Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №10

**На тему:**

“Стандартна бібліотека шаблонів. Контейнери та алгоритми”

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ-18

Юшкевич А.І.

**Прийняв:**

асис. каф. ПЗ

Далявський В.С.

« … » … 2023 р.

∑ = \_\_\_\_\_\_\_\_\_\_\_\_

Львів – 2023

**Тема**: Стандартна бібліотека шаблонів. Контейнери та алгоритми.

**Мета**: Навчитись використовувати контейнери стандартної бібліотеки шаблонів та вбудовані алгоритми.

Теоретичні відомості

**Контейнери**

Контейнери – це об’єкти, які зберігають інші об’єкти. Вони керують алокацією та деалокацією цих об’єктів через конструктори, деструктори, операції вставки та видалення.

Бібліотека контейнерів містить узагальнену колекцію шаблонів класів та алгоритмів, що дозволяють програмістам легко реалізувати поширені структури даних, такі як черги, списки чи стеки. Починаючи із C++11 існує три категорії контейнерів:

1. послідовні контейнери;
2. асоціативні контейнери;
3. невпорядковані асоціативні контейнери (починаючи із C++11).

Кожен контейнер яких спроектовано для підтримки певних операцій.

Контейнер керує місцем, необхідним для зберігання його елементів, надає функції-члени для доступу до них або безпосередньо, або через ітератори (об’єкти із властивостями схожими до вказівників).

Більшість контейнерів мають хоча б кілька схожих функцій та функціонал. Вибір кращого контейнера залежить від конкретної задачі та від ефективності реалізації функціоналу потрібного для розв’язування цієї задачі.

**Послідовні контейнери**

Послідовні контейнери реалізують структури даних, що надають можливість послідовного доступу до однотипних елементів.

Нижче наведені шаблони класів послідовних контейнерів

**Асоціативні контейнери**

Асоціативні контейнери реалізують впорядковані структури даних, у яких швидкий пошук за ключем (алгоритмічна складність - O(log n)).

Кожен контейнер параметризується типом ключа та типом об’єкта для порівняння.

Нижче наведені шаблони класів асоціативних контейнерів.

Завдання

1. Змінити реалізацію класу із лабораторної роботи №9 відповідно до варіанту таким чином:

* 1. для збереження елементів використати **послідовний контейнер** чи клас-рядок який найбільше підходить для вирішення задачі. Обґрунтувати вибір;
  2. змінити реалізацію існуючих функцій та операторів, використовуючи алгоритми із бібліотеки;
  3. додатково реалізувати функції члени, використовуючи алгоритми із бібліотеки.

1. Продемонструвати роботу із **асоціативним** та **невпорядкованим асоціативним контейнером** згідно із варіантом:
   1. вставку значень із вже присутнім та відсутнім ключем;
   2. успішний та неуспішний пошук за ключем;
   3. успішне та неуспішне видалення за ключем;
   4. послідовну ітерацію (проходження) по контейнеру для зчитування значень; 3. Продемонструвати роботу із **адаптером** згідно із варіантом:
   5. вставку значень;
   6. вичитку значень;
2. Продемонструвати роботу шаблону класу, контейнерів та адаптерів таким чином (на вибір): a. За допомогою тестів Google Test.

b. За допомогою інтерактивної програми із демонстрацією її роботи.

1. Оформити звіт до лабораторної роботи. В звіті, окрім стандартних пунктів, сформувати таблицю використаних в програмі методів для кожного залученого std-контейнера. Таблиця має містити назву контейнера, назву метода, пояснення дії метода.

Варіант №13:

|  |  |
| --- | --- |
| 13 | Шаблон класу CTable<typename T> – таблиця із колонками та рядками. Містить опис колонок та рядки із даними типу T. Колонки та рядки додаються динамічно.  Використати такі алгоритми:   * std::copy – для об’єднання рядків та декартового добутку * std::for\_each – для оператора виведення у потік Реалізувати такі функції: * розвертання порядку колонок – використати std:: reverse * повернення std::vector<std::vector<std::string>> із значеннями всіх рядків |

# Код програми



# Результат виконання

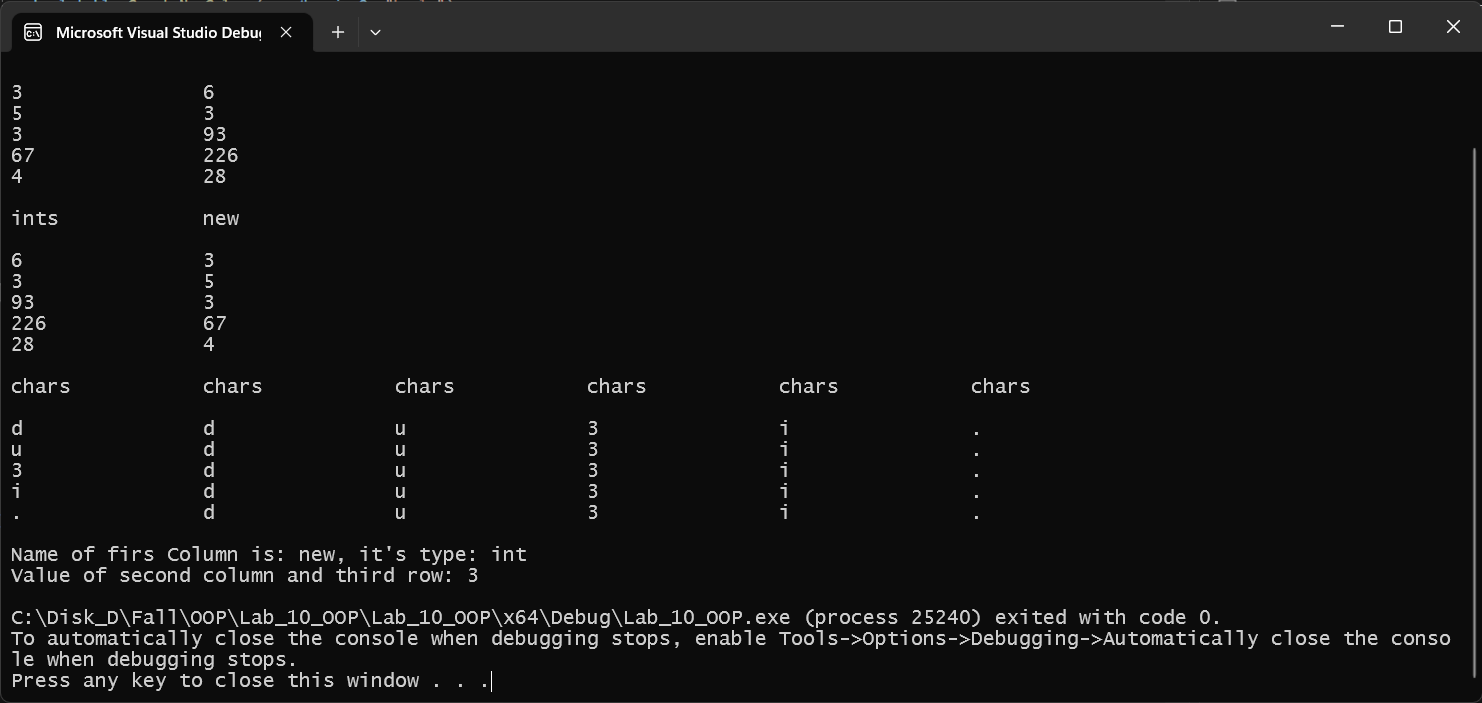


Рис. 1. Виконання тестів для функцій класу CTable<typename T>.

# Висновок

Під час виконання лабораторної роботи я навчився використовувати контейнери стандартної бібліотеки шаблонів та вбудовані алгоритми.