

1. Тема роботи

Розробка власних контейнерів. Ітератори. Серіалізація/десеріалізація об'єктів.
Бібліотека класів користувача

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 ВИМОГИ

1.1 Розробник: Мокрик Ярослав Любомирович; КН-108; варіант 19 (завдання 4).

1.2

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.

1.3 Задача

Ввести текст. У кожному слові тексту k-ту літеру замінити заданим символом. Якщо k більше довжини слова, видати повідомлення та заміну не виконувати. Початкові дані та результат вивести у вигляді таблиці.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Я застосував 4 класи, кожен з яких містить статичні і звичайні методи. Таким чином, я можу створювати об'єкти класу ітератора (який я сам написав), ітерувати їх, додавати чи віднімати елементи і змінювати їх. До того ж, я використовував клас `StringBuilder`, щоб створити рядок, який можна змінювати в ході роботи програми. Також, я створив методи для серіалізації та десеріалізації даних, щоб їх можна було довго зберігати.

2.2 Ієрархія та структура класів

Клас `Main` - головний клас. Керує роботою всієї програми і викликає статичні методи інших класів.

Клас `Menu` - клас меню. Керує показом головного меню, приймає ввід користувача і викликає відповідні методи.

Клас `StringMangager` - клас менеджера рядків. Керує вводом рядка, числа і символа, серіалізацією даних, а також застосовує цикл `foreach` з класом `StringIterator`. Генерує новий рядок відповідно до вимог завдання.

Клас `StringIterator` - клас ітератора `String`. Імплементує всі необхідні методи, а також містить внутрішній клас `InnerIterator`, який керує лічильником для ітерації.

2.3 Важливі фрагменти програми

Це - методи серіалізації та десеріалізації даних.

```
public static void serialize() {
    try {
        FileOutputStream fOutput = new FileOutputStream("file.txt");
        ObjectOutputStream sOutput = new
ObjectOutputStream(fOutput);
        sOutput.writeObject(si);
        sOutput.flush();
        sOutput.close();
    } catch (Exception e) {
        e.printStackTrace();
        System.exit(1);
    }
    System.out.println("Object written successfully.");
}
```

```

    }
    public static void deSerialize() {
        try {
            FileInputStream flnput = new FileInputStream("file.txt");
            ObjectInputStream slnput = new ObjectInputStream(flnput);
            si = (StringIterator) slnput.readObject();
            slnput.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
        System.out.println("Object read successfully.");
    }
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

Головне меню:

```

a. Add new string
b. Delete string
c. Edit string
d. Input N and C
e. Serialize strings
f. Deserialize strings
g. Output
h. Quit
Your option:

```

Редагування даних:

```

Your option: c
Enter string to delete: Hello world
Enter string to add: Hello everyone

```

Виведення в форматі таблиці:

```

Your option: g
|-----|
Goodbye world
Go.db.e .or.d
|-----|
Hello everyone
He.lo.ev.ry.ne
|-----|

```

Повідомлення аргументу -h (-help):

```
Author: Yaroslav Mokryk, Group KN-108
Task:
Ввести текст. У кожному слові тексту k-ту літеру замінити заданим
символом. Якщо k більше довжини слова, видати повідомлення та заміну
не виконувати. Початкові дані та результат вивести у вигляді таблиці.
Menu items:
Input data - input the data (string, number and character) through the console
Output data - output a table with the original and result strings
Compute - replace every k-th character with the input character
Quit - exit the program
Command line arguments:
-h or -help - display this help message
-d or -debug - run the program in debug mode
```

Обчислення результату в режимі -d (-debug):

```
H
He
Hel
HelI
HelIo
HelIo
HelIo w
HelIo wI
HelIo wIr
HelIo wIrl
HelIo wIrld
HelIo wIrldI
HelIo wIrldIh
HelIo wIrldIhe
HelIo wIrldIhel
```

ВИСНОВКИ Я навчився розробляти власні контейнери, ітератори. Дізнався про серіалізацію/десеріалізацію об'єктів і бібліотеки класів користувача.