# Тема8. Основи введення/виведення Java SE

Мета: Оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

**1 ВИМОГИ**

**1.1 Розробник**

Інформація про розробника:

- Гряник Георгій Володимиович

- КІТ-119Д;

- 6 варіант.

**1.2 Загальне завдання**

1. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
2. Забороняється використання стандартного протокола серіалізації.
3. Продемонструвати використання моделі Long Term Persistence.
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

1.3 Задача

### **Поліцейська картотека**

Дані про злочинця: П.І.Б.; дата народження; дата судимостей ; дата останнього позбавлення волі; дата останнього звільнення.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООПРозробка класу Серіалізаціі/десеріалізаціі, клас контейнер, конструктори, типів даних створенi користувачем.  
2.2 Ієрархія та структура класівКлас “PoliceFile ” – описує поліцейську картотеку з можливістю додавати та виводити дані класу. Клас описує дані про злочинця відповідно до завданн. Клас «Date” – опису формат часу :день, місяць, рік . Створений для ергономічного запису дат відомостей про злочинця. Клас « Console\_program» - клас керування програми , створений щоб надавати користувачеві можливість керувати програмою. Клас «Сontainer» - клас-контейнер створений для зберігання даних у масиві розміром 255. Реалізовано додавання, видалення та інші можливості для керування даними. Клас «Serializator» - клас розроблений для збереження даних контейнеру у файл. При цьому зберігання прохоже у звичайний файл та файл типу .xml. При цьому в класі реалізовано методи для відновлення даних як із звичайного файлу так із .xml файлу.

2.3 Важливі фрагменти програми **/\*\***

**\* @author <Георгiй>**

**\***

**\*/**

import java.beans.XMLDecoder;

import java.beans.XMLEncoder;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.filechooser.FileNameExtensionFilter;

public class Serializator {

public boolean serialization(Сontainer container )

{

boolean flag=false;

JFileChooser fileopen = new JFileChooser();

fileopen.setCurrentDirectory(new File("C:/Users/дом/Documents/Java seve"));

fileopen.showDialog(null, "Зберегти файл");

File file = fileopen.getSelectedFile() ;///pathname

ObjectOutputStream oos=null;

try {

XMLEncoder encoder = new XMLEncoder(

new BufferedOutputStream(

new FileOutputStream( fileopen.getSelectedFile()+".xml")));

encoder.writeObject(container);

encoder.close();

FileOutputStream fos=new FileOutputStream(file);

if(fos!=null) {

oos= new ObjectOutputStream(fos);

oos.writeObject(container);

flag=true;

}

} catch(FileNotFoundException e) {e.printStackTrace();}

catch (IOException e) {e.printStackTrace(); }

finally {

if(oos!=null)

{try {oos.close();} catch (IOException e) {e.printStackTrace();}}

}

return flag;

}

static public Сontainer deserializtion() {

Сontainer container=null;

JFileChooser fileopen = new JFileChooser();

fileopen.setCurrentDirectory(new File("C:/Users/дом/Documents/Java seve"));

fileopen.showDialog(null, "Відкрити файл");

File file = fileopen.getSelectedFile() ; ///pathname

ObjectInputStream ois=null;

try {

FileInputStream fis=new FileInputStream(file);

if(fis!=null) {

ois= new ObjectInputStream(fis);

container=(Сontainer)ois.readObject();

}

} catch(FileNotFoundException e) {e.printStackTrace();}

catch (IOException e) {e.printStackTrace(); }

catch (ClassNotFoundException e) {e.printStackTrace(); }

finally {

//if(ois!=null)

{try {ois.close();} catch (IOException e) {e.printStackTrace();}}

}

return container;

}

public Сontainer deserializtionXML() {

Сontainer container = null;

JFileChooser fileopen = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"\*.xml", "xml");

fileopen.setCurrentDirectory(new File("C:/Users/дом/Documents/Java seve"));

fileopen.setFileSelectionMode(JFileChooser.FILES\_ONLY);

fileopen.setFileFilter(filter);

fileopen.showDialog(null, "Відкрити \*.xml файл");

try {

XMLDecoder decoder = new XMLDecoder(

new BufferedInputStream(

new FileInputStream(fileopen.getSelectedFile())));

var b = (Сontainer ) decoder.readObject();

return b;

} catch(FileNotFoundException e) {e.printStackTrace();}

finally { System.out.print("GUT");;

}

return container;

}

}

////////////////////////////////////////////////////////////////////

import java.util.Scanner;

public class Console\_program {

public static Scanner in = new Scanner(System.in);

public static Serializator serializator=new Serializator();

public static int dialog()

{

System.out.print("\n\n Оберіть команду:"

+"\n\*1 - Вивести поточнi записанi даннi"

+"\n\*2 - Додати данні"

+"\n\*3 - Оновити дані"

+"\n\*4 - сортувати"

+"\n\*5 - пошук"

+"\n\*6 - Кількість злочинців у базі "

+"\n\*7 - зберегти дані(save)"

+"\n\*8 - завантажити дані(load \*.xml)"

+"\n\*9 (clear)-очистити данi"

+"\n\*10 (exit)-вийти"

+"\n\n ваша команда: ");

return in.nextInt();

}

public static void Menu() ///функціє проводить координування по можливостям програм

{

Сontainer List=new Сontainer();

PoliceFile PL=new PoliceFile();

while(true)///нескінченний цикл який дозволяє працювати програмі

{

try {

int k=dialog();

switch(k)///пошук введеної команди

{

case 1: List.Print\_run(List, PL);;///////////

break;

case 2: List= List.dop\_INFO(List, PL);///////////

break;

case 3:List.UpdateData(List,PL);

break;

case 4: List.Sort(List);

break;

case 5 : List.posuk(List);

break;

case 6 :System.out.println("Кількість злочинців у базі :"+List.size());

break;

case 7 :

System.out.print("\n\nЗбереження даних:"+ serializator.serialization(List));

break;

case 8 : List=serializator.deserializtionXML();

break;

case 9 :List.clear();

break;

case 10 :return ;

}

}catch(Exception e) { for (int i=0;i<10;i++)System.out.println(" EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR EROR ");

System.out.print("\n\n\nТрапилась помилка. Але тепер все добре!!\n\n");

System.out.println(e);}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

Створена програма працює за принципом додавання редагування видалення. Тобто програма може додати дати ваші дані про злочинця (ім’я, прізвище, по-батькові) та дати через які він потрапив у цю базу. Також програма може вивести ці дані на екран, відсортувати за одним із трьох полів та якщо є необхідність то відредагувати якісь відомості про злочинця. Алгоритм додавання простий. Ви пише саме ті дані які просить від вас програма. Якщо ви десь допустили помилку то можна завжди очистити список або відредагувати дані якогось злочинця. Алгоритм редагування подібний до додаванням за винятком того, що замість створення нового об’єкту береться готовий та змінюється дані поля.

Дата реалізована окремим класом задля ергономічності програми. Тобто створено клас який містить ці дані і створюється поле типу дата та записується відповідні значення.

Для зручної роботи розроблено метод зберігання та відновлення даних не тільки у форматі файлу, а й у форматі xml файлу. При цьому можна злегкістю відновити збереженні дані.

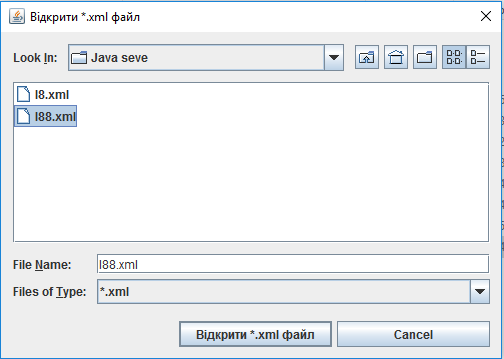


Рисунок 1 – Вікно із відкриттям xml файлу

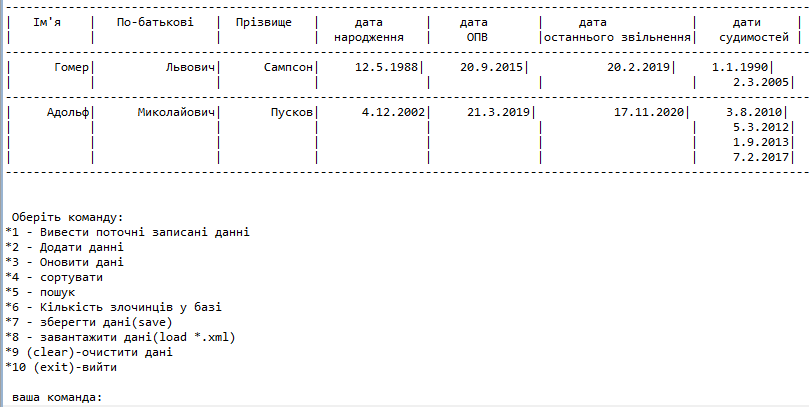


Рисунок 2. Вивод даних програми та меню



Рисунок 3 – фрагмент із .xml файлу

ВИСНОВКИ

Під час виконання лабораторної роботи було набуто навички роботи управління введенням/виведенням даних з використанням класів платформи Java SE.