# **Тема**: [Об'єктно-орієнтована декомпозиція](https://oop-khpi.gitlab.io/#tasks)

# Мета: Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.

**1 ВИМОГИ**

**1.1 Розробник**

Інформація про розробника:

- Резніченко Дмитро Володимирович

- КІТ-319;

- 6 варіант.

**1.2 Загальне завдання**

1. Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно [прикладної задачі](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/#_4) - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

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.

2.3 Важливі фрагменти програми/\*\*

\* @author <Дмитро>

\*

\*/

package Laba7;

import java.io.Serializable;

import java.util.Scanner;

public class PoliceFile implements Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

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

private String Name; ///Name

private String Surname; /// По-батькові

private String Lastname; ///прізвище

private Date DateOfBirth;/// dd.mm.yyyy дата народження;

public Date DatesOfConvictions[];/// dd.mm.yyyy дати судимостей

private int IndexDatesOfConvictions;/// dd.mm.yyyy index дати судимостей

private Date DateOfLastImprisonment;/// dd.mm.yyyy дата останнього позбавлення волі;

private Date DateOfLastreLease;/// dd.mm.yyyy дата останнього звільнення.

public PoliceFile(String Name,String Surname,String Lastname,Date DateOfBirth,Date DatesOfConvictions,Date DateOfLastImprisonment,Date DateOfLastreLease) {

this.Name=Name;

this.Surname=Surname;

this.Lastname=Lastname;

this.DateOfBirth=DateOfBirth;

this.DatesOfConvictions[0]=DatesOfConvictions;

this.IndexDatesOfConvictions=0;

this.DateOfLastImprisonment=DateOfLastImprisonment;

this.DateOfLastreLease=DateOfLastreLease;

}

public PoliceFile() {

this.Name="EROR";

this.Lastname="EROR";

this.Surname="EROR";

this.DateOfBirth=new Date();

this.DatesOfConvictions=new Date[255];

this.IndexDatesOfConvictions=-1;

this.DateOfLastImprisonment=new Date();

this.DateOfLastreLease=new Date();

}

public PoliceFile(PoliceFile h) {

}

public String getName() {return Name;}

public String getLastname() {return Lastname;}

public String getSurname() {return Surname;}

public Date getDateOfBirth() {return DateOfBirth;}

public Date getDatesOfConvictions(int a) {return DatesOfConvictions[a]; }

public Date getDateOfLastImprisonment() {return DateOfLastImprisonment;}

public Date getDateOfLastreLease() {return DateOfLastreLease; }

public int getIndexDatesOfConvictions() {return IndexDatesOfConvictions;}

public void setName(String name) {this.Name = name;}

public void setLastname(String lastname) {this.Lastname = lastname;}

public void setSurname(String surname) {this.Surname = surname;}

public void setDateOfBirth(Date dateOfBirth) {this.DateOfBirth = dateOfBirth; }

public void setDatesOfConvictions(Date datesOfConvictions,int a) {this.DatesOfConvictions[a] = datesOfConvictions;}

public void setDatesOfConvictions(Date[] datesOfConvictions ) {this.DatesOfConvictions = datesOfConvictions;}

public void setDateOfLastImprisonment(Date dateOfLastImprisonment) {this.DateOfLastImprisonment = dateOfLastImprisonment;}

public void setDateOfLastreLease(Date dateOfLastreLease) {this.DateOfLastreLease = dateOfLastreLease; }

public void setIndexDatesOfConvictions(int size) {this.IndexDatesOfConvictions=size-1;}

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

public void show() {

System.out.printf("|%11s|%17s|%13s|%7d.%d.%d|%7d.%d.%d|%13d.%d.%d|%6d.%d.%d|",this.Name,this.Surname,this.Lastname,

this.DateOfBirth.getDay(),this.DateOfBirth.getMoon(),this.DateOfBirth.getYear(),

this.DateOfLastImprisonment.getDay(),this.DateOfLastImprisonment.getMoon(),this.DateOfLastImprisonment.getYear(),

this.DateOfLastreLease.getDay(),this.DateOfLastreLease.getMoon(),this.DateOfLastreLease.getYear(),

this.DatesOfConvictions[0].getDay(),this.DatesOfConvictions[0].getMoon(),this.DatesOfConvictions[0].getYear());

for (int i=1;i<IndexDatesOfConvictions+1;i++)

{

System.out.printf("\n| | | | | | |%6d.%d.%d|" ,this.DatesOfConvictions[i].getDay(),this.DatesOfConvictions[i].getMoon(),this.DatesOfConvictions[i].getYear());

}

System.out.print( "\n----------------------------------------------------------------------------------------------------------------------\n");

}

void EL(PoliceFile add,boolean updata){

int i=0,n=2;

String value1=new String() , value2=new String(), value3=new String(), value4=new String();

int a=1,k=8;

if(updata==true) {

System.out.print( "1. Імя\n"+

"2. по-Батькові\n"+

"3. Прізвищеn\n"+

"4. Дата народження\n"+

"5. Дата судимостей\n"+

"6. Дата останнього позбавлення волі\n"+

"7. Дата останнього звільнення\n"+

"8. Все\n"+

"Ваш вибір\n") ;

k=in.nextInt();

if (k!=8)a=k; else a=1;

}

switch(a) {

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

case 1:

System.out.print( "Введiть ім'я злочинця : ");

value1=in.next();

System.out.println("додано" );

add.setName(value1);

if(k!=8) break;

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

case 2:

System.out.print("Введiть по-батькові : ");

value1=in.next();

add.setSurname(value1);

System.out.println( "додано");

if(k!=8) break;

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

case 3:

System.out.print("Введiть призвіще : ");

value1=in.next();

add.setLastname(value1);

System.out.println( "додано");

if(k!=8) break;

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

case 4:

i=0;

n=2;

value1=null;

Date date4=new Date();

System.out.print( "Введiть дату народження(dd.mm.yyyy) : ");

value1=in.next();

for(;i<n;i++)

value2+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=4;

else {i++ ; n=5;}

for(;i<n;i++)

value3+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=8;

else {i++ ; n=10;}

for(;i<n;i++)

value4+=value1.charAt(i);

date4.setDay(Integer.parseInt(value2));

date4.setMoon(Integer.parseInt(value3));

date4.setYear(Integer.parseInt(value4));

add.setDateOfBirth(date4);

System.out.println( " додано" );

if(k!=8) break;

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

case 5:

System.out.print( "Введiть кількість дат судимостей: ");

int size=in.nextInt();

System.out.print( "Введiть дати судимостей(dd.mm.yyyy): ");

for (int idex=0 ;idex <size; idex++)

{i=1;n=2;

System.out.print( "Дата "+(idex+1)+":");

value1=in.next();

value2=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value2+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=4;

else {i++ ; n=5;}

value3=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value3+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=8;

else {i++ ; n=10;}

value4=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value4+=value1.charAt(i);

date.setDay(Integer.parseInt(value2));

date.setMoon(Integer.parseInt(value3));

date.setYear(Integer.parseInt(value4));

add.setDatesOfConvictions((Date) date.clone(),idex);

System.out.println( " додано" );

setIndexDatesOfConvictions(size);

}

if(k!=8) break;/////////////////////////

case 6:

i=0;n=2;

Date date6=new Date();

System.out.print( "Введiть дату останнього позбавлення волі(dd.mm.yyyy): ");

value1=in.next();

value2=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value2+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=4;

else {i++ ; n=5;}

value3=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value3+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=8;

else {i++ ; n=10;}

value4=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value4+=value1.charAt(i);

date6.setDay(Integer.parseInt(value2));

date6.setMoon(Integer.parseInt(value3));

date6.setYear(Integer.parseInt(value4));

add.setDateOfLastImprisonment(date6);

System.out.println("додано" );

if(k!=8) break;

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

case 7:

i=0; n=2;

Date date7=new Date();

System.out.print( "Введiть дату останнього звільнення(dd.mm.yyyy): ");

value1=in.next();

value2=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value2+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=4;

else {i++ ; n=5;}

value3=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value3+=value1.charAt(i);

if(value1.charAt(i)>58&&value1.charAt(i)>48)n=8;

else {i++ ; n=10;}

value4=Character.toString(value1.charAt(i++));

for(;i<n;i++)

value4+=value1.charAt(i);

date7.setDay(Integer.parseInt(value2));

date7.setMoon(Integer.parseInt(value3));

date7.setYear(Integer.parseInt(value4));

add.setDateOfLastreLease(date7);

System.out.println("додано" );

if(k!=8) break;

}

}

}

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

/\*\*

\*

\*/

package Laba7;

import java.io.Serializable;

/\*\*

\* @author Дмитро

\*

\*/

public class Date implements Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private int Day;

private int Moon;

private int Year;

/\*\*

\*

\*/

public Date() {

this.Day=01;

this.Moon=01;

this.Year=1900;

}

public Date(int Day,int Moon,int Year ) {

if(Day<1)this.Day=1;

else this.Day=Day%31;

if(Moon<1)this.setMoon(1);

this.setMoon(Moon%13);

if(Year<1900)this.setYear(1900);

else this.setYear(Year);

}

public int getDay() {return Day; }

public int getMoon() {return Moon; }

public int getYear() {return Year; }

public void setDay (int day) { if(day<1)this.Day=1; else if(day>31) this.Day=31; else this.Day=day; }

public void setMoon(int moon) {if(moon<1)this.setMoon(1);else if(moon>12) this.Moon=12; else this.Moon=moon;}

public void setYear(int year) { if(year<1900)this.setYear(1900); else this.Year=year;}

}

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

package Laba7;

import java.io.Serializable;

import java.util.Arrays;

import java.util.Comparator;

import java.util.Scanner;

public class Сontainer implements Serializable {

private static final long serialVersionUID = 1L;

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

private int index=-1;

private PoliceFile[] m\_data=new PoliceFile[255];

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

public PoliceFile setInPoliceFile(int index)

{

return m\_data[index];

}

Сontainer() { ; }/// Конструктор

Сontainer(Сontainer rhs)/// Конструктор

{

for (int i = 0; i < rhs.index; i++) {

m\_data[i] = rhs.m\_data[i];

}

index = rhs.index;

}

void add(PoliceFile File) //додає вказаний елемент до кінця контейнеру;

{

if (index + 1 >= 255) return;

m\_data[++index] = (PoliceFile) File;

}

void add\_from\_keyboard(PoliceFile add\_from\_ )

{

PoliceFile add\_from\_keyboard=new PoliceFile();

add\_from\_keyboard.EL(add\_from\_keyboard,false);

add( add\_from\_keyboard);

System.out.print("\nВашi данi успiшно записанi\n");

}

void clear()/// видаляє всі елементи з контейнеру;

{

while (index!=-1)

iterator().remove();

}

void Print\_run(Сontainer A, PoliceFile pr) /// спадок вивести

{

int kol = A.size();

for (int i = 0; i < kol; i++)

{

pr = A.setInPoliceFile(i);

pr.show();

}

}

int size()/// повертає кількість елементів у контейнері;

{

return this.index+1;

}

void dop\_INFO(Сontainer A,PoliceFile h) /// Спосб додавання елементів

{

int a;

System.out.print("\n 1: З клавiатири\n 2: 3 Файлу(недоступно)\n Ваш вiбiр: ");

a=in.nextInt();

switch (a)

{

case 1:

A.add\_from\_keyboard(h);

break;

case 2:

//h.Riad\_FL(A), a = 0;

break;

}

}

void UpdateData(Сontainer A,PoliceFile h){

System.out.print("Введть індекс");

int index=in.nextInt();

if(index<0||index>this.index)

{

System.out.print("Такого злочиньця незнайдено");

return;

}

h= m\_data[index];

h.EL(m\_data[index],true);

m\_data[index]=h;

}

Object[] toArray() ///повертає масив, що містить всі елементи у контейнері;

{

return m\_data;

}

void posuk(Сontainer container)//// помилку пошуку

{

if (container.index==1) {

System.out.print("Масив пустий");

return ;

}

String text=new String();

System.out.print("Шукати:\n"+

" 1. Імя\n"+

" 2. по-Батькові\n"+

" 3. Прізвище\n"+

" 4. Index\n"+

"in: ");

int a=in.nextInt();

switch(a)

{

case 1:

System.out.print("Введіть Імя : ");

text=in.next();

for (int i=0;i<index+1 ;i++)

if (container.m\_data[i].getName().equals(text))

{

System.out.print("Позиція вашаго елемента: "+i+"\n");

container.m\_data[i].show();

return ;

}

System.out.print("Ваш елемент незнайденo");

break;

case 2:

System.out.print("Введіть по-Батькові : ");

text=in.next();

for (int i=0;i<index+1 ;i++)

if (container.m\_data[i].getSurname().equals(text))

{

System.out.print("Позиція вашаго елемента: "+i+"\n");

container.m\_data[i].show();

return ;

}

System.out.print("Ваш елемент незнайденo");

break;

case 3:

System.out.print("Введіть прізвище : ");

text=in.next();

for (int i=0;i<index+1 ;i++)

if (container.m\_data[i].getLastname().equals(text))

{

System.out.print("Позиція вашаго елемента: "+i+"\n");

container.m\_data[i].show();

return ;

}

System.out.print("Ваш елемент незнайденo");

break;

case 4:

System.out.print("Введіть Ім'я : ");

a=in.nextInt();

if (a>-1||a<this.index+1)

{

System.out.print("Позиція вашаго елемента: "+a+"\n");

container.m\_data[a].show();

return ;

}

System.out.print("Ваш елемент незнайденo");

break;

}

}

void Sort(Сontainer container)//// помилку пошуку

{

if (container.index==1) {

System.out.print("Масив пустий");

return ;

}

String text=new String();

System.out.print("Сортувати:\n"+

" 1. Імя\n"+

" 2. по-Батькові\n"+

" 3. Прізвище\n"+

"in: ");

int a=in.nextInt();

boolean pr; /// для проверки отсортирован ли массив

do

{

pr = false;

switch(a)

{

case 1:

for (int i = 0; i < index; i++) /// проходим слева направо

if (comparison(m\_data[i].getName() , m\_data[i + 1].getName())==1) { swap(i); pr = true; }

break;

case 2:

for (int i = 0; i < index; i++) /// проходим слева направо

if (comparison(m\_data[i].getSurname() , m\_data[i + 1].getSurname())==1) { swap(i); pr = true; }

break;

case 3:

for (int i = 0; i < index; i++) /// проходим слева направо

if (comparison(m\_data[i].getLastname() , m\_data[i + 1].getLastname())==1) { swap(i); pr = true; }

break;

}

} while (pr);

System.out.print(" \*сортування завершено ");

}

void swap(int i) /// Swap елемент

{

PoliceFile temp;

temp = m\_data[i];

m\_data[i] = m\_data[i + 1];

m\_data[i + 1] = temp;

}

//@SuppressWarnings("unused")

public static int comparison(String a,String b)

{

int len=0;

if(a.length()<b.length())len=a.length();

else len=b.length();

for (int i=0;i<len;i++)

{

if (a.charAt(i)>b.charAt(i)) return 1;

if (a.charAt(i)<b.charAt(i)) return -1;

}

if(a.length()<b.length())return -1;

else if (a.length()>b.length()) return 1;

return 0;

}

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

public Iterator<String> iterator() ///повертає ітератор відповідно до Interface Iterable.

{

return new m\_Iterator();

}

public class m\_Iterator implements Iterator<String>

{

int ix = 0;

public boolean hasNext() {

if(ix<this.ix+1)

return true;

return false;

}

public PoliceFile next()

{

return m\_data [ix++];

}

public void remove()

{

for (int i=ix; i < index-1; i++)

m\_data[i]=m\_data[i+1];

m\_data[index--] = null;

//throw new UnsupportedOperationException("remove");

}

public PoliceFile begin()

{

return m\_data[0];

}

public PoliceFile end()

{

return m\_data[index];

}

}

}

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

package Laba7;

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*.println("\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 - (недоступно)"

+"\n\*7 - зберегти дані(save)"

+"\n\*8 - завантажити дані(load)"

+"\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.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 ://s.d\_comparison(s);

break;

case 7 :

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

break;

case 8 : List=*serializator*.deserializtion();

break;

case 9 :List.clear();

break;

case 10 :return ;

}

}catch(Exception e) {

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

}

}

}

**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.

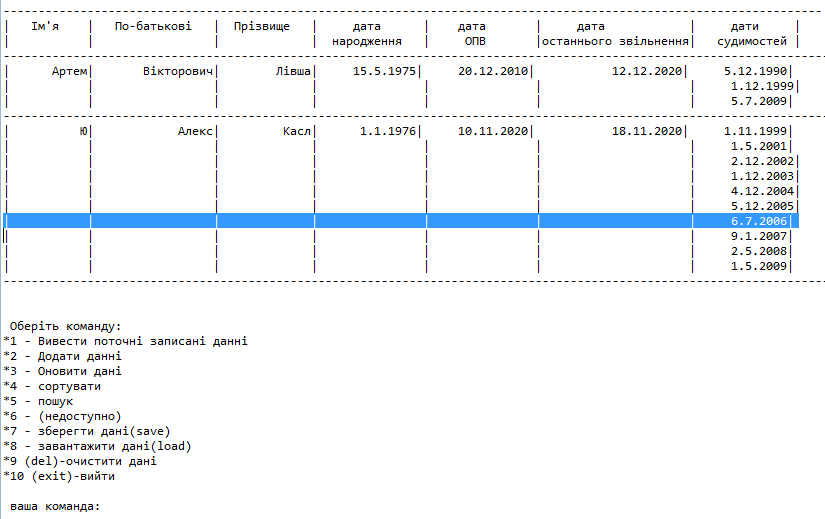


Рисунок 1 – результат додавання та відображення даних

ВИСНОВКИ

Під час виконання лабораторної роботи було набуто навички роботи з роботою об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.