

6. Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача

Мета:

- Тривале зберігання та відновлення стану об'єктів.
- Ознайомлення з принципами серіалізації/десеріалізації об'єктів.
- Використання бібліотек класів користувача.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Шарма Олександр Раджнішович
- НТУ “ХПІ” КІТ-1196
- Варіант 25

1.2 Загальне завдання

1. Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі д.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибірккову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

1.3 Задача

Ввести декілька рядків. Упорядкувати, а потім вивести рядки за алфавітом (перший пріоритет) та в порядку зростання їх довжини (другий пріоритет).

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі відсутні об'єктно-орієнтовані методи.

2.2 Ієрархія та структура даних

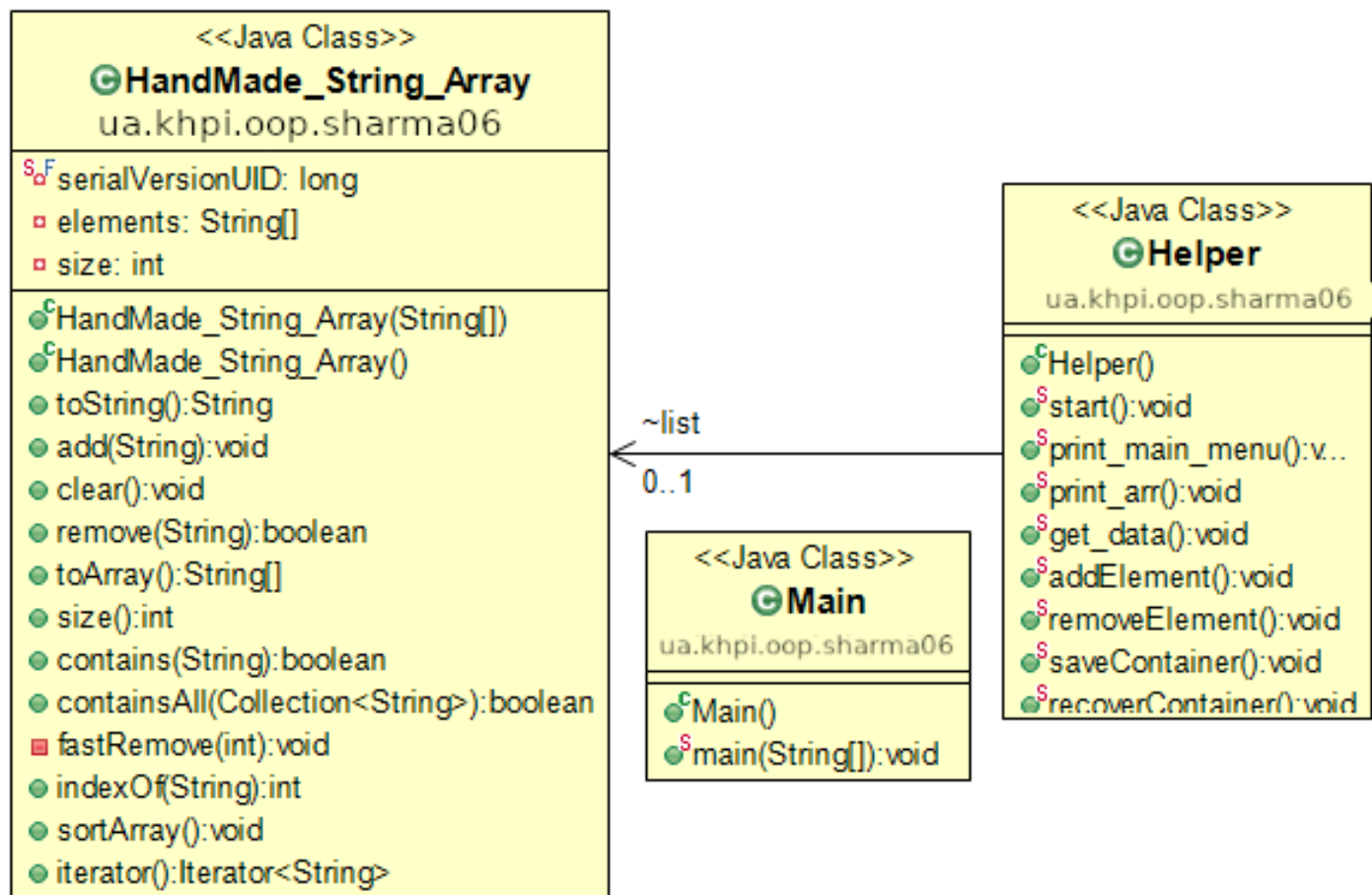


Рисунок 1 – діаграма класів

2.3 Важливі фрагменти програми

```

}
static public void saveContainer() throws IOException {
    FileOutputStream outputStream = new FileOutputStream("C:\\Users\\{Dean}\\Serialized\\savedContainer.txt");
    ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);
    objectOutputStream.writeObject(list);
    objectOutputStream.close();
}
static public void recoverContainer() throws IOException, ClassNotFoundException {
    FileInputStream fileInputStream = new FileInputStream("C:\\Users\\{Dean}\\Serialized\\savedContainer.txt");
    ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
    list = (HandMade_String_Array) objectInputStream.readObject();
}
}
    
```

Рисунок 1 – Реалізовані методи серіалізації\десеріалізації

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма створена для тестування роботи серіалізації та десеріалізації

```
main (4) Java Application, C:\Program Files
1.Enter data
2.Show current data
3.Sort by alphabet
4.Add data
5.Remove some element
6.Find index of element
7.Clear all
8.Save data
9.Recover data
10.Exit
2
Your array is:
1.Enter data
2.Show current data
3.Sort by alphabet
4.Add data
5.Remove some element
6.Find index of element
7.Clear all
8.Save data
9.Recover data
10.Exit
```

Рисунок 2 – результати роботи до виконання десеріалізації

```
9
1.Enter data
2.Show current data
3.Sort by alphabet
4.Add data
5.Remove some element
6.Find index of element
7.Clear all
8.Save data
9.Recover data
10.Exit
2
Your array is:
  cat
  cow
  kangaroo
  kid
  vombat
  wolf
```

Рисунок 3 – успішна десеріалізація даних

ВИСНОВКИ

В даній лабораторній роботі було розроблено методи серіалізації та десеріалізації, було набуто навичок роботи з ними, та використання бібліотек класів користувача.