# Міністерство освіти і науки України Львівський національний університет імені Івана Франка

**Факультет прикладної математики та інформатики**

Кафедра програмування

Лабораторна робота №3

# Таблиця

з курсу “Алгоритми та структури даних”

Виконав: студент групи ПМІ-16 Процай Іван Сергійович

Львів – 2024

**Мета:** метою даної роботи є дослідження та розробка класу Table для реалізації структури даних "Таблиця". Таблиця є динамічною структурою даних, яка дозволяє зберігати пари ключ-значення та здійснювати операції додавання, видалення, зміни значення за ключем та перевірки наявності ключа.

**Принцип роботи:** Клас Table реалізований у вигляді бінарного дерева пошуку, де кожен вузол представляє собою пару ключ-значення. Всі вузли впорядковані за значеннями ключів. Клас має методи для додавання, видалення, перевірки наявності та отримання значення за ключем. Кожен вузол містить посилання на лівого та правого нащадка, що дозволяє шукати, додавати та видаляти елементи ефективно.

**Приклад:** У цьому прикладі я створюю таблицю з наступними даними:

Ключ Значення

101 Luca

100 Giovanni

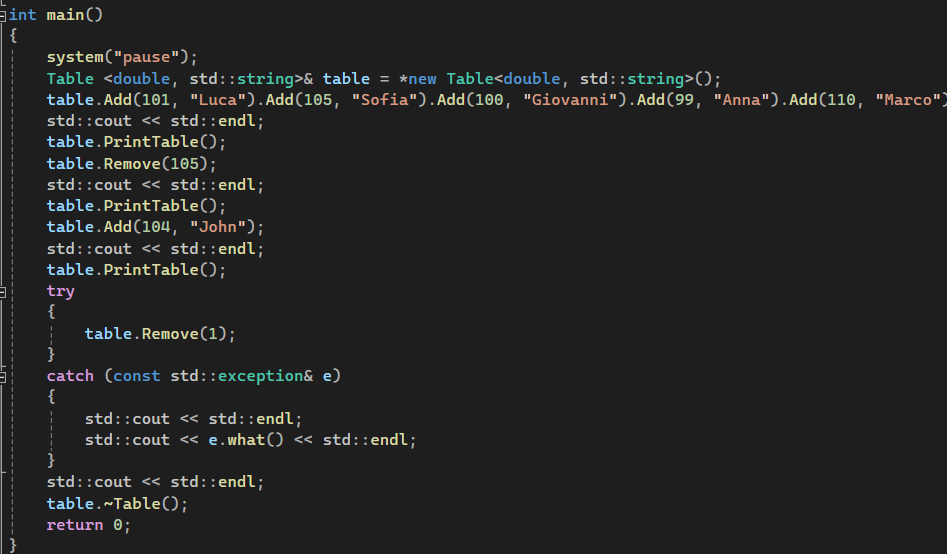
99 Anna

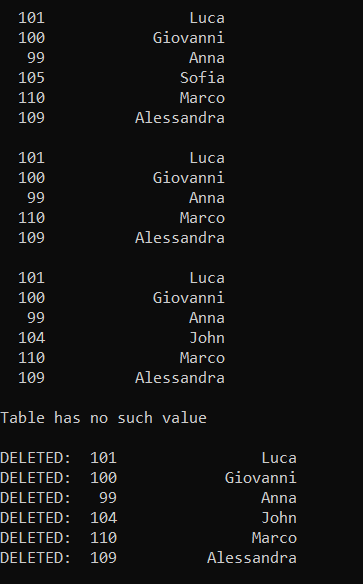
105 Sofia

110 Marco

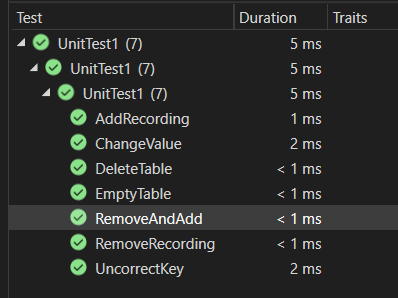
109 Alessandra

Пізніше я виводжу її на екран. За допомогою методу Remove(105) вилучаю із таблиці значення під ключем 105, виводжу змінену таблицю, додаю до таблиці нове значення та друкую змінену таблицю. Щоб продемонструвати коректність роботи винятків видаляю з таблиці неіснуючий ключ, що запускає вийняток std::runtime\_error, пізніше викликаю деструктор, який повідомляє про знищення таблиці.





**Тести**

****

Наведені вище тести є частиною набору тестів для класу Table, які використовуються для перевірки правильності роботи різних функціональностей цього класу:

1. EmptyTable: Цей тест перевіряє, чи правильно обробляється видалення елемента з порожньої таблиці. Очікується виняток, що підтверджує відсутність значення в таблиці.
2. AddRecording: Даний тест перевіряє коректність додавання запису до таблиці та перевіряє, чи правильно обробляються перевірки наявності запису з вказаним ключем.
3. RemoveRecording: Цей тест перевіряє правильність видалення записів з таблиці за допомогою методу Remove. Після видалення перевіряється, чи були вони дійсно видалені.
4. RemoveAndAdd: Тест перевіряє, чи коректно працює видалення та додавання елементів до таблиці, а також чи правильно обробляються зміни в структурі після цих операцій.
5. ChangeValue: Цей тест перевіряє коректність зміни значення за ключем. Перевіряється, чи правильно змінюється значення, що зберігається за ключем.
6. UncorrectKey: Тест перевіряє, як обробляється спроба отримання значення за неправильним ключем. Очікується виняток, оскільки вказаного ключа немає в таблиці.
7. DeleteTable: Даний тест перевіряє правильність видалення всієї таблиці. Після видалення очікується, що таблиця буде порожньою.

**Висновок:** У результаті виконання даної роботи було успішно реалізовано клас Table, що відповідає вимогам завдання. Використовуючи цей клас, можна ефективно працювати зі структурою даних "Таблиця", виконуючи різноманітні операції з додавання, видалення, зміни значень та перевірки наявності ключів. Тестування, проведене за допомогою набору тестів, підтвердило коректність роботи класу та його відповідність специфікації.