

## Лабораторна робота № 9

### Створення структурованих типів даних на основі класів.

**Мета роботи:** оволодіння навичками опису класів та використання об'єктів для зберігання та опрацювання структурованих даних.

**Завдання .** Описати клас для зберігання даних про заданий тип об'єктів у заданій предметній області. При описі класу дотримуватися принципу інкапсуляції – усі поля з даними повинні бути закритими, а для роботи з даними, що містяться у цих полях, реалізувати відповідні методи. Для створення екземплярів класу реалізувати відкриті конструктори, що заповнюють поля об'єктів даними. Реалізувати перегрузку одного – двох методів класу та конструктора класу. Для демонстрації функціонування класу створити масив з 10-15 примірників класу. Для створеного масиву реалізувати:

- а) вивід на консоль даних з усіх елементів масиву;
- б) вивід даних лише тих елементів, які відповідають заданому у варіанті завдання критерію;
- в) пошук в масиві та вивід на консоль об'єктів з вказаною властивістю (згідно до варіанта).

#### **Варіанти завдань**

1. Предметна область: дослідна лабораторія, клас: обладнання для дослідів, орієнтовний перелік полів: назва обладнання, кому видано, кількість, видано-повернуто, дата видачі. Вивести окремо список наявного та окремо список виданого обладнання. Реалізувати пошук обладнання за назвою.
2. Предметна область: оренда житла, клас: помешкання, орієнтовний перелік полів: адреса, кількість кімнат, вартість оренди, помешкання орендовано (так/ні), дата оренди, термін оренди. Вивести окремо список вільних та окремо список орендованих помешкань. Реалізувати пошук вільних помешкань з вказаною кількістю кімнат та допустимою вартістю оренди.
3. Предметна область: автостоянка, клас: автомобіль, орієнтовний перелік полів: марка автомобіля, прізвище та адреса власника, номерний знак, номер місця на стоянці, наявність на стоянці, дата та час заїзду чи виїзду зі стоянки. Вивести окремо список автомобілів, присутніх на стоянці, та автомобілів, що виїхали зі стоянки. Реалізувати пошук автомобіля за вказаним номерним знаком.
4. Предметна область: фільмотека, клас: фільм, орієнтовний перелік полів: назва фільму, рік випуску, перелік акторів у головних ролях, формат файлу, розмір в мб, можливість безкоштовного завантаження. Вивести окремо список фільмів у різних форматах. Реалізувати пошук фільмів за прізвищем актора.
5. Предметна область: інтернет-магазин, клас: товар, орієнтовний перелік полів: назва товару, категорія, опис товару, ціна за одиницю, наявність на складі, кількість, дата поставки. Вивести окремо список відсутніх товарів. Реалізувати пошук товарів за вказаною категорією.
6. Предметна область: прохідна підприємства, клас: працівник, орієнтовний перелік полів: прізвище, посада, відділ, номер посвідчення, присутній на території чи вийшов за межі, час останнього входу чи виходу. Вивести окремо список працівників, присутніх на території, та відсутніх. Реалізувати пошук даних працівника за вказаним прізвищем.
7. Предметна область: оренда автомобілів, клас: автомобіль, орієнтовний перелік полів: марка автомобіля, номерний знак, рік випуску, автомобіль орендовано (так/ні), дата оренди, термін оренди. Вивести окремо список вільних та окремо список орендованих автомобілів. Реалізувати пошук автомобілів, що звільняться з оренди у вказаному місяці.

8. Предметна область: банківські послуги, клас: картковий рахунок, орієнтовний перелік полів: номер картки, прізвище клієнта, адреса клієнта, сума на рахунку, кредит/депозит, дата видачі, термін дії. Вивести окремо список кредитних та окремо список депозитних карток. Реалізувати пошук даних за номером картки.
9. Предметна область: служба доставки, клас: замовлення, орієнтовний перелік полів: назва товару, адреса для доставки, вартість, кількість, проведення попередньої оплати, дата та час замовлення. Вивести окремо список оплачених та неоплачених замовлень. Реалізувати пошук замовлення за вказаною адресою, або її частиною.
10. Предметна область: відділ кадрів, клас: працівник, орієнтовний перелік полів: прізвище, домашня адреса, посада, рік народження, кількість дітей, дата прийому на роботу, постійний працівник/ працівник за сумісництвом. Вивести окремо список постійних працівників та окремо список сумісників. Реалізувати пошук даних працівника за вказаною посадою та допустимим стажем роботи.
11. Предметна область: готель, клас: готельний номер, орієнтовний перелік полів: опис номера, кількість місць, вартість проживання, вільний/зайнятий, дата заселення, термін проживання. Вивести окремо список вільних та окремо список зайнятих номерів. Реалізувати пошук вільних помешкань з вказаною кількістю місць та допустимою вартістю проживання.
12. Предметна область: інтернет-магазин, клас: замовлення, орієнтовний перелік полів: назва товару, адреса для доставки, ціна за одиницю, кількість, проведення попередньої оплати, дата замовлення. Вивести окремо список оплачених та неоплачених замовлень. Реалізувати пошук замовлення за вказаною адресою, або її частиною.
13. Предметна область: автотранспортне підприємство, клас: автомобіль, орієнтовний перелік полів: марка автомобіля та номерний знак, прізвище водія, вантажопідйомність, автомобіль в рейсі (так/ні), дата виїзду в рейс, тривалість рейсу в днях. Вивести окремо список автомобілів, вільних для перевезень, та автомобілів, що виїхали у рейс. Реалізувати пошук вільних автомобілів з достатньою для вказаного вантажу вантажопідйомністю .
14. Предметна область: банківські послуги, клас: операція з картковим рахунком, орієнтовний перелік полів: номер картки, номер терміналу, адреса терміналу, час, сума операції, прихід/розхід. Вивести окремо список прихідних та окремо список розхідних операцій. Реалізувати пошук даних про операції за вказаним номером терміналу.
15. Предметна область: прийом на роботу, клас: потенційний працівник, орієнтовний перелік полів: прізвище, домашня адреса, спеціальність, рік народження, рейтинг за співбесідою, запрошення надіслано (так/ні), дата запрошення. Вивести окремо список претендентів, яким надіслано запрошення. Реалізувати пошук даних претендентів за вказаною спеціальністю та рівнем рейтингу.
16. Предметна область: бібліотека, клас: книга, орієнтовний перелік полів: назва, автор, рік видання, кількість сторінок, наявна у фонді/видана читачеві дата видачі. Вивести окремо список наявних та окремо список виданих читачам книжок. Реалізувати пошук книжок за прізвищем автора.

**Приклад.**

Варіант 16. Предметна область: бібліотека, клас: книга, орієнтовний перелік полів: назва, автор, рік видання, кількість сторінок, наявна у фонді/видана читачеві дата видачі. Вивести окремо список наявних та окремо список виданих читачам книжок. Реалізувати пошук книжок за прізвищем автора.

Створюємо новий проект. В нього додаємо клас Book (без головного методу main), в якому описуємо структуру та методи вказаного об'єкта (див. коментарі до коду):

```

package lab9;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Book {

    /*
    Нижче описано поля класу, що використовуватимуться для зберігання
    даних кожного окремого екземпляра
    Дотримуючись інкапсуляції всі поля робимо закритими (private)
    їх читання чи зміни реалізуємо спеціальні методи get-ери і set-ери
    В зміні автора, назви, року і к-сті сторінок немає сенсу, оскільки
    в такому випадку слід створювати новий екземпляр класу "книга".
    Поля у яких немає методу для зміни значення прийнято помічати
    як final
    */

    private final String title;// назва книги
    private final String author;// автор або декілька авторів
    private final int year;// рік видання
    private final int countOfPages;//кількість сторінок
    private boolean available;//наявність книги в бібліотеці
    //книга може бути видана користувачеві - false
    //або зберігатися у фонді - true
    private Date issueDate;//дата видачі, має сенс лише при available == false
    //якщо книга є у фонді issueDate = Null

    // методи-геттери для усіх полів

    public String getTitle(){
        return title;
    }
    public String getAuthor(){
        return author;
    }
    public int getYear(){
        return year;
    }
    public int getCountOfPages(){
        return countOfPages;
    }
    public boolean isAvailable(){
        return available;
    }
    public String getIssueDate() throws ParseException{
        if(issueDate!=null){
            SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yy");
            return dateFormat.format(issueDate);
        }
        else return "";
    }
    // методи-сеттери змінюють лише наявність книги у фонді
    // та дату видачі користувачеві, причому зміна повинна
    // обидва поля одночасно
    public void setIssue() {
        //видача книги зареєстрована поточною датою
        if(available){

```

```

        available = false;
        issueDate = new Date();
    }
}
public final void setIssue(String date) throws ParseException{
    //видача книги за вказаною датою
    if(available){
        available = false;
        issueDate = new SimpleDateFormat("dd.MM.yy").parse(date);
    }
}
public void setReturn() {
    if(!available) available = false;
    issueDate = null;
}

//Конструктори з використанням перегрузки
public Book(String author, String title, int year, int n){
    //основний конструктор
    this.author = author;
    this.title = title;
    this.year = year;
    countOfPages = n;
    available = true;
    issueDate = null;
}
public Book(String author, String title, int year, int n, String date) throws
ParseException{
    //конструктор із вказаною датою видачі книги користувачеві
    //суперчить логіці предметної області, але дозволяє спростити
    //тестування можливостей класу (див. головний клас проекту)
    this(author, title, year, n);
    this.setIssue(date);
    this.available = false;
}

public Book(String author, String title, int n){
    //конструктор для книг, що вийшли з друку в поточному році
    //перегрузка спрощує реєстрацію книжки
    this.author = author;
    this.title = title;
    countOfPages = n;
    available = true;
    issueDate = null;
    Date now = new Date();
    SimpleDateFormat nowFormat = new SimpleDateFormat("yyyy");
    year = Integer.parseInt(nowFormat.format(now));
}
@Override
public String toString(){
    //первизначений метод toString дозволяє зручно виводити інформацію,
    //яку зберігає екземпляр класу, на консоль
    return author + " " + title + " - " + year + ", " + countOfPages + " с.";
}
}
}

```

Для демонстрації можливостей створеного класу, та виконання додаткових завдань з варіанта слід створити принаймні ще один клас з методом main. В поданому нижче кодї використано два класи: клас Library об'єднує масив книг (екземплярів класу Book)

методами для обробки цього масиву: виводом елементів та пошуком за заданим критерієм.

```
package lab9;

public class Library {

    private Book[] books;

    public Library(Book[] list){
        //відкритий конструктор приймає посилання на масив з даними
        books = list;
    }
    /* використані нижче цикли з ітератором
    * for (Book book : books)
    * можна записувати і в традиційній формі
    * for (int i = 0; i<books.length; i++)
    * але подана нижче форма надає коду більшу гнучкість,
    * оскільки придатна для роботи з колекціями
    * такими, як, наприклад, множини і поданий код допускає
    * в перспективі заміну масиву books на множину books
    * без жодних затрат на його модифікацію
    */

    public void printList(){
        //друк всіх елементів масиву
        for (Book book : books) {
            System.out.println(book);
        }
    }

    public void printList(boolean isAvailable){
        //друк наявних/виданих книг
        for (Book book : books) {
            if (book.isAvailable() == isAvailable)System.out.println(book);
        }
    }

    public String findBookByAuthor (String author){
        //пошук елементів, що містять у полі author заданий фрагмент

        String result="";
        for (Book book : books) {
            if (book.getAuthor().contains( author))
                result += book.toString()+ "\n"+
                    (book.isAvailable()?" в наявності":" видана") + "\n";
        }
        return result.isEmpty()?"нічого не знайдено":result;
    }
}
```

В методі main головного класу створено і проініціалізовано статичними даними масив книг, на основі якого створюється примірник класу Library. Далі продемонстровано роботу усіх реалізованих в класі Library методів.

```
package lab9;

import java.text.ParseException;
import java.util.Scanner;

public class Lab9 {
    public static void main(String[] args) throws ParseException {

        Library myLibrary;
        Book [] books = {new Book("Ткаченко О.М.",
            "Комп'ютерне програмування на мові Java.", 2013, 147),
            new Book("Копитко М.Ф., Іванків К.С. ",
            "Основи програмування мовою Java: Тексти лекцій.", 2002, 83, "25.12.14"),
            new Book("Р. А. Мельник, М. М. Сенів",
            "Програмування на JAVA: метод. " +
            "Вказівки до лаборатор. робіт з дисципліни Об'єкт.-оріент. програмув.",
            2007,42),
            new Book("Брюс Эккель", "Философия Java", 1168),
            new Book("Горбань А.Г.", "Програмування в Java", 2008,
            310, "03.03.15"),
            new Book("И.Н.Блинов, В.С.Романчик.",
            "Java. Методы программирования", 2013, 896),
            new Book("И.Н.Блинов, В.С.Романчик.",
            "Java. Промышленное программирование", 2007, 704,"01.02.15"),
            new Book("Герберт Шилдт",
            "Java. Полное руководство", 2012, 1104),
            new Book("Брнакевич І.Є., Вагін П.П.",
            "Програмування мовою Java: використання " +
            "фундаментальних класів: Тексти лекцій.",2002,
            75,"14.12.14"),
            new Book("Ментинський С.М.",
            "Використання класів для опрацювання структур даних. " +
            "Завдання до лабораторних робіт № 9-11.",36)};

        myLibrary = new Library(books);

        System.out.println("Список книг у бібліотеці:");
        myLibrary.printList();

        System.out.println("Список книг виданих у користування:");
        myLibrary.printList(false);

        System.out.println("Список книг що залишилися у книгосховищі:");
        myLibrary.printList(true);

        Scanner input = new Scanner(System.in);
        System.out.println("Пошук книг. Введіть прізвище автора");
        String author = input.next();
        input.close();

        String authorBooks = myLibrary.findBookByAuthor(author);
        System.out.println(authorBooks);

    }
}
```