

## Лабораторна робота № 13

### Створення графічного інтерфейсу користувача для роботи зі структурованими типами даних засобами пакету Swing.

**Мета роботи:** оволодіння навичками роботи з візуальними компонентами та контейнерами, знайомство з менеджерами компоновки візуальних контейнерів, вдосконалення навичок моделювання обробки подій компонентів в Swing.

**Завдання .** Спроекувати та запрограмувати візуальний віконний інтерфейс користувача для роботи зі списком об'єктів, отриманим у результаті виконання лабораторних робіт 9-11. Вікно графічного інтерфейсу обов'язково повинно мати наступні елементи:

- а) поля для вводу (JTextField) властивостей описаних об'єктів;
- б) візуальний список (JList), що відображатиме дані раніше доданих примірників створеного класу;
- в) кнопки (JButton) для виконання маніпуляцій зі списком та його елементами, зокрема, додавання нового примірника класу, запису списку у файл, решта – на розгляд розробника.

**Варіанти завдань** відповідають варіантам завдань лабораторної роботи 9.

#### **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 копію проекту, в якому виконувалися лабораторні роботи 9-11 (після зміни назви пакету рекомендовано перезаписати новий файл з даними). Створений раніше клас Book, в якому описано структуру та методи об'єкта-книги можна залишити без змін. Клас Library, що містить методи для роботи зі списком книг для використання разом з компонентом JList потрібно підкорегувати. Компонент JList зберігає дані у вигляді списку об'єктів в окремій колекції, що називається модель і може бути представлена екземпляром класу DefaultListModel. Набір методів цієї колекції дещо відрізняється від методів списку ArrayList (зокрема відсутній ітератор). Тому замінивши тип списку books на DefaultListModel<Book> слід також замінити у методах цикли виду

```
for (Book book : books) {  
    System.out.println(book);  
}
```

на аналогічні цикли

```
for(int i=0; i<books.size();i++) {  
    System.out.println(books.get(i));  
}
```

В класі Library також потрібно додати відкритий гетер для самого списку:

```
public DefaultListModel<Book> getBooks(){  
    return books;  
}
```

щоб можна було використати посилання на нього як посилання на модель з даним для компонента JList в головному класі

Головний клас скопійованого проекту краще замінити новим. Він має наслідувати клас JFrame з пакету javax.swing, в його конструкторі слід описати, розмістити та додати до форми усі необхідні візуальні компоненти. Обробники подій кнопок (де це доцільно) можна реалізувати окремими вкладеними класами цього ж класу. Орієнтовний код головного класу-фрейма.

```
package lab14;

import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.ParseException;
import javax.swing.*;

public class LibraryWindow extends JFrame{

    private JTextField authorField, titleField, yearField, pagesField, dateField ;
    private JCheckBox issueCheck;
    private Library myLibrary;

    public LibraryWindow(){
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(700, 400));//задаємо розміри вікна

        JPanel addBook = new JPanel();//панель для елементів додавання книги
        addBook.setLayout(new GridLayout(6,2));
        addBook.setBounds(0, 0, 300, 350);
        JLabel authorLbl = new JLabel("Автор");
        addBook.add(authorLbl);
        authorField = new JTextField();
        addBook.add(authorField);
        JLabel titleLbl = new JLabel("Назва");
        addBook.add(titleLbl);
        titleField = new JTextField();
        titleField.setPreferredSize(new Dimension(150, 20));
//задання PreferredSize одному з компонентів змушує
//табличний LayoutManager виділяти більше місця усім
        addBook.add(titleField);
        JLabel yearLbl = new JLabel("Рік");
        addBook.add(yearLbl);
        yearField = new JTextField();
        addBook.add(yearField);
        JLabel pagesLbl = new JLabel("К-сть сторінок");
        addBook.add(pagesLbl);
        pagesField = new JTextField();
        addBook.add(pagesField);
        JLabel issueLbl = new JLabel("Видана");
        addBook.add(issueLbl);
        issueCheck = new JCheckBox();
        addBook.add(issueCheck);
        JLabel dateLbl = new JLabel("Дата");
        addBook.add(dateLbl);
        dateField = new JTextField();
        addBook.add(dateField);

        JPanel components = new JPanel();//панель для розміщення елементів форми
```

```

components.setLayout(new java.awt.BorderLayout());
components.add(addBook, java.awt.BorderLayout.WEST);

myLibrary = new Library();
myLibrary.loadFromFile("C:\\BooksArray.dat");
JList<Book> list = new JList<Book>(myLibrary.getBooks());

//список JList зберігає дані в моделі окремо
//в нашому прикладі модель є полем класу Library
//список розміщуємо в спеціальний контейнер зі смугами прокрутки
JScrollPane listBox = new JScrollPane(list);
listBox.setSize(400, 350);
components.add(listBox, java.awt.BorderLayout.CENTER);
setContentPane(components);

JMenuBar commands = new JMenuBar();
JButton btOk = new JButton("Додати книгу");
commands.add(btOk);
AddBookListener abl = new AddBookListener();
btOk.addActionListener(abl);
JButton btClear = new JButton("Очистити дані");
commands.add(btClear);
ClearListener clearLst = new ClearListener();
btClear.addActionListener(clearLst);
JButton btClose = new JButton("Завершити роботу");
// до кнопки закриття вікна додаємо обробник анонімно
btClose.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent a){
        setVisible(false);
        System.exit(0);
    }
});
commands.add(btClose);
components.add(commands, java.awt.BorderLayout.NORTH);
//кнопки з діями додаються на спеціальну панель меню,
//що потім додається в верхню частину вікна
//щоб кнопки діяли, їм слід створити призначити відповідні
//обробники подій
pack();//команда для завантаження малюнка форми в оперативну пам'ять
setVisible(true);
}

private class ClearListener implements ActionListener{
    // "слухач", що відповідає на натискання кнопки "Очистити дані"
    @Override
    public void actionPerformed(ActionEvent a) {
        authorField.setText("");
        titleField.setText("");
        yearField.setText("");
        pagesField.setText("");
        dateField.setText("");
        issueCheck.setSelected(false);
    }
}

private class AddBookListener implements ActionListener{
    // "слухач", що відповідає на натискання кнопки "Додати книгу"
    @Override
    public void actionPerformed(ActionEvent a) {

        String autor = authorField.getText();

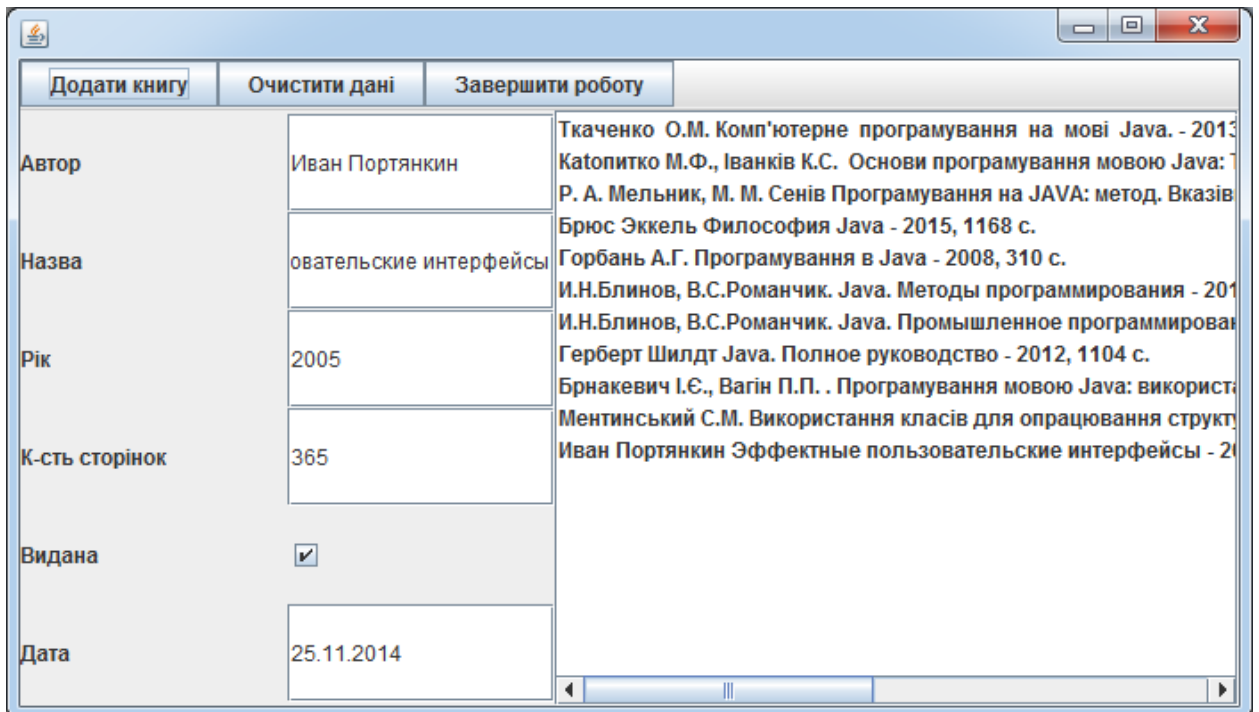
```

```

String title = titleField.getText();
int year = Integer.parseInt(yearField.getText());
int pages = Integer.parseInt(pagesField.getText());
Book newBook = new Book(autor,title,year,pages);
if (issueCheck.isSelected()){
    try {
        newBook.setIssue(dateField.getText());
    } catch (ParseException e) {
        e.printStackTrace();
    }
}
myLibrary.add(newBook);
}
}

public static void main(String[] args) {
    LibraryWindow form1 = new LibraryWindow();
}
}

```



## Лабораторна робота № 14

### Використання компонентів swing для проектування зручного GUI.

**Мета роботи:** знайомство з графічними елементами пакету swing, їх використанням для роботи з даними різного типу та використанням набору їх подій для керування візуальними проектами.

**Завдання .** Вдосконалити компонентну структуру проекту попередньої лабораторної роботи, замінивши деякі його елементи елементами з більшою функціональністю та додавши нові можливості, зокрема:

1. замінити текстові поля, куди вводиться великий текст на текстові вікна JTextArea;
2. текстові поля для вводу числової інформації та дат замінити більш зручними, наприклад полями зі списками чи лічильниками,
3. зв'язати подіями встановлення/відключення прапорця JCheckBox з доступністю/вимкненням відповідних полів вводу,
4. додати запис даних у файл, як реакцію на закривання вікна,
5. додати кнопки для вилучення виділеного у списку об'єкта та запису списку у файл з можливістю вибору файла.

#### **Варіанти завдань**

Варіанти завдань відповідають лабораторній роботі № 9.

#### **Приклад.**

Варіант 16. Предметна область: бібліотека, клас: книга, орієнтовний перелік полів: назва, автор, рік видання, кількість сторінок, наявна у фонді/видана читачеві дата видачі.