МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра дискретного аналізу та інтелектуальних систем

**Звіт з лабораторної роботи №9**

**“Множина”**

Роботу виконав:

**Тимчишин Ярема Андрійович**

Студент групи Пмі-13

Перевірив:

**\_,**

\_ Львівського національного

університету імені Івана Франка

Львів – 2022

**ЗМІСТ**

ВСТУП 3

РОЗДІЛ 1 6

***1.1. Створення класу “множина”*** 6

***1.2. Метод для добавляння елементу в множину*** 6

***1.3. Метод для вилучення елементу з множини*** 6

***1.4. Метод для перевірки чи є вже елемент у множині*** 7

***1.5. Функції для отримання max та min елементів, а також розміру*** 7

***1.6. Функція для об’єднання двох множин*** 7

РОЗДІЛ 2 9

***2.1. Ініціалізація двох множин*** 9

***2.2. Три тести для перевірки справності функцій з’єднання, перетину та віднімання множин відповідно*** 9

***2.3. Результати тестів*** 10

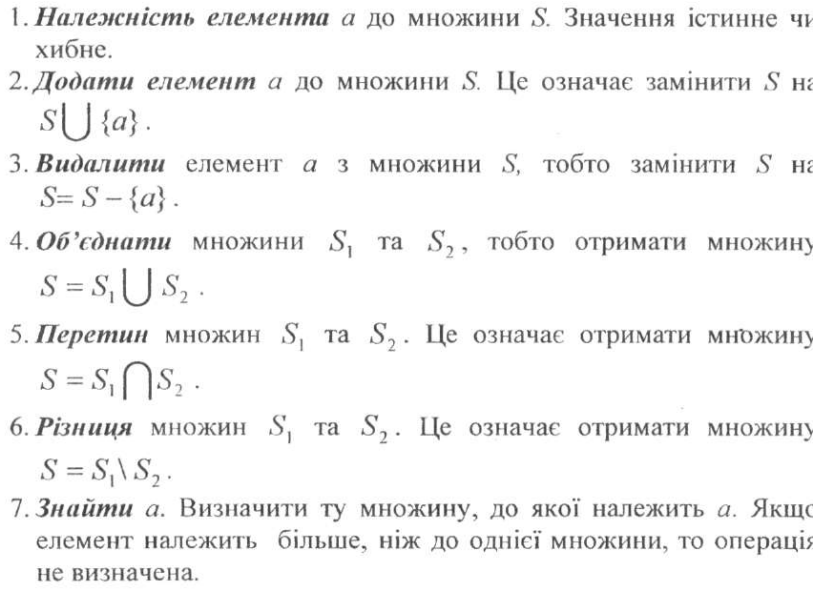
ВИСНОВКИ 10

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 10

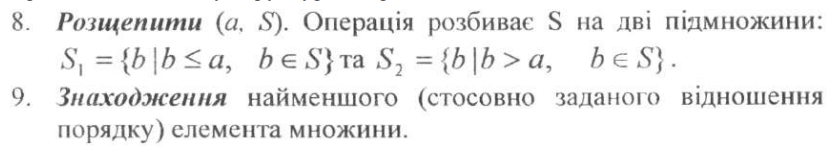
# ВСТУП

Множини є основними об'єктами вивчення дискретної математики. Множина - це невпорядкована сукупність деяких об'єктів, які називають елементами множини і над якими не визначено жодного відношення. Над множиною можна виконувати певні операції, які визначають функціональну специфікацію структури.

Над множинами визначено такі основні операції:



Деякі автори вводять операції, які стосуються впорядкованих множин. У них передбачається, що елементи множини **S** знаходяться у деякому лінійному порядку, який позначають звичайними символами відношення. Над такими множинами вводять операції.



Багато задач, які трапляються на практиці, можна звести до однієї чи кількох гіідзадач, таких, що їх можна сформулювати як послідовність основних операцій.

**Відображення множин у структури зберігання**

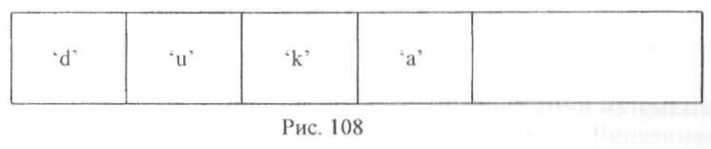
Існує два основних підходи до відображення множин в пам'ять комп'ютера.

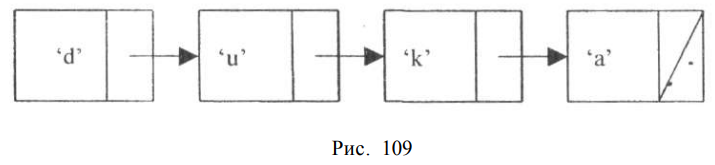
1. Для кожного елемента множини зберігати його опис аналогічно до математичного способу задания множини переліком її елементів.
2. Визначити всі потенційно можливі елементи множини, а потім для всякої підмножини такої універсальної множини вказувати для кожного можливого елемента, чи належить він цій підмножині, чи ні. Цей спосіб аналогічний гіредикатній формі задания множини у математиці.

У першому випадку множину зручно зберігати у вигляді вектора або лінійного списку, у другому — використовують бітові вектори.

Розглянемо ці два способи для зображення множини А з чотирьох елементів A={'d', 'u', 'k', 'а'}.

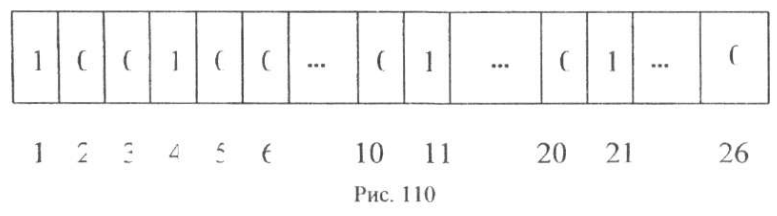
Цю множину можна зобразити вектором, для якого вказати кількість елементів, (див. рис. 108), або ланцюговим списком (див. рис. 109).

****

****

У процесі відображення множини у бітовий вектор усі можливі елементи множини треба перенумерувати, наприклад, від 1 до N. У пам'яті виділяється поле з N бітів і належність елемента до множини відзначається встановленням в 1 і-того біт, якщо і-тий можливий елемент присутній у множині. Інакше цей біт встановлюється в 0.

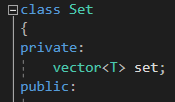
Наприклад, для множини малих букв латинського алфавіту вектор може складатись з 26 бітів, а множину А можна зобразити так (див. рис. 110).

****

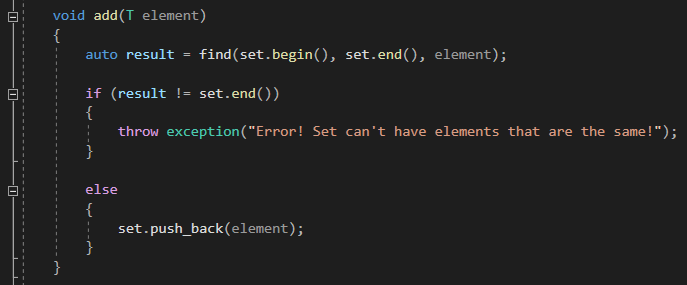
# РОЗДІЛ 1

**Написання головної програми**

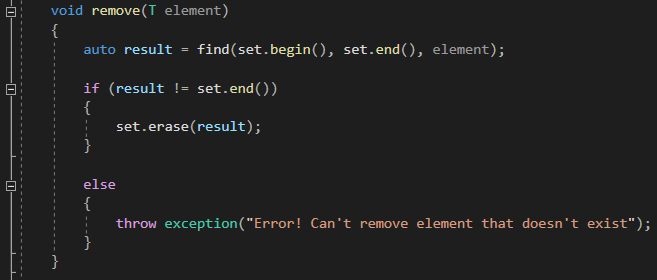
## *1.1. Створення класу “множина”*

****

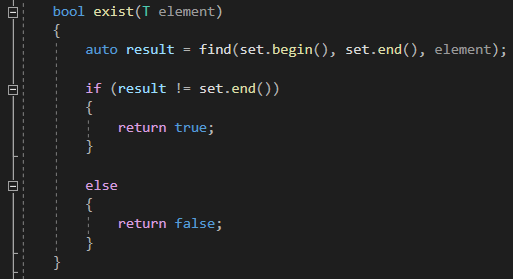
## *1.2. Метод для добавляння елементу в множину*



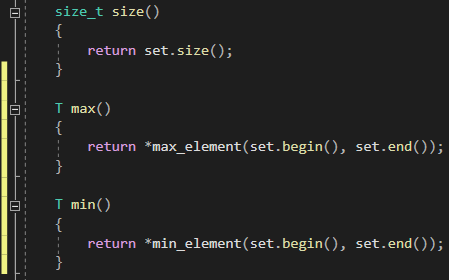
## *1.3. Метод для вилучення елементу з множини*



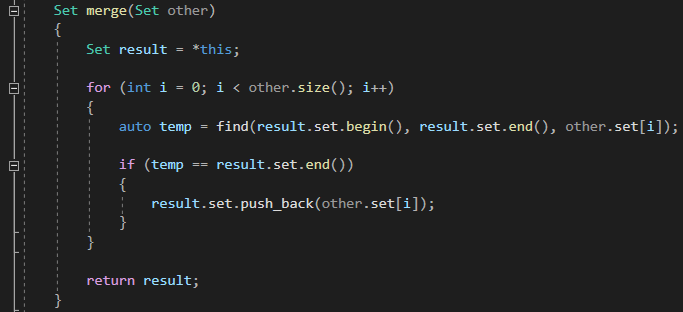
## *1.4. Метод для перевірки чи є вже елемент у множині*



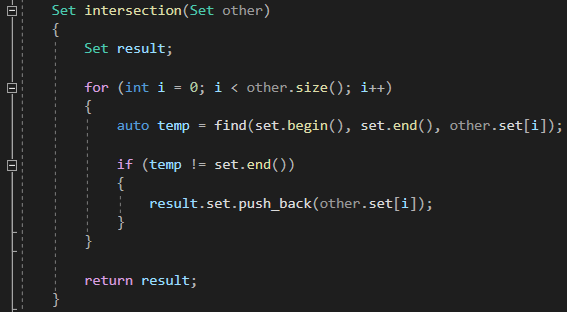
## *1.5. Функції для отримання max та min елементів, а також розміру*



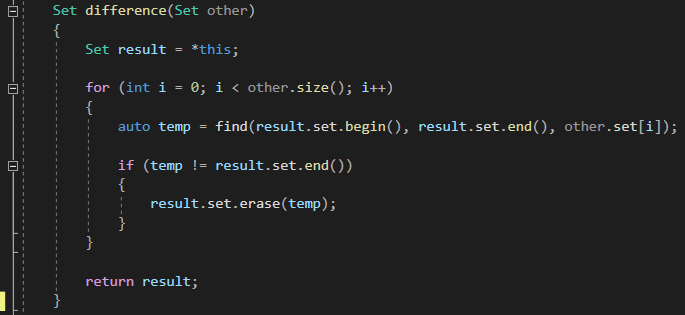
## *1.6. Функція для об’єднання двох множин*



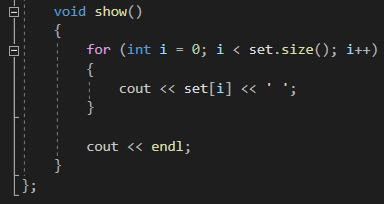
***1.7. Функція для знаходження перетину двох множин***

****

***1.8. Функція для віднімання однієї множини від іншої***



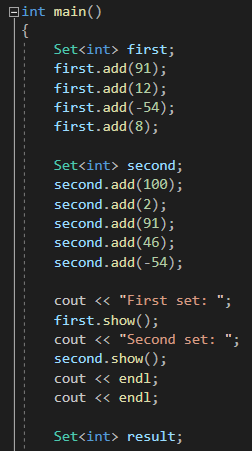
***1.9. Функція для відображення множини***



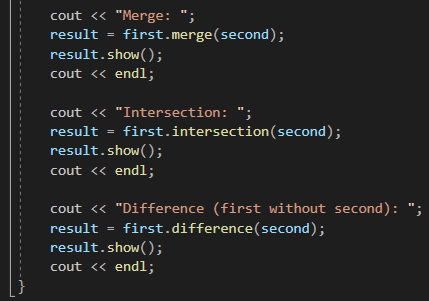
# РОЗДІЛ 2

**Перевірка справності написаної програми**

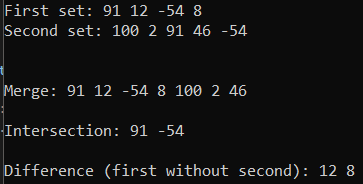
## *2.1. Ініціалізація двох множин*



## *2.2. Три тести для перевірки справності функцій з’єднання, перетину та віднімання множин відповідно*



## *2.3. Результати тестів*



# ВИСНОВКИ

У результаті виконання роботи:

1. Розроблено програму для роботи з множиною.
2. Перевірено правильність виконання написаної програми.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. Костів. Структури даних: Навч. посібн. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2005. – 146 с