**Мета:** вивчити особливості роботи з власними класами у Java.

**Обладнання:** ПК, програмне забезпечення Intellij IDEA, методичні вказівки та завдання до лабораторної роботи.

**Завдання.**

1.**Розробити консольний застосунок для роботи з класом згідно свого варіанту. Створити не менше 10 записів для виконання операцій, вказаних у завданні. Для ідентифікації спроби введення з клавіатури некоректних даних описати виключення.**

Описати клас для бази зданих з інформацією про книги у формі: автор, назва книги, видавництво, рік видання. Вивести дані про книги з програмування (перевіряти, чи є частиною назви книги слово «програмування» з малої або великої літери) у порядку спадання років видань.

2.**Розробити консольний застосунок для роботи з базою даних, що зберігається у текстовому файлі (початковий масив не менше 10 записів). Структура бази даних описується класом згідно вашого варіанта. Передбачити роботу з довільною кількістю записів. Для ідентифікації спроби введення з клавіатури некоректних даних описати виключення. Всі поля класу зробити закритими, а доступ до них реалізувати через get i set. Реалізувати конструктори з параметрами та без параметрів, а ініціалізацію полів виконати через властивості. Реалізувати методи для:**

**– додавання записів; редагування записів; знищення записів; виведення інформації з файла на екран; пошук потрібної інформації за конкретною ознакою (поле Параметр пошуку); сортування за різними полями (поле Параметр сортування).**

**Меню програми реалізувати по натисненню на певні клавіші: наприклад, Enter – вихід, п - пошук, р – редагування тощо.**

Клас : Колекція фільмів  
Поля : Назва, прізвище режисера, рік випуску, кіностудія, тривалість фільму  
Параметр сортування : Назва  
Параметр пошуку : Рік випуску

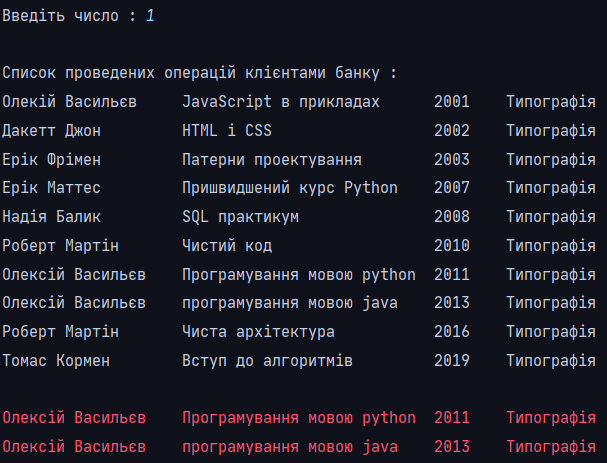
[**Код на GitHub**](https://github.com/Kolya532/Java/tree/main/Лабараторна%20робота%203(Класи))

**Результат виконання :**

1. **Код :**

*public static void* TaskOne() {  
 Scanner scanner = *new* Scanner(System.*in*);  
 ArrayList<Book> list = *new* ArrayList<Book>();  
 list.add(*new* Book("Олексій Васильєв ","Програмування мовою python",2011,"Типографія"));  
 list.add(*new* Book("Роберт Мартін ","Чиста архітектура ",2016,"Типографія"));  
 list.add(*new* Book("Олексій Васильєв ","програмування мовою java ",2013,"Типографія"));  
 list.add(*new* Book("Ерік Фрімен ","Патерни проектування ",2003,"Типографія"));  
 list.add(*new* Book("Томас Кормен ","Вступ до алгоритмів ",2019,"Типографія"));  
 list.add(*new* Book("Надія Балик ","SQL практикум ",2008,"Типографія"));  
 list.add(*new* Book("Ерік Маттес ","Пришвидшений курс Python ",2007,"Типографія"));  
 list.add(*new* Book("Роберт Мартін ","Чистий код ",2010,"Типографія"));  
 list.add(*new* Book("Дакетт Джон ","HTML і CSS ",2002,"Типографія"));  
 list.add(*new* Book("Олекій Васильєв ","JavaScript в прикладах ",2001,"Типографія"));  
 list.sort(*Comparator*.*comparing*(Book::getYear));  
 System.*out*.println("\nСписок проведених операцій клієнтами банку :");  
 *for* (Book book : list) {  
 System.*out*.println(book.toString());  
 }  
 System.*out*.println();  
 *for* (Book book : list) {  
 *if*(book.getNameBook().matches(".\*[Пп]рограмування.\*"))  
 System.*err*.println(book.toString());  
 }  
 *try* {  
 Thread.*sleep*(100);  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
}

**Результат :**

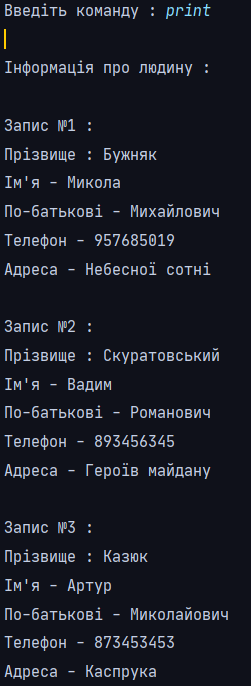


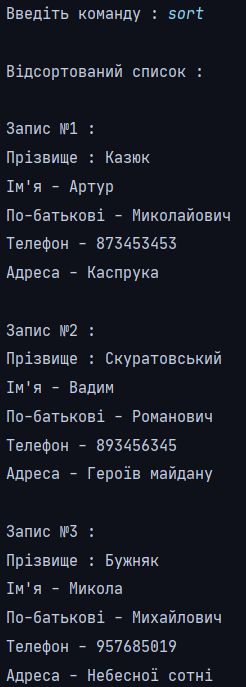
1. **Код :**

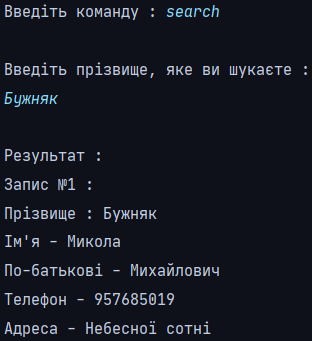
*public static void* TaskTwo(){  
 *try* {  
 Scanner scanner = *new* Scanner(System.*in*);  
 Task task = *new* Task();  
 *for*(;;){  
 System.*out*.println("\nadd - додати " + "\n" +  
 "edit - редагувати " + "\n" +  
 "print - висести список" + "\n" +  
 "search - пошук " + "\n" +  
 "sort - сортування " + "\n" +  
 "delete - видалити " + "\n" +  
 "exit - вихід");  
 System.*out*.print("\nВведіть команду : ");  
 String command = scanner.nextLine();  
  
 *switch* (command){  
 *case* "add":  
 task.add();  
 *break*;  
 *case* "edit":  
 task.edit();  
 *break*;  
 *case* "print":  
 task.print();  
 *break*;  
 *case* "search":  
 task.search();  
 *break*;  
 *case* "sort":  
 task.sort();  
 *break*;  
 *case* "delete":  
 task.delete();  
 *break*;  
 *case* "exit":  
 *return*;  
 }  
 }  
 }  
 *catch* (IOException ex){  
 System.*err*.println(ex.getMessage());  
 }  
}

*public class* Task {  
 ArrayList<TelephoneDirectory> telephoneDirectory;  
  
 *public* Task() *throws* IOException {  
 telephoneDirectory = *read\_from\_file*("Movies.txt");  
 }  
  
 *public void* add() *throws* IOException{  
 TelephoneDirectory new\_telephoneDirectory = *new* TelephoneDirectory();  
 System.*out*.println("\nВведіть інформацію :");  
 new\_telephoneDirectory.input();  
 telephoneDirectory.add(new\_telephoneDirectory);  
  
 System.*out*.println("\nЗапис доданий!");  
  
 *this*.write\_to\_file("Movies.txt");  
 }  
  
 *public void* edit() *throws* IOException {  
 count\_note();  
  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("\nВведіть номер, який ви хочете редагувати : ");  
 String line = scanner.nextLine();  
 *if*(!line.matches("\\d+") || Integer.*parseInt*(line) <= 0 || Integer.*parseInt*(line) > telephoneDirectory.size()){  
 System.*err*.println("Не вірно введений номер!");  
 *this*.edit();  
 *return*;  
 }  
 *int* nomer = Integer.*parseInt*(line);  
 nomer--;  
  
 System.*out*.println("\nВідредагуйте :");  
 telephoneDirectory.get(nomer).input();  
 System.*out*.println("\nРедагування пройшло успішно!");  
  
 *this*.write\_to\_file("Movies.txt");  
 }  
  
 *public void* print() *throws* IOException {  
 count\_note();  
  
 *int* i = 0;  
 System.*out*.println("\nІнформація про людину :");  
 *for* (*var* p:  
 telephoneDirectory) {  
 System.*out*.println("\nЗапис №" + (i + 1) + " :");  
 p.print();  
 i++;  
 }  
 }  
  
 *public void* search() *throws* IOException {  
 count\_note();  
  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.println("\nВведіть прізвище, яке ви шукаєте : ");  
 String surname = scanner.nextLine();  
  
 *boolean* flag = *false*;  
 *int* i = 0;  
 *for* (*var* p:  
 telephoneDirectory) {  
 *if*(p.getSurname().equals(surname)) {  
 *if*(flag == *false*){  
 System.*out*.println("\nРезультат : ");  
 flag = *true*;  
 }  
 System.*out*.println("Запис №" + (i + 1) + " :");  
 p.print();  
 System.*out*.println();  
 i++;  
 }  
 }  
  
 *if*(flag == *false*){  
 System.*out*.println("Пошук не вдалий!");  
 }  
 }  
  
 *public void* sort() *throws* IOException {  
 count\_note();  
  
 ArrayList<TelephoneDirectory> sortTelephone = *new* ArrayList<TelephoneDirectory>(telephoneDirectory);  
 sortTelephone.sort(*Comparator*.*comparing*(TelephoneDirectory::getPhone));  
  
 *int* i = 0;  
 System.*out*.println("\nВідсортований список :");  
 *for* (*var* p: sortTelephone) {  
 System.*out*.println("\nЗапис №" + (i + 1) + " :");  
 p.print();  
 i++;  
 }  
 }  
  
 *public void* delete() *throws* IOException {  
 count\_note();  
  
 Scanner scanner = *new* Scanner(System.*in*);  
 System.*out*.print("\nВведіть номер, який ви хочете видалити : ");  
 String line = scanner.nextLine();  
 *if*(!line.matches("\\d+") || Integer.*parseInt*(line) <= 0 || Integer.*parseInt*(line) > telephoneDirectory.size()){  
 System.*err*.println("Не вірно введений номер!");  
 *this*.delete();  
 *return*;  
 }  
 *int* surname = Integer.*parseInt*(line);  
 surname--;  
  
 telephoneDirectory.remove(surname);  
 System.*out*.println("\nВидалення пройшло успішно!");  
  
 *this*.write\_to\_file("Movies.txt");  
 }  
  
 *public void* count\_note() *throws* IOException {  
 *if*(telephoneDirectory.size() == 0)  
 *throw new* IOException("Записів нема!");  
 }  
  
 *private static void* isFile(String filepath) *throws* IOException{  
 *//перевірка чи файл існує* File file = *new* File(filepath);  
 *if*(!file.isFile()) *throw new* IOException("Файл не знайдено!");  
 *//file.createNewFile();* }  
  
  
 *public void* write\_to\_file(String filepath) *throws* IOException{  
 *isFile*(filepath);  
  
 BufferedWriter output = *new* BufferedWriter(*new* FileWriter(filepath));  
 *for* (*var* p:  
 telephoneDirectory) {  
 output.write(p.to\_string());  
 }  
 output.flush();  
 output.close();  
 output.close();  
 }  
  
  
 *private static* ArrayList<TelephoneDirectory> read\_from\_file(String filepath) *throws* IOException {  
 *isFile*(filepath);  
  
 BufferedReader reader = *new* BufferedReader(*new* FileReader(filepath));  
 String text;  
 ArrayList<TelephoneDirectory> arrayList = *new* ArrayList<>();  
 TelephoneDirectory movies;  
 *while* ((text = reader.readLine()) != *null*) {  
 arrayList.add(*new* TelephoneDirectory(text, reader.readLine(), Integer.*parseInt*(reader.readLine()), reader.readLine(), reader.readLine()));  
 }  
 reader.close();  
 *return* arrayList;  
 }  
}

**Результат :**







**Висновок:** я вивчив особливості роботи з власними класами у Java.