

Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача

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

Вимоги

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 л.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибірково 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.

Розробник: Рябов Олексій Володимирович КІТ119а №18.

Опис програми

Засоби ООП: клас, метод класу.

Структура класів: один публічний клас Main, один утилітарний клас Task, один клас колекція Container

Важливі фрагменти програми:

```
import var18.Helper;  
  
import java.io.FileInputStream;  
  
import java.io.FileOutputStream;  
  
import java.io.IOException;  
  
import java.io.ObjectInputStream;  
  
import java.io.ObjectOutputStream;  
  
import java.util.Iterator;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args ) throws IOException,  
    ClassNotFoundException {
```

```
    //    Container container = new Container("Hi user. ","This lab aims to show how I  
    can deal with the container development problem. ","All this is needed so that I can  
    keep strings with palindromes safe and sound.");
```

```
        Container container = new Container();
```

```
        String command = "";
```

```
        String str = "";
```

```
        do
```

```
        {
```

```
            System.out.println("Доступні команди\r\n" +
```

```
                "1 - заповнення контейнера даними          - \r\n" +
```

```
                "2 - виведення змісту контейнера          - \r\n" +
```

```
                "3 - додавання нового елементу в контейнер - \r\n" +
```

```
                "4 - видалення елементу з контейнера      - \r\n" +
```

```
                "5 - пошук елементу                        - \r\n" +
```

```
                "6 - індивідуальне завдання                - \r\n" +
```

```
                "7 - сортування контейнеру                - \r\n" +
```

```
                "8 - порівняння контейнерів                - \r\n" +
```

```
                "9 - серіалізація                          - \r\n" +
```

```
                "10 - десіалізація                         - \r\n" +
```

```
                "11 - завдання іншого студента            - \r\n" +
```

```
+      "0 - завершення програми(видалення контейнера автоматичне) - \r\n"

      "");
System.out.println("Введіть команду: ");
Scanner scan = new Scanner(System.in);
command = scan.nextLine();
switch (command) {
case "1": {
    container = Create_container();
    break;
}
case "2": {
    if(container.size()!=0)
    {
        System.out.println("Виведення змісту контейнера на екран");
        System.out.println(container.toString());
    }
    else {
        System.out.println("Ви ще не створили контейнер");
    }
    break;
}
case "3": {
    if(container.size()!=0)
    {
        System.out.println("Введіть новий рядок");
        str = scan.nextLine();
        container.add(str);
    }
}
```

```
else {  
    System.out.println("Ви ще не створили контейнер");  
}  
  
break;  
}  
case "4": {  
    if(container.size()!=0)  
    {  
        System.out.println("Введіть рядок що видалиться");  
        str = scan.nextLine();  
        System.out.println("Результат видалення елементу - " +  
container.remove(str));  
    }  
    break;  
}  
case "5": {  
    if(container.size()!=0)  
    {  
        System.out.println("Введіть рядок який потрібно перевірити");  
        str = scan.nextLine();  
        System.out.println("Результат пошуку елементу в контейнері" +  
container.contains(str));  
    }  
    else {  
        System.out.println("Ви ще не створили контейнер");  
    }  
    break;  
}  
case "6": {
```

```
        if(container.size()!=0)
        {
            Task t = new Task();
            t.start(container.toString());
        }
        else {
            System.out.println("Ви ще не створили контейнер");
        }
        break;
    }
    case "7": {
        if(container.size()!=0)
        {
            container.Sort();
        }

        else {
            System.out.println("Ви ще не створили контейнер");
        }
        break;
    }
    case "8": {
        if(container.size()!=0)
        {
            Container new_cont = Create_container();
            container.containsAll(new_cont);
        }
        else {
            System.out.println("Ви ще не створили контейнер");
```

```
}

break;
}
case "9": {

    FileOutputStream fs = new FileOutputStream("Serial.ser");
    ObjectOutputStream os = new ObjectOutputStream(fs);
    os.writeObject(container);
    os.close();
    System.out.println("Serialization successful");

    break;
}
case "10": {

    FileInputStream fis = new FileInputStream("Serial.ser");
    ObjectInputStream ois = new ObjectInputStream(fis);
    Container var = (Container) ois.readObject();
    container = var;
    ois.close();

    break;
}
case "11": {

    Helper.countVowelsAndConsonants(container.toString());
    break;
```

```

    }
    case "0": {
        System.out.println("Кінець програми");
        container.clear();
        break;
    }
    default:
        System.out.println("Ви помилилися командой");
        break;
    }

} while (!(command.equals("0")));
}

```

```

public static Container Create_container()

```

```

{
    Scanner scan = new Scanner(System.in);
    String enter_data = "";
    System.out.println("Введіть кількість елементів у контейнері");
    enter_data = scan.nextLine();
    int size = Integer.parseInt(enter_data);
    System.out.println("
        + "Введіть елементи - ");
    String [] arr = new String[size];
    for (int i = 0; i < size; i++) {
        System.out.print(i+1 + " елемент - ");
        enter_data = scan.nextLine();
    }
}

```

```

    arr[i] = enter_data;
}

Container cont = new Container(arr);

return cont;
}

```

Результати роботи

```

Доступні команди
1 - заповнення контейнера даними -
2 - виведення змісту контейнера -
3 - додавання нового елементу в контейнер -
4 - видалення елементу з контейнера -
5 - пошук елементу -
6 - індивідуальне завдання -
7 - сортування контейнеру -
8 - порівняння контейнерів -
9 - серіалізація -
10 - десеріалізація -
11 - завдання іншого студента -
0 - завершення програми(видалення контейнера автоматичне) -

Введіть команду:
1
Введіть кількість елементів у контейнері
3
Введіть елементи -
1 елемент - 1
2 елемент - 2
3 елемент - 3
Доступні команди
1 - заповнення контейнера даними -
2 - виведення змісту контейнера -
3 - додавання нового елементу в контейнер -
4 - видалення елементу з контейнера -
5 - пошук елементу -
6 - індивідуальне завдання -
7 - сортування контейнеру -
8 - порівняння контейнерів -
9 - серіалізація -
10 - десеріалізація -
11 - завдання іншого студента -
0 - завершення програми(видалення контейнера автоматичне) -

Введіть команду:
2
Виведення змісту контейнера на екран
1 2 3

```



```
Доступні команди
1 - заповнення контейнера даними -
2 - виведення змісту контейнера -
3 - додавання нового елемента в контейнер -
4 - видалення елемента з контейнера -
5 - пошук елемента -
6 - індивідуальне завдання -
7 - сортування контейнеру -
8 - порівняння контейнерів -
9 - серіалізація -
10 - десіалізація -
11 - завдання іншого студента -
0 - завершення програми(видалення контейнера автоматичне) -
```

Введіть команду:

3

Введіть новий рядок

1341234

```
Доступні команди
1 - заповнення контейнера даними -
2 - виведення змісту контейнера -
3 - додавання нового елемента в контейнер -
4 - видалення елемента з контейнера -
5 - пошук елемента -
6 - індивідуальне завдання -
7 - сортування контейнеру -
8 - порівняння контейнерів -
9 - серіалізація -
10 - десіалізація -
11 - завдання іншого студента -
0 - завершення програми(видалення контейнера автоматичне) -
```

Введіть команду:

4

Введіть рядок що видалиться

2

Результат видалення елемента - true

-

Доступні команди

- 1 - заповнення контейнера даними -
- 2 - виведення змісту контейнера -
- 3 - додавання нового елементу в контейнер -
- 4 - видалення елемнту з контейнера -
- 5 - пошук елементу -
- 6 - індивідуальне завдання -
- 7 - сортування контейнеру -
- 8 - порівняння контейнерів -
- 9 - серіалізація -
- 10 - десіалізація -
- 11 - завдання іншого студента -
- 0 - завершення програми(видалення контейнера автоматичне) -

Введіть команду:

6
Lines are greater than average:

1
Line lenght:1

3
Line lenght:1
Lines less than average:

Доступні команди

- 1 - заповнення контейнера даними -
- 2 - виведення змісту контейнера -
- 3 - додавання нового елементу в контейнер -
- 4 - видалення елемнту з контейнера -
- 5 - пошук елементу -
- 6 - індивідуальне завдання -
- 7 - сортування контейнеру -
- 8 - порівняння контейнерів -
- 9 - серіалізація -
- 10 - десіалізація -
- 11 - завдання іншого студента -
- 0 - завершення програми(видалення контейнера автоматичне) -

Введіть команду:

11

Sentence	Vowels	Consonant	Result
1 3	0	0	1 3

Доступні команди

1 - -----

Висновки

Оволодів серіалізацією та десеріалізацією, навчився підключати сторонні бібліотеки та jar файли.