

# 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 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.

**Розробник:** Білий Вадим Іванович КІТ-119а №3.

## 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.next();
                collection.add(scan.next());
                break;
            case 2:
                System.out.println(collection.toString());
```

```

        break;
    case 3:
        Helper.start(collection.        ());
        break;
    case 4:
        VowelsOrConsonants.splitTextInSentences(new
(collection.        ()));
        break;
    case 5:
        System.out.        ("Enter file name: ");
        scan.        ();
        try {
            collection.        (scan.        ());
        } catch (IOException e) {
            e.        ();
        }
        break;
    case 6:
        System.out.        ("Enter file name: ");
        scan.        ();
        try {
            collection.        (scan.        ());
        } catch (ClassNotFoundException | IOException e) {
            // TODO Auto-generated catch block
            e.        ();
        }
        break;
    case 7:
        collection.        ();
        break;
    case 8:
        scan.        ();
        System.out.        ("Enter data:");
        int index = collection.        (scan.        ());
        if(index== -1)
            System.out.        ("Element don`t found");
        else
            System.out.        ("Element index in mass: "+index);
        break;
    case 9:
        System.out.        ("Enter file name: ");
        scan.        ();
        try {
            temp.        (scan.        ());
        } catch (ClassNotFoundException | IOException e) {
            // TODO Auto-generated catch block
            e.        ();
        }
        if(collection.        (temp))
            System.out.        ("Are equal");
        else
            System.out.        ("Not equal");
        break;
        default:
            System.out.        ("End program");
            prz = false;
        break;
    }
    scan.        ();
}

}

public void Save(String fileName) throws IOException {

```

```

        FileOutputStream outputStream= new                (fileName);
        ObjectOutputStream objectOutputStream = new
            (outputStream);
        objectOutputStream.                (this);
        objectOutputStream.                ();
    }
    public void Download(String fileName) throws IOException,
    ClassNotFoundException{
        FileInputStream inStream= new                (fileName);
        ObjectInputStream objectInStream = new                (inStream);
        MyCollection temp = (MyCollection) objectInStream.                ();
        this.                (temp.                ());
        objectInStream.                ();
    }
}

```

## Висновки

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