

Завдання  
до лабораторної роботи № 4  
по курсу "(Об'єктно-орієнтоване) Програмування"  
кафедра комп'ютерних технологій, ДНУ  
2020/2021

**Тема:** "Узагальнене програмування"

**Зміст**

- Постановка задачі
- Індивідуальні варіанти завдань
- Література
- Примітки

**Постановка задачі**

Розробити програму – консольний застосунок для ОС Windows, що демонструє можливості мови C++, а також його стандартної бібліотеки по роботі з потоками, контейнерами та ітераторами.

Використання всіх вищезгаданих можливостей є обов'язковим в кожному з варіантів індивідуальних завдань.

В кожному з нижченаведених варіантів завдань Вами повинні бути реалізовані основні можливості бібліотеки стандартних шаблонів: ітератор(и), пошук, упорядкування, сума або щось аналогічне зі згоди викладача.

При написанні програми слід враховувати необхідність забезпечення її зборки у двох режимах:

1. Створення виконуваного файлу із підключенням/використанням типів і алгоритмів бібліотеки стандартних шаблонів (STL).
2. Створення виконуваного файлу без застосування STL.

Наприклад: Ваше завдання – втілити алгоритм швидкого сортування для контейнеру «вектор»; в такому разі Ви мусите продемонструвати, як Ваша програма працює із Вашою реалізацією вектора, а також як вона працює із стандартною реалізацією вектора з бібліотеки STL. Увага: Ви пишете одну програму, а не дві. Необхідні зміни вихідного тексту для переналаштування з/на STL бажано робити за допомогою директив умовної компіляції.

**Індивідуальні варіанти завдань**

1. (16.8) Взявши за основу клас Itor [1, 16.2.1], подумайте, як забезпечити ітератори для перебору у зростаючому порядку, в порядку зменшення, для перебору контейнера, що зда-

тний змінюватися при ітерації, та перебору контейнеру, що не змінюється (immutable). Організуйте цей набір контейнерів так, щоб користувач міг взаємозамінюючим способом використовувати ітератори, що забезпечують достатню функціональність алгоритмів. Мінімізуйте повторення фрагментів коду в реалізації цих контейнерів. Які ще ітератори можуть знадобитися користувачу? Проаналізуйте переваги та недоліки вашого підходу.

2. (16.13) Напишіть класи, щоб `Vector` з 16.2.1 та `Vector` з 16.2.2 використовувались як стандартні контейнери.
3. (16.14) Напишіть шаблон, що реалізує контейнер з тими ж функціями-членами та типами членів, як і стандартний `vector` (деякого нестандартного, неучбового) контейнерного типу. Не модифікуйте вже існуючий контейнерний тип. Як би ви стали працювати з функціональністю, яка запропонована нестандартним `vector`, але не реалізована стандартним?
4. (17.2) Багато телефонних номерів не вміщуються в тип `long`. Напишіть тип `phone_number` та клас, що забезпечує набір корисних операцій з контейнером телефонних номерів типу `phone_number`.
5. (17.3) Напишіть програму, яка заносила б різні слова в файл в алфавітному порядку. Зробіть дві версії: нехай в одній слово буде просто послідовністю символів, яка не містить символів-роздільників, а в іншій – послідовністю символів, яка не містить символів-нелітер.
6. (17.19) Реалізуйте шаблон функції, яка виконує злиття двох контейнерів.
7. (17.21) Прочитайте послідовність пар (ім'я, значення) та створіть сортований список четвірок (ім'я, сума, середнє значення, медіана). Роздрукуйте цей список.
8. (17.6) Визначте чергу використовуючи (тільки) два стека.
9. (18.9) Напишіть набір об'єктів-функцій, які виконують бітові логічні операції над своїми операндами. Перевірте ці об'єкти на векторах з елементами `char`, `int`.
10. (19.3) Розробіть клас `reverse_iterator`, який описаний в [1, 19.2.5].
11. (19.5) Розробіть клас `istream_iterator`, який описаний в [1, 19.2.6].
12. (19.7) Розробіть клас `Checked_iter`, який описаний в [1, 19.3].
13. (19.8) Придумайте та реалізуйте допоміжний клас, який міг би працювати як замісник контейнера, забезпечуючи для користувачів повний інтерфейс контейнера. Його реалізація повинна містити покажчик на контейнер та реалізації контейнерних операцій з перевіркою діапазону.
14. (18.10) Напишіть зв'язувач `binder3()` [1, 18.4.4], який би зв'язував другий та третій аргументи трьохаргументної функції для отримання унарного предиката. Наведіть приклад, де функція `binder3()` є корисною функцією.

15. (18.13) Реалізуйте алгоритм `move()` в стилі `copy()` таким чином, щоб вхідна та вихідна послідовність могли перекриватися. Намагайтесь забезпечити прийнятну ефективність, задаючи в якості аргументів ітератори з довільним доступом.
16. (18.19) Для двонаправлених ітераторів не існує алгоритму `sort()`. Існує гіпотеза, що копіювання послідовності в вектор та наступне сортування буде швидше, ніж сортування послідовності за допомогою двонаправлених ітераторів. Реалізуйте універсальне сортування для двонаправлених ітераторів та перевірте цю гіпотезу.
17. (18.20) Уявіть, що ви ведете записи про групу спортсменів-риболовів. Для кожної виловленої риби запишіть її вигляд, довжину, вагу, дату улову, фамілію риболова і т.ін. Відсортуйте та роздрукуйте записи у відповідності з різними критеріями. Підказка: `inplace_merge`.
18. (21.3) Визначить тип `Name_and_address` (ім'я та адреса). Визначте для нього оператори `<<` та `>>`. Забезпечте можливість копіювання потоку об'єктів типу `Name_and_address`, в який ви вставили стільки помилок, скільки змогли придумати (наприклад, помилки формату та передчасні кінці рядків). Обробіть ці помилки таким чином, щоб гарантувати, що функція копіювання прочитає більшість правильно відформатованих об'єктів `Name_and_address`, хоча при введенні будуть чергуватися „погані” значення з „добрими”.
19. (21.5) Напишіть декілька функцій для запиту та читання інформації різного типу. Наприклад: цілі числа, числа з плаваючою точкою, імена файлів, поштові адреси, анкетні дані та т.ін. Намагайтесь зробити їх захищеними від помилок.
20. (21.8) Напишіть „мініатюрну” систему потокового введення/ виведення, яка містить класи `istream`, `ostream`, `ifstream`, `ofstream`, функції `operator <<()` та `operator >>()` для цілих чисел та такі операції, як `open()` та `close()`, для файлів.
21. Напишіть програму, в якій би клас „Матриці: сортування та інші перетворення” використовувався як стандартний контейнер.
22. Напишіть програму, в якій би клас „Відображення (асоціативні масиви)” використовувався як стандартний контейнер.
23. Напишіть програму, в якій би клас „Динамічні списки” використовувався як стандартний контейнер.
24. Напишіть програму, в якій би класи „Багаточлени” використовувався як стандартний контейнер.
25. Напишіть програму, в якій би клас „Дерева довільного вигляду” використовувався як стандартний контейнер.
26. Напишіть програму, в якій би клас „Дерева синтаксичного розбору” використовувався як стандартний контейнер.

27. Напишіть програму, в якій би клас „Текстовий документ” використовувався як стандартний контейнер.
28. Напишіть програму, в якій би клас „Файлова система” використовувався як стандартний контейнер.

### **Література**

1. Страуструп Б. Язык программирования C++, 3-е издание. - СПб.; М.: "Невский диалект" - "Издательство БИНОМ", 1999. - 991 с.
2. Пышкин Е.В. Основные концепции и механизмы объектно-ориентированного программирования. – СПб.: БХВ-Петербург, 2005. – 640 с.
3. Скотт Мейерс. Эффективное использование STL. – СПб.: «Питер», 2002. – 224 с.
4. Андрей Александреску Современное проектирование на C++: Обобщенное программирование и прикладные шаблоны проектирования. – М.: «Вильямс», 2004. – 336 с.
5. Мэтью Г. Остерн. Обобщённое программирование и STL: Использование и наращивание стандартной библиотеки шаблонов в C++. – СПб.: Невский Диалект, 2004. – 544 с.
6. Дэвид Вандевурд, Николаи М. Джосаттис. Шаблоны C++: справочник разработчика. – М.: Издательский дом «Вильямс», 2003. – 544 с.