

## 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

- Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
- Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі [HYPERLINK "https://oophpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/"](https://oophpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/) [HYPERLINK "https://oophpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/"](https://oophpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/) №3 з іншим студентом (визначає викладач).
- Прогонювати послідовну та вибірку обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
- Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
- Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

**Розробник:** Єлєсін Артем Олександрович

**Група :** КІТ-1196

### Опис програми

**Засоби ООП:** клас, метод класу.

**Структура класів:** один публічний клас Main, один утилітарний клас Helper, один клас колекція MyCollection

**Важливі фрагменти програми:**

```

public static void Menu() {
    boolean prz = true;
    Scanner scan = new Scanner(System.in);
    MyCollection collection = new MyCollection();
    MyCollection temp = new MyCollection();
    while(prz){
        System.out.println("1.Enter word\n2.Show
data\n3.Start\n4.Start method other
student\n5.Save\n6.Download\n7.Sorting\n8.Find\n9.Compare with
saving\n0.End\nInput command id: ");
        switch (scan.nextInt()) {
            case 1:
                System.out.println("Enter word: ");
                scan.nextLine();
                collection.add(scan.nextLine());
                break;
            case 2:
                System.out.println(collection.toString());
                break;
            case 3:
                Helper.start(collection.toString());
                break;
            case 4:
                VowelsOrConsonants.splitTextInSentenses(new
StringBuffer(collection.toString()));
                break;
            case 5:

```

```
System.out.println("Enter file name: ");
scan.nextLine();
try {
    collection.Save(scan.nextLine());
} catch (IOException e) {
    e.printStackTrace();
}
break;
```

case 6:

```
System.out.println("Enter file name: ");
scan.nextLine();
try {
    collection.Download(scan.nextLine());
} catch (ClassNotFoundException | IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
break;
```

case 7:

```
collection.sort();
break;
```

case 8:

```
scan.nextLine();
System.out.println("Enter data:");
int index = collection.find(scan.nextLine());
if(index==-1)
```

```

        System.out.println("Element don`t found");
    else
        System.out.println("Element index in mass: "+index);
    break;
case 9:
    System.out.println("Enter file name: ");
    scan.nextLine();
    try {
        temp.Download(scan.nextLine());
    } catch (ClassNotFoundException | IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    if(collection.equals(temp))
        System.out.println("Are equal");
    else
        System.out.println("Not equal");
    break;
    default:
        System.out.println("End program");
        prz = false;
    break;
}
}
scan.close();
}

```

```

public void Save(String fileName) throws IOException {
    FileOutputStream outputStream= new FileOutputStream(fileName);
    ObjectOutputStream objectOutputStream = new
ObjectOutputStream(outputStream);
    objectOutputStream.writeObject(this);
    objectOutputStream.close();
}

public void Download(String fileName) throws IOException,
ClassNotFoundException{
    FileInputStream inStream= new FileInputStream(fileName);
    ObjectInputStream objectInStream = new
ObjectInputStream(inStream);
    MyCollection temp = (MyCollection) objectInStream.readObject();
    this.setMass(temp.getMass());
    objectInStream.close();
}

```

## **Результати роботи**

```
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
1
Enter word:
aaaa
1.Enter word
2.Show data|
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
1
Enter word:
ddd
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
1
```

```
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
```

Input command id:

4

Sentence	Vowels	Consonant	Result	
aaaa ddd c	4	4	draw	

```
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
```

Input command id:

5

Enter file name:

text.txt

```
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
```

Input command id:

1

```
Enter word:
c
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
2
aaaa ddd c
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
3
Lines are greater than average:
aaaa
Line lenght:4
ddd
Line lenght:3
Lines less than average:
c
Line lenght:1
```



```
Enter word:
aa
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
2
aaaa ddd c aa
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
6
Enter file name:
text.txt
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
```

```
Input command id:
2
aaaa ddd c
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
7
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
2
aaaa c ddd
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
8
```

```
Enter data:
c
Element index in mass: 1
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
9
Enter file name:
text.txt
Not equal
1.Enter word
2.Show data
3.Start
4.Start method other student
5.Save
6.Download
7.Sorting
8.Find
9.Compare with saving
0.End
Input command id:
0
End program
```

## Висновки

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