порядку, відмінному від їх природного порядку).
- Оптимізаційні шляхи: Використання математичних методів, таких як теорія графів, дозволяє моделювати шляхи через цей простір для вирішення оптимізаційних задач, таких як знаходження найкоротшого шляху, який відвідує всі можливі перестановки за певними критеріями.

Операції з евклідовими комбінаторними множинами переставлень включають ряд дій, які можуть бути виконані для аналізу або оптимізації цих множин:

- Генерація перестановок: Ця операція полягає у створенні всіх можливих унікальних перестановок визначеної множини елементів. Це може бути зроблено через рекурсивні алгоритми або за допомогою ітеративних методів, які систематично генерують перестановки без повторень.
- Пошук перестановок: Це включає вибір специфічних перестановок, які задовольняють певні умови або критерії. Методи пошуку можуть базуватися на властивостях перестановок, таких як кількість інверсій або відповідність патерну.
- Оптимізація: Застосування математичних алгоритмів для визначення "найкращої" перестановки відповідно до заданої цільової функції, такої як мінімізація вартості, часу або дистанції у задачах логістики або виробництва.

Прикладні задачі

Евклідові комбінаторні множини переставлень застосовуються в широкому спектрі оптимізаційних задач:

- Задача комівояжера (Traveling Salesman Problem, TSP): Визначення найкоротшого можливого маршруту, який включає відвідування серії міст і повернення до початкового пункту. Розв'язання цієї задачі часто включає вивчення всіх можливих перестановок маршрутів.
- Планування і розклад: Оптимізація розподілу завдань або ресурсів з мінімальними витратами або часом очікування. Перестановки використовуються для оцінки різних можливих розкладів.
- Логістика та розподіл ресурсів: Ефективне розподілення ресурсів або планування маршрутів доставки залежить від здатності моделювати та оптимізувати перестановки відповідно до даних критеріїв.

Науковий та педагогічний потенціал

Евклідові комбінаторні множини переставлень мають значний науковий та педагогічний потенціал:

- Наукові дослідження: Вони служать основою для розвитку нових теоретичних

підходів в математиці та інформатиці, включаючи алгоритмічні стратегії для вирішення складних комбінаторних задач.

- Освітній інструмент: Використання цих множин в освітніх тренажерах і курсах дозволяє студентам краще зрозуміти принципи комбінаторики, алгоритмів і оптимізації, підвищуючи їхню аналітичну компетенцію та практичні навички розв'язання задач.

Таким чином, евклідові комбінаторні множини переставлень є незамінними в академічному дослідженні та освіті, що стимулює розвиток та застосування математичних знань в реальних умовах.

3.2 Блок-схема алгоритму тренажера

У цьому розділі буде продемонтровано блок-схеми роботи тренажеру(див рис. 2.1, 2.2, 2.3).

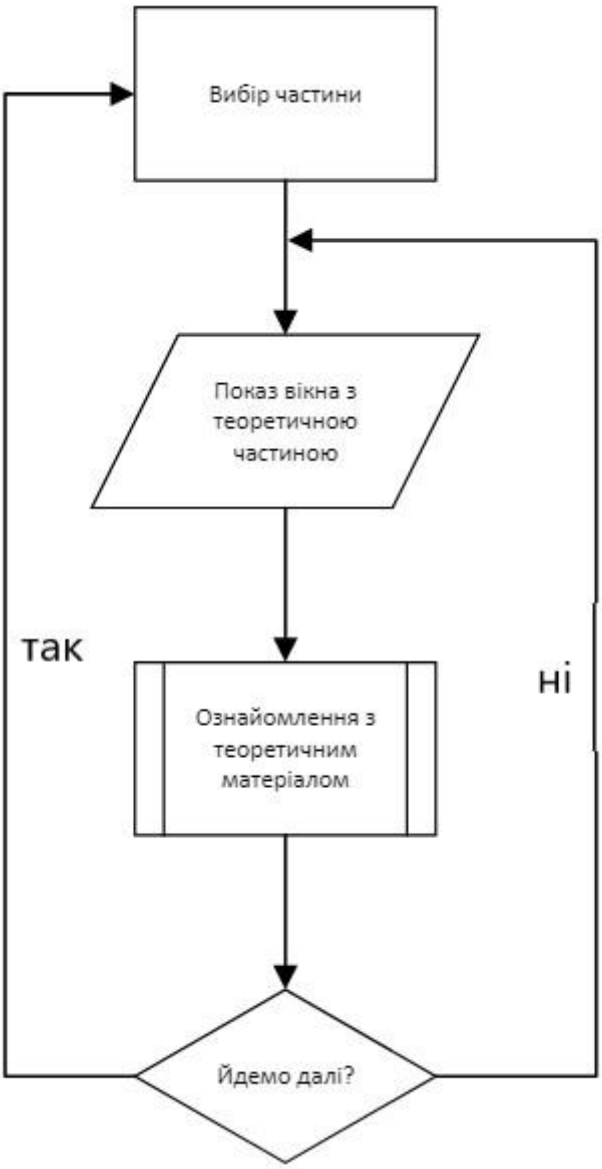

Рисунок 2.1 – Блок-схема теоретичної частини

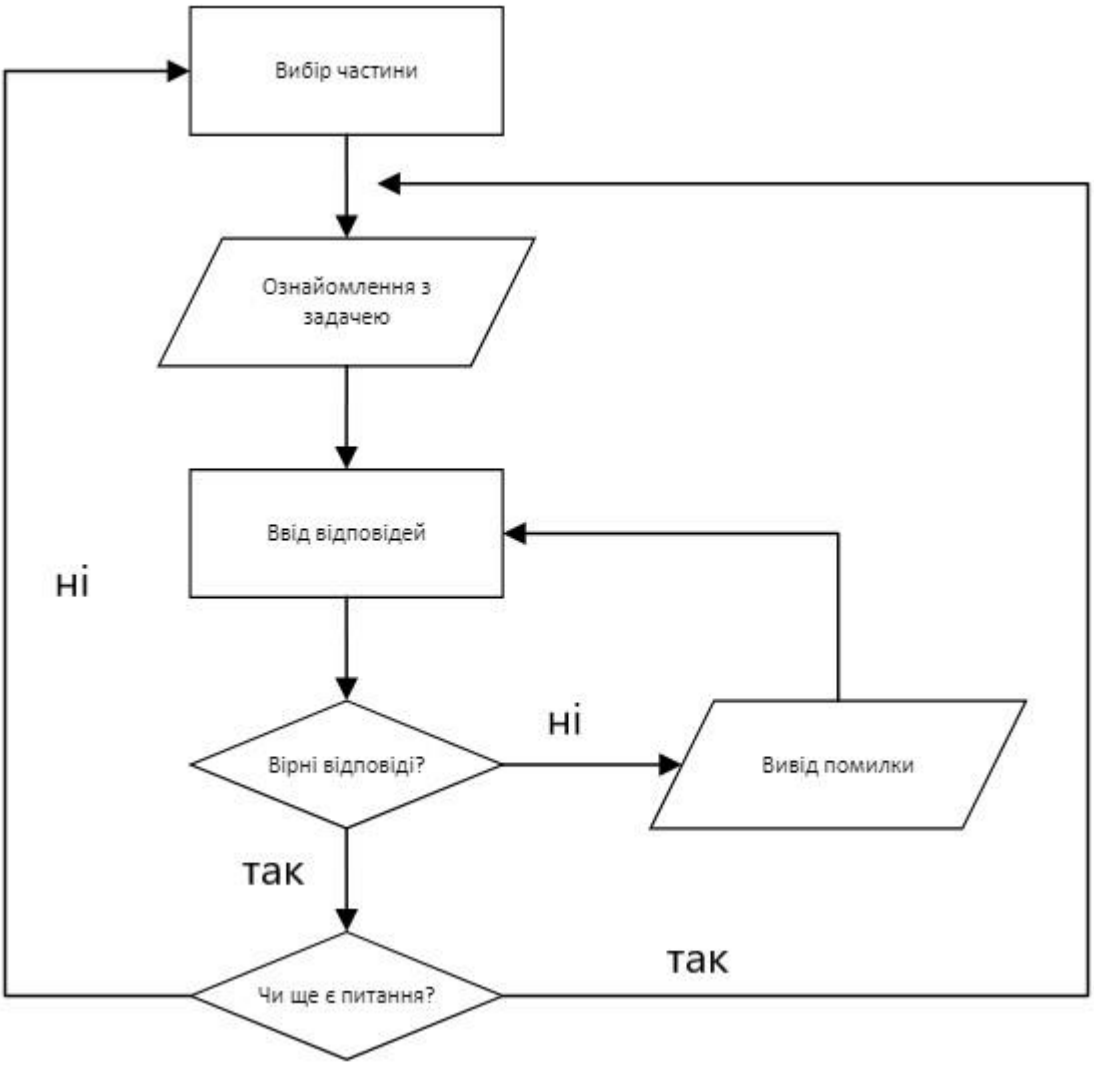

Рисунок 2.2 – Блок-схема практичної частини

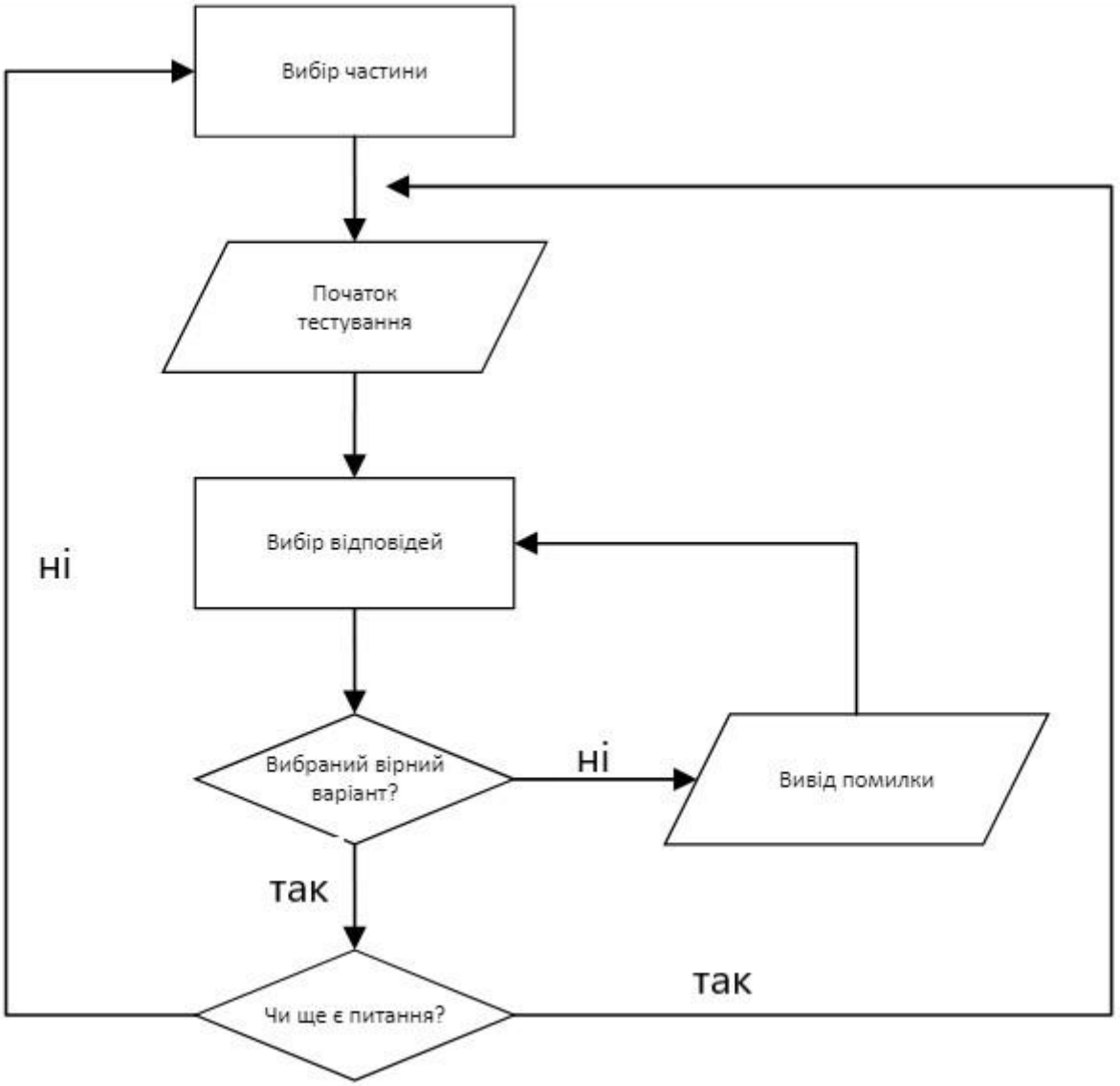

Рисунок 2.3 – Блок-схема теоретичної частини

3.3 Алгоритм роботи

Екран вітання: "Ласкаво просимо до тренажера з евклідових комбінаторних множин переставлень!

Вибір розділу:

1. Теоретична частина
2. Практична частина
3. Тестова частина"

Теоретична частина:

Вступ в комбінаторну оптимізацію

Нехай X - скінчена комбінаторна множина, x - її довільний елемент, x ∈ X, а |X| - кількість елементів в X, та нехай є заданий функціонал F: X → R'.

Основною задачею комбінаторної оптимізації називають задачу знаходження екстремуму

F* = F(x*) = extr F(x) по x ∈ X

і екстремалі

x* = arg extr F(x) по x ∈ X.

Розв'язком задачі (1), (2) називають пару <F*, x*>. Обидві задачі можуть ставитися незалежно одна від одної,

тоді під розв'язком задачі (1) розуміють F*, а задачі (2) - x*.

Основною задачею мінімізації на комбінаторній множині X називають задачу знаходження

F* = min F(x) по x ∈ X, x* = arg min F(x) по x ∈ X,

основною задачею максимізації — відповідно задачу визначення

F* = max F(x) по x ∈ X, x* = arg max F(x) по x ∈ X.

Позначимо Jn - множину n перших натуральних чисел, Jn = {1,...,n}, а також позначимо J0 = Jn ∪ {0}. Для спрощення формул та єдності форм

їх представлення важливим є, що в них J0 = ∅.

Мультимножиною G = {g1, g2, ..., gn} називають сукупність n елементів g1,...,gn, серед яких є, зазвичай кажучи, однакові (нерозрізнимі).

Мультимножина, всі елементи якої різні, є множиною. Будь-яку мультимножину G можна представити її основою S(G), тобто упорядкованою

множиною (кортежем, вектором) S(G) = (e1^j1, e2^j2, ..., em^jm) - всіх різних елементів із G, n ≤ m, (якщо і тільки тоді j1 + j2 + ... + jm = n),

то G-множина, і перевіряюча специфікація [G] - кортеж кратностей елементів основи мультимножини: k(a) = j, якщо k(a) = j; k(a) = 0,

якщо елемент a не є елементом G, тоді [G] = (η1, η2,..., ηm,..., ηn) - перша специфікація. Те, що елемент a мультимножини A має кратність k(a)

записують у вигляді a^k(a).

Інше представлення мультимножини: G = {e1^η1, e2^η2,...,em^ηm,...,en^ηn}; η1 +...+ ηn = n, де ηi ≥ 0.

Приклад 1. Нехай мультимножина така: G={1,1,3,3,3,5,6}. Знайдіть основу S(G) та перевіряючу специфікацію [G].

Маємо: S(G)={1,3,5,6}, [G]={2,3,1}.

Приклад 2. Нехай G={a,a,b,b,b,c,c,c}. Тоді S(G)=(a,b,c); [G]=(2;3;3), G={a^2,b^3,c^3}.

Дії (операції) над мультимножинами детально розглянуті в [1]. Далі буде підбірка сума мультимножин. Дамо її означення.

Нехай A - мультимножина з основою S(A)=(x,y,z, ...) і кратностями (k(a),k(y),k(a),k(z),...), а B - мультимножина з основою S(B)=(x,y,z, ...)

і кратностями (k(x),k(y),k(z),k(y),...), де k(x)≥0, k(y)≥0, k(z)≥0, k(y)≥0, k(g)≥0, k(b)≥0, k(z)≥0 тощо.

Тоді під сумою A+B мультимножин A та B розуміють мультимножину з основою

S(A+B)=S(A)∪S(B) і крайностями {k(a)+k(x),k(a)+k(y),k(a)+k(z),...}={k(a)+k(x)+k(g)+k(y),k(a)+k(z)+k(g),...}.

Суму мультимножин Ai будемо позначати ∑_i=1^N A_i .

Мультимножина B з основою S(B) називається підмультимножиною мультимножини A з основою S(A), якщо S(B)⊆S(A), і для кожного елементу

a∈S(B) виконується нерівність k(b)≤k(a), де k(a) - кратність елемента a в мультимножині. Позначається це тим же знаком ⊆, що і для множин,

тобто B⊆A.

Приклад 3. Для G з прикладу 1 G={1,1,3,3,3,5,6} - мультимножина G; G2={1,3,7,7} - не є підмультимножиною G;

G3={1,3,6,6} - не є підмультимножиною G.

Приклад 4. Нехай A={a,b,b,c,c}, B={a,a,a,b,b}, C={a,b}, G - з прикладу 2. Розглянемо чи є серед цих мультимножин підмультимножини?

A⊆G: S(A)=(a,b,c), S(A)⊆S(G), [A]=(1,2,2), [G]=(2,3,3), отже A⊆G. Розглянемо B та G: S(B)=(a,b) S(B)⊆S(G).

Позначимо для зручності порівняння [B] та [G] кратність c в B нулем: [B]=(3,2,0), де k(a)=3; [G]=(2,1,3), де k(c)=3; отже B⊈G.

Порівняємо C та G.

S(C)=(a,b)⊆S(G); k(c)≤k(z)=k(a); k(c)≤k(b)=k(b); отже C⊆G. Наведемо, як в [1], k-елементну мультимножину B у мультимножині A k-вибіркою,

тобто B - k-вибірка, якщо B⊆A, |B|=k.

Нехай k та n - натуральні константи, gj - дійсні числа j в Jn, i в In, a G = {g1,...,gn} - мультимножина з основою S(G) = (e1,...,en), первинною

специфікацією [G] = (n1,...,nn), n1 >= 1 для всіх i в Jn, n1 + ... + nm = n, n2 <= k.

Не порушуючи загальності, можна вважати, що ni <= k для всіх i в Jn.

За означенням основи мультимножини ei не дорівнює ej для всіх i, j в In, j не дорівнює i. Якщо G - мультимножина, то існує принаймні одне число

i в Jn, що ni > 1, а з цього випливає, що n > n. Розглянемо упорядкування k-вибірки у мультимножині G:

(e[g1],...,e[gi],...,e[gi],...,e[gk]),

де gi, e[gi] в G, i не дорівнює j якщо e[gi] не дорівнює e[gj] для всіх i, j в Ik.

Означення. Множину E, елементами якої є k-вибірки e(bar) = (e1,...,ek), e в E = (e1,...,ek) вважають (3) з мультимножини G наявним

евклідовим комбінаторним множиною, якщо 3 умови: gj не дорівнює k там, де ej не дорівнює ei, виливаючись e не дорівнює e'

(коротка назва: e-множина).

При цьому умові e, e' в E може виконуватись умова e' = em = ei = e'i, j не дорівнює m, j,m в Ik. Евклідова комбінаторна множина E має таку

властивість: два елементи e в E, що належать множині E, відмінні один від одного, якщо вони виризняються порядком слідування символів,

що їх утворюють, або самих символів. Тобто евклідові комбінаторні множини можна розглядати як множини точок евклідового простору R^k.

Практична частина:

Приклад 1. Нехай G={a,a,b}. Утворити множину переставлень з елементів G .

Відповідь: P3,2(G)={(a,a,b),(a,b,a),(b,a,a)}

Приклад 2. Нехай G={a,b,b,b,c,c} . Знайти S(G), [G].

Відповідь: S(G)=(a,b,c) [G]=(1,2,3)

Приклад 3. Нехай мультимножина така: G={1,3,3,4,4,4,7} Знайти основу S(G) та первинну специфікацію [G]

Відповідь S(G)=(1,3,4,7) [G]=(1,2,3,1)

Приклад 4. Нехай G={a,b,c}. Утворити множину переставлень без повторень

Відповідь: РЗ(G)={(a,b,c),(a,c,b),(b,a,c),(b,c,a),(c,a,b),(c,b,a)}

Тестова частина: Перевірка знань з евклідових комбінаторних множин переставлень

Інструкція: Оберіть найбільш відповідний варіант відповіді для кожного з наведених нижче питань.

1.1. Переставлення з повтореннями - це:

A. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів;

B. Упорядкована вибірка всіх елементів з заданої мультимножини;

C. Неупорядкована вибірка всіх елементів з заданої множини;

D. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;

Правильна відповідь: B

1.2. Мультимножиною називається

A. Набір елементів, взятих по одному разу;

B. Набір елементів, кожен з яких обов'язково має однакову кількість повторень;

C. Набір елементів, кожен з яких має свою кількість повторень;

D. Набір елементів, в якому певна кількість елементів повторюється по разу, а певна - по кілька раз.

Правильна відповідь: C

1.3. Переставлення без повторень - це:

A. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;

B. Упорядкована вибірка всіх елементів з заданої множини.

C. Неупорядкована вибірка всіх елементів з заданої множини;

D. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів;

Правильна відповідь: B

1.4. Розміщення без повторень - це:

A. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів;

B. Неупорядкована вибірка всіх елементів з заданої множини;

C. Упорядкована вибірка всіх елементів з заданої мультимножини;

D. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;

Правильна відповідь: A

4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис процесу розробки симулятора

Розробка симулятора для тренажера з евклідових комбінаторних множин переставлень включала кілька ключових етапів, кожен з яких вимагав зосередження на технічних деталях та користувацькому досвіді. Цей процес був розділений на наступні основні кроки:

1. Визначення вимог

На цьому етапі було зібрано вхідні дані від потенційних користувачів та викладачів, які могли б використовувати симулятор у навчальному процесі. Основні вимоги включали забезпечення інтерактивності, наочності матеріалу, легкість використання інтерфейсу, а також можливість демонстрації та аналізу різних комбінаторних перестановок.

2. Проектування архітектури

Була розроблена архітектура програми, що включала модуль для генерації перестановок, інтерфейс користувача (UI), модуль для візуалізації даних, а також систему оцінювання та зворотного зв'язку. Вся система була спроектована так, щоб бути масштабованою та легко налаштовуваною.

3. Вибір технологій

Вибір програмних засобів був здійснений на основі їх можливостей, гнучкості та відповідності вимогам проекту:

- **Мова програмування Java:** Обрана через її широкі можливості у створенні крос-платформних застосунків. Java має потужну стандартну бібліотеку, яка включає засоби для роботи з графікою, мережами та іншими важливими компонентами. Крім того, Java є однією з найбільш використовуваних мов програмування в навчальних закладах, що полегшує інтеграцію тренажера у навчальний процес.
- **JavaFX:** Використання JavaFX для створення графічного інтерфейсу користувача (UI) дозволило забезпечити високу інтерактивність та наочність. JavaFX підтримує сучасні технології візуалізації, що робить його ідеальним

вибором для розробки навчальних додатків з багатим користувацьким інтерфейсом.

- **Git:** Система контролю версій Git використовувалась для управління вихідним кодом проекту. Це дозволило ефективно співпрацювати в команді, відслідковувати зміни та швидко вирішувати конфлікти в коді.

4. Розробка модулів

Кожен модуль симулятора був розроблений окремо:

- Модуль генерації перестановок використовував алгоритмічні методи для створення всіх можливих комбінацій елементів.
- Інтерфейс користувача (UI) був зроблений інтуїтивно зрозумілим, з меню вибору різних опцій та візуалізацій.
- Модуль візуалізації представляв результати в графічному вигляді, підвищуючи наочність матеріалу.
- Система зворотного зв'язку дозволяла користувачам отримувати відгуки про свої дії в режимі реального часу.

5. Інтеграція та тестування

Після розробки окремих модулів, вони були інтегровані в єдину систему. Проводилось грунтовне тестування з метою виявлення та виправлення помилок, забезпечення стабільності та відповідності вимогам.

6. Розгортання та оцінка користувацького досвіду

Фінальний продукт був розгорнутий для використання в навчальних закладах. Проводилося оцінювання користувацького досвіду для подальшого вдосконалення програми.

Цей підхід до розробки симулятора дозволив створити ефективний і корисний інструмент для вивчення евклідових комбінаторних множин переставлень, що сприяє глибшому розумінню предмету і покращенню аналітичних навичок студентів.

4.2 Опис програми

Програма-симулятор, розроблена для навчання та дослідження евклідових

комбінаторних множин переставлень, є інтерактивним застосунком, який включає кілька ключових модулів для оптимізації навчального процесу. Нижче описано основні аспекти програми, її функціональні можливості, архітектуру та інтерфейс.

Архітектура програми

Програма має модульну архітектуру, що дозволяє легко розширювати та модифікувати її функціонал:

1. Модуль генерації перестановок: Відповідає за створення всіх можливих перестановок заданої множини чисел або символів.
2. Інтерфейс користувача (UI): Забезпечує візуальний інтерфейс для взаємодії користувача з програмою. Включає панелі навігації, відображення теоретичних матеріалів та виконання завдань.
3. Модуль візуалізації: Використовує графічні елементи для наочного представлення перестановок та їх властивостей, сприяючи кращому засвоєнню матеріалу.
4. Модуль зворотного зв'язку: Оцінює відповіді користувачів та надає конструктивний зворотний зв'язок, включаючи підказки та рекомендації для покращення.

Функціональні можливості

Програма забезпечує наступні основні функції:

- Вивчення теорії: Користувачі можуть переглядати теоретичні матеріали, які пояснюють основи комбінаторних множин та їх застосування.
- Виконання практичних завдань: Задачі на перестановки для тренування вирішення практичних проблем, що дозволяє закріпити теоретичні знання.
- Інтерактивні візуалізації: Динамічні графіки та схеми, що візуалізують процес генерації та аналізу перестановок.
- Тестування знань: Модуль для оцінювання рівня засвоєння матеріалу з можливістю отримання зворотного зв'язку.

Інтерфейс користувача

Інтерфейс програми спроектований таким чином, щоб забезпечити інтуїтивне та зручне використання:

- Головне меню: Забезпечує доступ до різних розділів програми, таких як теорія, практика, тестування.
- Вікно перегляду теорії: Текстовий та мультимедійний контент для вивчення основних понять.
- Вікно практичних завдань: Інтерактивні завдання з можливостями введення відповідей та отримання миттєвого зворотного зв'язку.
- Тестовий модуль: Задачі на самоперевірку з варіантами відповідей або відкритими питаннями.

Цей опис дає змогу оцінити основні аспекти програми-симулятора для навчання роботі з евклідовими комбінаторними множинами перестановлень, підкреслюючи її універсальність та ефективність у навчальному процесі.

4.3 Інструкція користувача

Вступ до тренажера

Після запуску програми тренажера користувача зустрічає стартовий екран, де відображається привітальне повідомлення та кнопки для навігації між розділами: "Теоретична частина", "Практична частина", і "Тестова частина". Виберіть потрібний розділ для подальшого використання тренажера (див рис. 4.1).

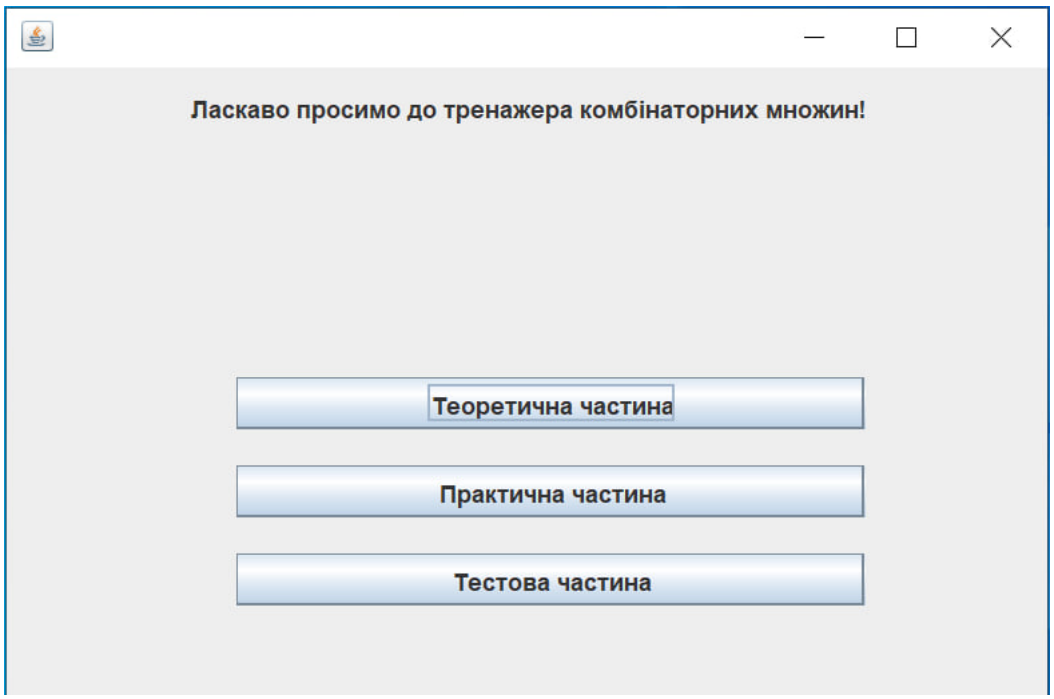

Рисунок 4.1 – Головна сторінка програми

Теоретична частина

1. Вибір "Теоретична частина": В цьому розділі представлені основні теоретичні матеріали, необхідні для вивчення комбінаторних оптимізацій та евклідових комбінаторних множин переставлень. Текстовий контент відображається в окремому вікні, де користувач може ознайомитись з теорією (див рис. 4.2).
2. Повернення до головного меню: Щоб повернутись до стартового екрану з теоретичної частини, скористайтесь кнопкою "Йдемо далі?".

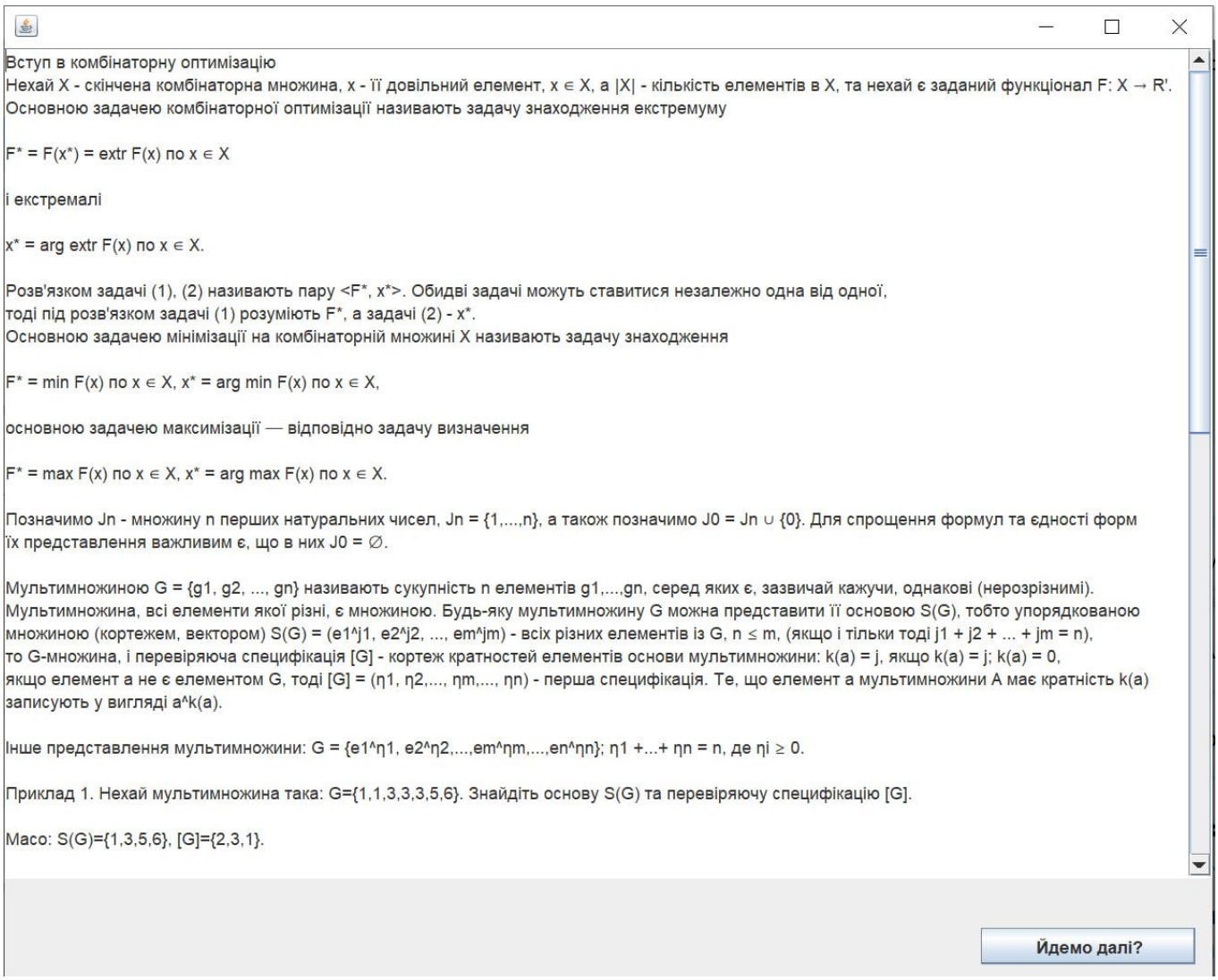

Рисунок 4.2 – Теоретична сторінка

Практична частина

1. Вибір "Практична частина": У цьому розділі користувач має можливість застосувати отримані знання на практиці. Представлено декілька завдань для введення власних рішень, які можна перевірити на правильність натисканням кнопки "Перевірити" (див рис 4.3).
2. Виконання завдань: Для кожного завдання введіть відповідь у відповідне поле і натисніть кнопку "Перевірити" для перевірки результату. У разі правильної відповіді з’явиться позитивний відгук, у разі помилки - порада або правильна відповідь.
3. Повернення до головного меню: Для переходу назад до стартового меню скористайтесь кнопкою "Перейти до вибору частини".

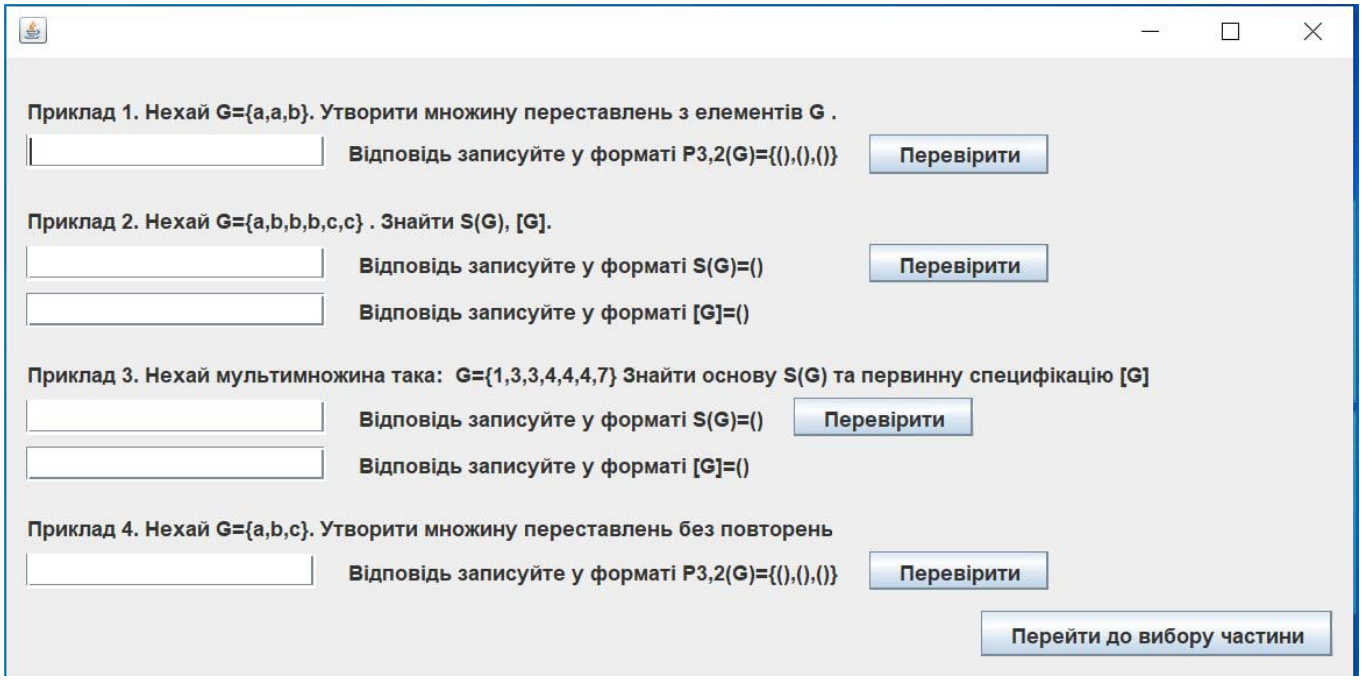

Рисунок 4.3 – Практична сторінка

Тестова частина

1. Вибір "Тестова частина": Тестова частина включає кілька питань з множинним вибором, які допоможуть оцінити знання користувача по темі (див рис. 4.4).
2. Відповідь на питання: Оберіть найвідповідніший варіант відповіді для кожного запитання і натисніть "Перевірити відповіді" для відображення результатів.
3. Завершення тестування: Після завершення тестування результати будуть автоматично відображені, і користувач матиме змогу повернутися до головного меню.

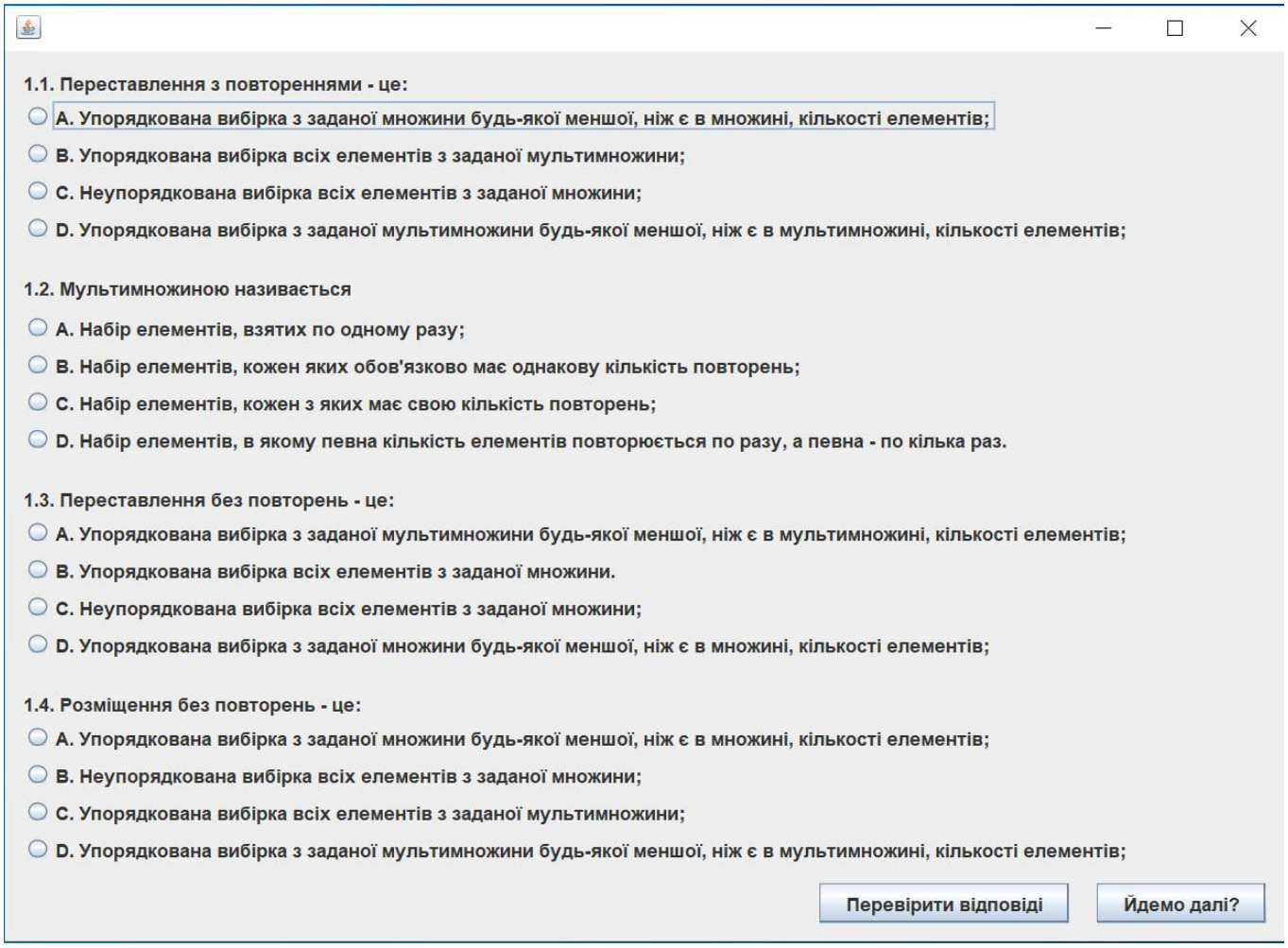

Рисунок 4.4 – Тестова частина

Загальні вказівки

- Програма легка у використанні та інтуїтивно зрозуміла, забезпечена функціями зворотного зв’язку, які допомагають у процесі навчання.
- Використовуйте кнопку виходу для закриття програми в будь-який час.

Ця інструкція служить для ознайомлення з основними можливостями тренажера і має допомогти користувачам ефективно навчатися з його допомогою.

ВИСНОВКИ

У ході виконання цієї дипломної роботи було здійснено повний цикл розробки навчального тренажера для вивчення евклідових комбінаторних множин переставлень. Значна увага приділялася інтеграції теоретичних знань та їх практичному застосуванню через інтерактивні завдання та тести.

Результати роботи включають:

1. Інформаційний огляд: Було проведено аналіз наявних навчальних тренажерів та методик викладання комбінаторної оптимізації. Досліджено різноманітні джерела, які описують алгоритмічне моделювання та використання комбінаторних множин у різних областях науки та техніки.
2. Теоретична розробка: Опрацьовано та систематизовано основні принципи евклідових комбінаторних множин переставлень, їх математичні моделі та алгоритмічне застосування у вирішенні оптимізаційних задач.
3. Розробка програмної частини: Створено тренажер, який включає модулі для теоретичного навчання, виконання практичних завдань, а також блок тестування знань. Програма реалізована на мові Java, що забезпечує високу гнучкість та адаптивність інтерфейсу.
4. Перевірка та тестування: Виконано комплексне тестування тренажера з метою ідентифікації та виправлення можливих помилок, забезпечення стабільності роботи програми та її компонентів.
5. Інструкція користувача: Розроблено докладну інструкцію з використання тренажера, що містить опис всіх функціональних можливостей, поради для ефективного навчання та вказівки щодо розв’язання типових завдань.

Впровадження тренажера у навчальний процес забезпечило студентам доступ до інноваційного інструмента для глибокого розуміння та практичного застосування знань у сфері комбінаторної оптимізації. Завдяки використанню тренажера, показники успішності студентів помітно покращилися, що підтверджує ефективність розробленої навчальної системи та важливість її подальшого розвитку та інтеграції у навчальні курси.

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Schildt, Herbert. Java: a beginner's guide. McGraw-Hill Education, 2022.
2. Придатко О. В. Основи програмування (мовою Java): курс лекцій / О. В. Придатко, О. В. Хлевной, Н. Є. Бурак,. – Львів : ЛДУ БЖД, 2019. – 180 с.
3. Основи програмування (PYTHON, JAVA): лабораторний практикум для студентів спеціальності 122 «Комп‘ютерні науки»; денної та заочної форм навчання / уклад.: О.О. Смотр., О.В. Придатко, І.О. Малець. – Львів, 2019. – 134 с.
4. Черненко Оксана Олексіївна. "Методичні рекомендації щодо оформлення пояснювальних записок". / Черненко Оксана Олексіївна - Вищий навчальний заклад Укоопспілки «Полтавський університет економіки і торгівлі», 2023 – 59 с.
5. Стоян Ю.Г. Теорія і методи евклідової комбінаторної оптимізації: монографія / Ю.Г. Стоян, О.О. Ємець. - К: ІСДО, 1993. - 188 с. – Режим доступу http://dspace.uccu.org.ua/handle/123456789/487 .
6. Ємець О.О. Моделі евклідової комбінаторної оптимізації: монографія / О.О. Ємець, О.О. Черненко. - Полтава: ПУЕТ, 2011. - 204 с. – Режим доступу http://dspace.uccu.org.ua/handle/123456789/354.
7. Шилдт Г. *Повне керівництво Java*. 10-те вид. Видавництво "Діалектика", 2018. 1488 с.
8. Фаулер М. *Рефакторинг: поліпшення існуючого коду*. 2-ге вид. Addison-Wesley Professional, 2018. 448 с.

ДОДАТОК А.

1. MainForm

```
package com.sirik.combinatorialsetstrainer;

public class MainForm extends javax.swing.JFrame {

    public MainForm() {
        initComponents();
        setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("Ласкаво просимо до тренажера комбінаторних множин!");

        jButton1.setText("Теоретична частина");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Практична частина");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton2ActionPerformed(evt);
        }
    });

    jButton3.setText("Тестова частина");
    jButton3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton3ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap(92, Short.MAX_VALUE)
                    .addComponent(jLabel1))
                .addGroup(layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 314,
Short.MAX_VALUE)
                        .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton3, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
                .addGap(91, 91, 91))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addGap(127, 127, 127)
            .addComponent(jButton1)
            .addGap(18, 18, 18)
            .addComponent(jButton2)
            .addGap(18, 18, 18)
            .addComponent(jButton3)
            .addContainerGap(46, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    TheoryForm theoryForm = new TheoryForm();
    theoryForm.setVisible(true);
    this.setVisible(false);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    PracticeForm practiceForm = new PracticeForm();
    practiceForm.setVisible(true);
    this.setVisible(false);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    TestForm testForm = new TestForm();
    testForm.setVisible(true);
    this.setVisible(false);
}

public static void main(String args[]) {
```

```
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new MainForm().setVisible(true);
                }
            });
        }

        // Variables declaration - do not modify
        private javax.swing.JButton jButton1;
        private javax.swing.JButton jButton2;
        private javax.swing.JButton jButton3;
        private javax.swing.JLabel jLabel1;
        // End of variables declaration
}
```

2. TheoryForm

```
package com.sirik.combinatorialsetstrainer;

public class TheoryForm extends javax.swing.JFrame {

    public TheoryForm() {
        initComponents();
        setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTextArea1.setColumns(20);
```

jTextArea1.setRows(5);

jTextArea1.setText("Вступ в комбінаторну оптимізацію\nНехай X - скінчена комбінаторна множина, x - її довільний елемент, x ∈X, а |X| - кількість елементів в X, та нехай є заданий функціонал F: X → R'. \nОсновною задачею комбінаторної оптимізації називають задачу знаходження екстремуму\n\nF = F(x*) = extr F(x) по x ∈X\n\ni екстремалі\n\nx* = arg extr F(x) по x ∈X.\n\nРозв'язком задачі (1), (2) називають пару <F*, x*>. Обидві задачі можуть ставитися незалежно одна від одної, \nтоді під розв'язком задачі (1) розуміють F*, а задачі (2) - x*. \nОсновною задачею мінімізації на комбінаторній множині X називають задачу знаходження\n\nF* = min F(x) по x ∈X, x* = arg min F(x) по x ∈X,\n\nосновною задачею максимізації — відповідно задачу визначення\n\nF* = max F(x) по x ∈X, x* = arg max F(x) по x ∈ X.\n\nПозначимо Jn - множину n перших натуральних чисел, Jn = {1,...,n}, а також позначимо J0 = Jn ∪{0}. Для спрощення формул та єдності форм \nїх представлення важливим є, що в них J0 = Ø.\n\nМультимножиною G = {g1, g2, ..., gn} називають сукупність n елементів g1,...,gn, серед яких є, зазвичай кажучи, однакові (нерозрізнимі). \nМультимножина, всі елементи якої різні, є множиною. Будь-яку мультимножину G можна представити її основою S(G), тобто упорядкованою \nмножиною (кортежем, вектором) S(G) = (e1^j1, e2^j2, ..., em^jm) - всіх різних елементів із G, n ≤ m, (якщо і тільки тоді j1 + j2 + ... + jm = n), \nто G-множина, і перевіряюча специфікація [G] - кортеж кратностей елементів основи мультимножини: k(a) = j, якщо k(a) = j; k(a) = 0, \nякщо елемент a не є елементом G, тоді [G] = (η1, η2,..., ηm,..., ηn) - перша специфікація. Те, що елемент a мультимножини A має кратність k(a) \nзаписують у вигляді a^k(a).\n\nІнше представлення мультимножини: G = {e1^η1, e2^η2,...,em^ηm,...,en^ηn}; η1 +...+ ηn = n, де ηi ≥ 0.\n\nПриклад 1. Нехай мультимножина така: G={1,1,3,3,3,5,6}. Знайдіть основу S(G) та перевіряючу специфікацію [G].\n\nМаємо: S(G)={1,3,5,6}, [G]={2,3,1}.\n\nПриклад 2. Нехай G={a,a,b,b,b,c,c,c}. Тоді S(G)=(a,b,c); [G]=(2;3;3), G={a^2,b^3,c^3}.\n\nДії (операції) над мультимножинами детально розглянуті в [1]. Далі буде підбірка сума мультимножин. Дамо її означення.\n\nНехай A - мультимножина з основою S(A)=(x,y,z, ...) і кратностями (k(a),k(y),k(a),k(z),...), а B - мультимножина з основою S(B)=(x,y,z, ...) \nі кратностями (k(x),k(y),k(z),k(y),...), де k(x)≥0, k(y)≥0, k(z)≥0, k(y)≥0, k(g)≥0, k(b)≥0, k(z)≥0 тощо. \nТоді під сумою A+B мультимножин A та B розуміють мультимножину з основою \nS(A+B)=S(A)∪S(B) і крайностями {k(a)+k(x),k(a)+k(y),k(a)+k(z),...}={k(a)+k(x)+k(g)+k(y),k(a)+k(z)+k(g),...}. \nСуму мультимножин Ai будемо позначати ∑_i=1^N A_i .\n\nМультимножина B з основою S(B) називається підмультимножиною мультимножини A з основою S(A), якщо S(B)⊆S(A), і для кожного елементу \nа∈S(B) виконується нерівність k(b)≤k(a), де k(a) - кратність елемента a в мультимножині. Позначається це тим же знаком ⊆, що і для множин, \nтобто B⊆A.\n\nПриклад 3. Для G з прикладу 1 G={1,1,3,3,3,5,6} - мультимножина G; G2={1,3,7,7} - не є*

підмультимножиною G; \nG3={1,3,6,6} - не є підмультимножиною G.\n\nПриклад 4. Нехай A={a,b,b,c,c}, B={a,a,a,b,b}, C={a,b}, G - з прикладу 2. Розглянемо чи є серед цих мультимножин підмультимножини? \nA⊆G: S(A)=(a,b,c), S(A)⊆S(G), [A]=(1,2,2), [G]=(2,3,3), отже A⊆G. Розглянемо B та G: S(B)=(a,b) S(B)⊆S(G). \nПозначимо для зручності порівняння [B] та [G] кратність c в B нулем: [B]=(3,2,0), де k(a)=3; [G]=(2,1,3), де k(c)=3; отже B⊄G.\n\nПорівняємо C та G.\nS(C)=(a,b)⊆S(G); k(c)≤k(z)=k(a); k(c)≤k(b)=k(b); отже C⊆G. Наведемо, як в [1], k-елементну мультимножину B у мультимножині A k-вибіркою, \nтобто B - k-вибірка, якщо B⊆A, |B|=k.\n\nНехай k та n - натуральні константи, gj - дійсні числа j в Jn, i в In, а G = {g1,...,gn} - мультимножина з основою S(G) = (e1,...,en), первинною \nспецифікацією [G] = (n1,...,nn), n1 >= 1 для всіх i в Jn, n1 + ... + nm = n, n2 <= k. \nНе порушуючи загальності, можна вважати, що ni <= k для всіх i в Jn. \nЗа означенням основи мультимножини ei не дорівнює ej для всіх i, j в In, j не дорівнює i. Якщо G - мультимножина, то існує принаймні одне число \ni в Jn, що ni > 1, а з цього випливає, що n > n. Розглянемо упорядкування k-вибірки у мультимножині G:\n\n(e[g1],...,e[gi],...,e[gi],...,e[gk]),\n\nде gi, e[gi] в G, і не дорівнює j якщо e[gi] не дорівнює e[gj] для всіх i, j в Ik.\n\nОзначення. Множину E, елементами якої є k-вибірки e(bar) = (e1,...,ek), e в E = (e1,...,ek) вважають (3) з мультимножини G наявним \nевклідовим комбінаторним множиною, якщо 3 умови: gj не дорівнює k там, де ej не дорівнює ei, виливаючись e не дорівнює e' \n(коротка назва: e-множина).\n\nПри цьому умові e, e' в E може виконуватись умова e' = em = ei = e'i, j не дорівнює m, j,m в Ik. Евклідова комбінаторна множина E має таку \nвластивість: два елементи e в E, що належать множині E, відмінні один від одного, якщо вони виризняються порядком слідування символів, \nщо їх утворюють, або самих символів. Тобто евклідові комбінаторні множини можна розглядати як множини точок евклідового простору R^k.");

```
        jTextArea1.setCaretPosition(0);
        jScrollPane1.setViewportView(jTextArea1);

        jButton1.setText("Йдемо далі?");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(684, Short.MAX_VALUE)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 840, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 582,
Short.MAX_VALUE)
                .addGap(34, 34, 34)
                .addComponent(jButton1)
                .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainForm().setVisible(true);
    }

    public static void main(String args[]) {
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

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

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

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

java.util.logging.Logger.getLogger(TheoryForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TheoryForm().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
```

```
    // End of variables declaration
}
```

3. PracticeForm

```
package com.sirik.combinatorialsetstrainer;

import javax.swing.JOptionPane;

public class PracticeForm extends javax.swing.JFrame {

    public PracticeForm() {
        initComponents();
        setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextField1 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jButton2 = new javax.swing.JButton();
        jTextField2 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jButton3 = new javax.swing.JButton();
        jTextField3 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jButton4 = new javax.swing.JButton();
        jTextField4 = new javax.swing.JTextField();
```

```
jTextField5 = new javax.swing.JTextField();
jButton5 = new javax.swing.JButton();
jTextField6 = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jButton1.setText("Перевіриmи");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jLabel1.setText("Приклаd 1. Нехай G={a,a,b}. Утворити множину переставлень з елементів G .");

jLabel2.setText("Віdповіdь записуйте у формаmі P3,2(G)={(),(),()}");

jLabel3.setText("Приклаd 2. Нехай G={a,b,b,b,c,c} . Знайти S(G), [G]. ");

jLabel4.setText("Віdповіdь записуйте у формаmі S(G)=()");

jButton2.setText("Перевіриmи");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
```

```
        });

        jTextField2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField2ActionPerformed(evt);
            }
        });

        jLabel5.setText("Приклад 3. Нехай мультимножина така:  G={1,3,3,4,4,4,7} Знайти основу S(G) та первинну специфікацію [G]");

        jLabel6.setText("Відповідь записуйте у форматі S(G)=()");

        jButton3.setText("Перевірити");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        jTextField3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField3ActionPerformed(evt);
            }
        });

        jLabel7.setText("Приклад 4. Нехай G={a,b,c}. Утворити множину переставлень без повторень");

        jLabel8.setText("Відповідь записуйте у форматі P3,2(G)={(),(),()}");

        jButton4.setText("Перевірити");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
```

```
    }
});

jTextField4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField4ActionPerformed(evt);
    }
});

jTextField5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField5ActionPerformed(evt);
    }
});

jButton5.setText("Перейти до вибору частини");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});

jTextField6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField6ActionPerformed(evt);
    }
});

jLabel9.setText("Відповідь запишуйте у форматі [G]=()");

jLabel10.setText("Відповідь запишуйте у форматі [G]=()");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
```

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addComponent(jLabel3)
                        .addComponent(jLabel5)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(jLabel6)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jButton3))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(jLabel10))
                        .addComponent(jLabel7)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jTextField4)
                                .addGap(18, 18, 18)
                                .addComponent(jLabel8)
                                .addGap(18, 18, 18)
                                .addComponent(jButton4))
                            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addGroup(layout.createSequentialGroup()
                                        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
172, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(12, 12, 12)
                                .addComponent(jLabel2))
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
172, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(18, 18, 18)
                                .addComponent(jLabel4)))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jButton2)
                            .addComponent(jButton1)))))
                .addGap(0, 118, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jButton5))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel9)
                        .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(22, 22, 22)
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2)))
            .addGap(18, 18, 18)
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel4)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel9))
            .addGap(18, 18, 18)
            .addComponent(jLabel5)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel6)
                .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel10))
```

```
            .addGap(18, 18, 18)
            .addComponent(jLabel7)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel8)
                .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(12, 12, 12)
            .addComponent(jButton5)
            .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String answer = jTextField1.getText().trim();
        if(answer.equals("P3,2(G)={(a,a,b),(a,b,a),(b,a,a)}")) {
            JOptionPane.showMessageDialog(this, "Вірно! P3,2(G)={(a,a,b),(a,b,a),(b,a,a)}",
"Результат", JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Невірно. Спробуйте ще раз.", "Результат",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        String SAnswer = jTextField2.getText().trim();
        String GAnswer = jTextField6.getText().trim();
        if(SAnswer.equals("S(G)=(a,b,c)") && GAnswer.equals("[G]=(1,2,3)")) {
            JOptionPane.showMessageDialog(this, "Вірно!", "Результат",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Невірно. Спробуйте ще раз.", "Результат",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        String SAnswer = jTextField3.getText().trim();
        String GAnswer = jTextField5.getText().trim();
        if(SAnswer.equals("S(G)=(1,3,4,7)") && GAnswer.equals("[G]=(1,2,3,1)")) {
            JOptionPane.showMessageDialog(this, "Вірно!", "Результат",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Невірно. Спробуйте ще раз.", "Результат",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        String userAnswer = jTextField4.getText().trim();
        if(userAnswer.equals("P3(G)={(a,b,c),(a,c,b),(b,a,c),(b,c,a),(c,a,b),(c,b,a)}")) {
            JOptionPane.showMessageDialog(this, "Вірно!", "Результат",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Невірно. Спробуйте ще раз.", "Результат",
JOptionPane.ERROR_MESSAGE);
        }
```

```
}

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jTextField5ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new MainForm().setVisible(true);
}

private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt) {

}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> {
        new PracticeForm().setVisible(true);
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
```

```
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    // End of variables declaration
}
```

4. TestForm

```
package com.sirik.combinatorialsetstrainer;

import javax.swing.JOptionPane;

public class TestForm extends javax.swing.JFrame {

    public TestForm() {
        initComponents();
        setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        buttonGroup1 = new javax.swing.ButtonGroup();
        buttonGroup2 = new javax.swing.ButtonGroup();
        buttonGroup3 = new javax.swing.ButtonGroup();
        buttonGroup4 = new javax.swing.ButtonGroup();
        jLabel1 = new javax.swing.JLabel();
```

```
jRadioButton1 = new javax.swing.JRadioButton();
jRadioButton2 = new javax.swing.JRadioButton();
jRadioButton3 = new javax.swing.JRadioButton();
jRadioButton4 = new javax.swing.JRadioButton();
jLabel2 = new javax.swing.JLabel();
jRadioButton5 = new javax.swing.JRadioButton();
jRadioButton6 = new javax.swing.JRadioButton();
jRadioButton7 = new javax.swing.JRadioButton();
jRadioButton8 = new javax.swing.JRadioButton();
jLabel3 = new javax.swing.JLabel();
jRadioButton9 = new javax.swing.JRadioButton();
jRadioButton10 = new javax.swing.JRadioButton();
jRadioButton11 = new javax.swing.JRadioButton();
jRadioButton12 = new javax.swing.JRadioButton();
jLabel4 = new javax.swing.JLabel();
jRadioButton13 = new javax.swing.JRadioButton();
jRadioButton14 = new javax.swing.JRadioButton();
jRadioButton15 = new javax.swing.JRadioButton();
jRadioButton16 = new javax.swing.JRadioButton();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("1.1. Переставлення з повтореннями - це: ");

buttonGroup1.add(jRadioButton1);
jRadioButton1.setText("А. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів; ");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});
```

buttonGroup1.add(jRadioButton2);

jRadioButton2.setText("B. Упорядкована вибірка всіх елементів з заданої мультимножини;");

buttonGroup1.add(jRadioButton3);

jRadioButton3.setText("C. Неупорядкована вибірка всіх елементів з заданої множини;");

buttonGroup1.add(jRadioButton4);

jRadioButton4.setText("D. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;");

jLabel2.setText("1.2. Мультимножиною називається");

buttonGroup2.add(jRadioButton5);

jRadioButton5.setText("A. Набір елементів, взятих по одному разу;");

buttonGroup2.add(jRadioButton6);

jRadioButton6.setText("B. Набір елементів, кожен яких обов'язково має однакову кількість повторень;");

buttonGroup2.add(jRadioButton7);

jRadioButton7.setText("C. Набір елементів, кожен з яких має свою кількість повторень;");

buttonGroup2.add(jRadioButton8);

jRadioButton8.setText("D. Набір елементів, в якому певна кількість елементів повторюється по разу, а певна - по кілька раз.");

jLabel3.setText("1.3. Переставлення без повторень - це: ");

buttonGroup3.add(jRadioButton9);

jRadioButton9.setText("A. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;");

buttonGroup3.add(jRadioButton10);

jRadioButton10.setText("B. Упорядкована вибірка всіх елементів з заданої множини.");

```
buttonGroup3.add(jRadioButton11);
jRadioButton11.setText("C. Неупорядкована вибірка всіх елементів з заданої множини;");

buttonGroup3.add(jRadioButton12);
jRadioButton12.setText("D. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів;");

jLabel4.setText("1.4. Розміщення без повторень - це:");

buttonGroup4.add(jRadioButton13);
jRadioButton13.setText("A. Упорядкована вибірка з заданої множини будь-якої меншої, ніж є в множині, кількості елементів;");

buttonGroup4.add(jRadioButton14);
jRadioButton14.setText("B. Неупорядкована вибірка всіх елементів з заданої множини;");

buttonGroup4.add(jRadioButton15);
jRadioButton15.setText("C. Упорядкована вибірка всіх елементів з заданої мультимножини;");

buttonGroup4.add(jRadioButton16);
jRadioButton16.setText("D. Упорядкована вибірка з заданої мультимножини будь-якої меншої, ніж є в мультимножині, кількості елементів;");

jButton1.setText("Йдемо далі?");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Перевірити відповіді");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
```

```
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jButton2)
                        .addGap(18, 18, 18)
                        .addComponent(jButton1)
                        .addContainerGap())
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jRadioButton6)
                            .addComponent(jRadioButton7)
                            .addComponent(jRadioButton8)
                            .addComponent(jRadioButton1)
                            .addComponent(jRadioButton2)
                            .addComponent(jRadioButton3)
                            .addComponent(jRadioButton4)
                            .addComponent(jRadioButton5)
                            .addComponent(jLabel1)
                            .addComponent(jLabel2)
                            .addComponent(jRadioButton13)
                            .addComponent(jRadioButton10)
                            .addComponent(jRadioButton11)
                            .addComponent(jRadioButton12)
                            .addComponent(jRadioButton15)
                            .addComponent(jLabel3)
```

```
                .addComponent(jRadioButton14)
                .addComponent(jLabel4)
                .addComponent(jRadioButton16)
                .addComponent(jRadioButton9))
            .addGap(33, 98, Short.MAX_VALUE))))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton1)
            .addComponent(jButton2))
        .addContainerGap())
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton4)
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(jRadioButton5)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jRadioButton6)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jRadioButton7)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
                .addComponent(jRadioButton8))
            .addComponent(jLabel2))
        .addGap(18, 18, 18)
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton9)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton10)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton11)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton12)
        .addGap(18, 18, 18)
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton13)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton14)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton15)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jRadioButton16)
        .addContainerGap(48, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new MainForm().setVisible(true);
```

```
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        int score = 0;
        String resultMessage;

        if (jRadioButton2.isSelected()) score++;
        if (jRadioButton7.isSelected()) score++;
        if (jRadioButton10.isSelected()) score++;
        if (jRadioButton13.isSelected()) score++;

        resultMessage = "Ваш результат: " + score + " з " + "4";
        JOptionPane.showMessageDialog(this, resultMessage, "Результати тесту",
JOptionPane.INFORMATION_MESSAGE);
    }


    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TestForm().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.ButtonGroup buttonGroup2;
    private javax.swing.ButtonGroup buttonGroup3;
    private javax.swing.ButtonGroup buttonGroup4;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
```

```
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton10;
    private javax.swing.JRadioButton jRadioButton11;
    private javax.swing.JRadioButton jRadioButton12;
    private javax.swing.JRadioButton jRadioButton13;
    private javax.swing.JRadioButton jRadioButton14;
    private javax.swing.JRadioButton jRadioButton15;
    private javax.swing.JRadioButton jRadioButton16;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    private javax.swing.JRadioButton jRadioButton5;
    private javax.swing.JRadioButton jRadioButton6;
    private javax.swing.JRadioButton jRadioButton7;
    private javax.swing.JRadioButton jRadioButton8;
    private javax.swing.JRadioButton jRadioButton9;
    // End of variables declaration
}
```