# Лабораторна робота №2.

**Мета:**

● Набуття навичок розробки власних контейнерів.

● Використання ітераторів.

● Тривале зберігання та відновлення стану об’єктів.

● Ознайомлення з принципами серіалізації/десеріалізації об’єктів.

● Використання бібліотек класів користувача.

# Вимоги

1. Розробити клас-контейнер, що ​ ітерується​ для збереження початкових даних Вашого варіанту завдання з попередньої роботи ​ (Прикладні задачі. Список з 1-15 варіантів)​ у вигляді ​ масиву рядків​ з можливістю додавання, видалення і зміни елементів.

2. В контейнері реалізувати та продемонструвати наступні методи:

○ String toString()​ повертає вміст контейнера у вигляді рядка;

○ void add(String string)​ додає вказаний елемент до кінця контейнеру;

○ void clear()​ видаляє всі елементи з контейнеру;

○ boolean remove(String string)​ видаляє перший випадок вказаного елемента з контейнера;

○ Object[] toArray()​ повертає масив, що містить всі елементи у контейнері;

○ int size()​ повертає кількість елементів у контейнері;

○ boolean contains(String string)​ повертає ​ true​ , якщо контейнер містить вказаний елемент;

○ boolean containsAll(Container container)​ повертає ​ true​ , якщо контейнер містить всі елементи з зазначеного у параметрах;

○ public Iterator<String> iterator()​ повертає ітератор відповідно до Interface Iterable​ .

3. В класі ітератора відповідно до ​ Interface Iterator​ реалізувати методи:

○ public boolean hasNext()​ ;

○ public String next()​ ;

○ public void remove()​ .

4. Продемонструвати роботу ітератора за допомогою циклів ​ while ​ и ​ for each ​ .

5. Забороняється використання контейнерів (колекцій) і алгоритмів з ​ Java Collections Framework​ .

6. Реалізувати і продемонструвати тривале зберігання/відновлення розробленого контейнера за допомогою ​ серіалізації/десеріалізації​ .

7. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення одного варіанту задачі ​ (Прикладні задачі. Список з 1-15 варіантів)​ з сусіднім номером. 1 міняється з 2, 2 з 3, 3 з 4, 4 з 5 і т.д. Останній, 15 міняється з 1 варіантом і далі аналогічно.

8. Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.

9. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.

10.Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

# Розробник

Зінько Павло, КН-108

# Задача

Написати власний клас(контейнер) у якому зберігаються методи, які вказані вище. Створити меню для роботи з користувачем, реалізувати порівняння, сортування та пошук елементів у контейнері.

# Опис програми

Програма створює ваш власний контейнер з різними методами, які допоможуть організувати роботу зі стрічками. Створено меню для поліпшення роботи з користувачем.

# Засоби ООП

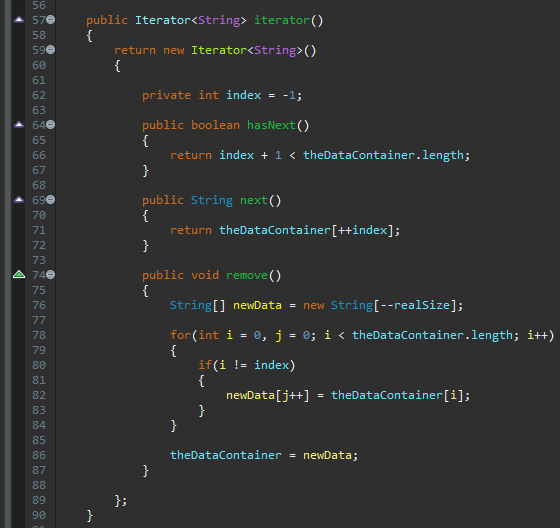
Були використані різні класи та методи, структури даних та модифікатори доступу.

# Ієрархія та структура класів

1. Клас Main, який викликає та провіряє всі створені методи у контейнері.
2. Клас Container , зберігає та обробляє всю інформацію за допомогою методів, щодо стрічок введених користувачем.

# Важливі фрагменти програми.

Ітератор:



# Варіанти використання

Можна використовувати для спрощення роботи зі стрічками.

# ВИСНОВКИ

У ході роботи ми навчились створювати власний контейнер з різними методом та ітератором.

Розробили консольну програму та забезпечили діалоговий режим роботи з користувачем для демонстрації та тестування рішення