

1.Тема роботи

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.

1.1 Розробник

-Бердник Данило Ігорович

-КН-108

-1 Варіант

2.Загальне завдання

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 вибірккову 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.

2 Опис програми

2.1 Засоби ООП:

Я створив додатковий до задачі з минулої лабораторної роботи клас-контейнер що ітерується, він імплементує інтерфейси `Iterable<String>`, `Serializable`

Завдяки цьому клас може ітеруватитись і записувати дані в файл. В середині класу головним об'єктом є масив, над яким і проводяться операції.

2.2 Ієрархія та структура класів:

Клас main – головний клас. Залежно від вводу користувача викликає методи класу Menu і класу StringIterator. Клас Menu – виводить в консоль меню дій. Функції цього класу збирають до купи функції класу StringFuncт Клас StringFuncт – клас у якому виконуються всі під дії з 1ої лабораторної з рядками . StringIterator – клас-контейнер у якому доданий новий функціонал для лабораторної №2.

2.3 Важливі фрагменти програми:

```
public Iterator<String> iterator() {  
    return new InnerIterator();  
}  
  
private class InnerIterator implements Iterator<String>{  
    int cursor = -1;  
    @Override  
    public boolean hasNext() {  
        if(stringsIter[cursor+1] != null)  
            return cursor < stringsIter.length -1;  
    }  
}
```

```

        else
            return false;
    }

    @Override
    public String next() {
        return stringsIter[++cursor];
    }

    public void remove() {
        for(String i : stringsIter) {
            if(stringsIter[cursor].equals(i)) {
                for(int j = cursor; j<size-1;j++) {
                    stringsIter[j] = stringsIter[j+1];
                }
                size--;
                break;
            }
        }
    }
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

```

a - create new container
b - container operating menu
c - display the container by index
d - show list of your containers
e - end the program

```

```

a
created
a - create new container
b - container operating menu
c - display the container by index
d - show list of your containers
e - end the program
a
created
a - create new container
b - container operating menu
c - display the container by index
d - show list of your containers
e - end the program
b
enter index of container:
2
a - convert container into string
b - adds new string to the end of the container
c - clears the container data
d - removes element by name
e - returns array that contains all elements in container
f - returns number of elements in container
g - returns true if container contains given element
h - compares 2 containers
i - sort elements
k - iterates through container
l - calculations from the previous lab
m - close menu
s - serialise container
p - deserialise container

```

```

b
enter string to add:
fjffj
added
a - convert container into string
b - adds new string to the end of the container
c - clears the container data
d - removes element by name
e - returns array that contains all elements in container
f - returns number of elements in container
g - returns true if container contains given element
h - compares 2 containers
i - sort elements
k - iterates through container
l - calculations from the previous lab
m - close menu
s - serialise container
p - deserialise container
b
enter string to add:
aowwww
added
a - convert container into string
b - adds new string to the end of the container
c - clears the container data
d - removes element by name
e - returns array that contains all elements in container
f - returns number of elements in container
g - returns true if container contains given element
h - compares 2 containers
i - sort elements
k - iterates through container
l - calculations from the previous lab
m - close menu
s - serialise container
p - deserialise container

```

```

1
Group 1
aowwww Length - 6
Group 2
fjffj Length - 4
Group 3
4jfjffjff Length - 8

```

```
s
Serialized
a - convert container into string
b - adds new string to the end of the container
c - clears the container data
d - removes element by name
e - returns array that contains all elements in container
f - returns number of elements in container
g - returns true if container contains given element
h - compares 2 containers
i - sort elements
k - iterates through container
l - calculations from the previous lab
m - close menu
s - serialise container
p - deserialise container
p
Deserialized:
fjffj
aowwww
4jffjffjf
```