

ЛАБОРАТОРНА РОБОТА 5

Тема: Розробка графічного віконного інтерфейсу.

Мета: отримати навички створення додатків із графічним інтерфейсом користувача. Вивчити можливості пакета Swing.

Теоретичні питання

1. Можливості та особливості бібліотек Swing і AWT.
2. Поняття і принципи Usability.
3. GUI-редактор IDEA і його можливості.
4. Клас JFrame.
5. Класи графічних компонент JPanel, JLabel, JButton (і його нащадки), JTextBox, JTextArea, JComboBox, JScrollPane, JSlider, JTabbedPane.
6. Допоміжні класи: Color, Dimension, Font.
7. Класи розміщень (layouts).
8. Вирівнювання елементів. Ручне і автоматичне задання розмірів елементів.
9. Події. Механізм обробки подій Swing.
10. Меню і панелі інструментів. Обробка їх подій.

Контрольні питання

1. Перерахуйте основні відмінності бібліотек Swing і AWT. Яку з них краще використовувати і чому?
2. Перерахуйте основні стандарти, що регулюють розташування елементів екранної форми.
3. Наведіть приклад коду, що створює вікно і кнопку на ньому.
4. Як за допомогою методів класу JFrame можна динамічно змінювати зовнішній вигляд вікна?
5. У якому вигляді зберігається графічний дизайн форми, створений GUI-редактором?
6. Коротко опишіть призначення та основні методи класів графічних компонентів: JPanel, JLabel, JButton, JTextBox, JTextArea, JComboBox.

7. Які класи розміщень (layouts) реалізовані в Swing і чим вони відрізняються?
8. Опишіть призначення класів Color, Dimension, Font і наведіть приклади їх використання.
9. Як можна задати вирівнювання компонента, його мінімальний і максимальний розміри?
10. Як у Swing обробляються події GUI?
11. Що таке слухач (listener) і які стандартні класи слухачів є у Swing?
12. Чим відрізняється Adapter від Listener?
13. Як використовувати абстрактні класи MouseAdapter і ActionListener для обробки натискання кнопки (JButton) мишкою? Наведіть приклади.
14. Як можна обробити поєднання клавіш Ctrl-Alt-F5 за допомогою KeyAdapter?
15. Опишіть основні методи класів JMenu і JToolBar. Як найлегше додати однакові обробники і для пункту меню, і для відповідної кнопки панелі інструментів?

Індивідуальні завдання

Загальне завдання

1. Створіть вікно входу до системи. Вікно повинне містити:
 - текстові поля для введення логіна і пароля з підписами (введений пароль не має бути видно);
 - кнопки «Увійти», «Реєстрація» і «Скасувати».Вікно має відповідати стандартам розташування елементів управління.
При виборі «Скасувати» робота програми завершується.
При виборі «Увійти» виконується перевірка правильності логіна і пароля і видається повідомлення про результат перевірки.
При виборі «Реєстрація» вікно переходить у режим реєстрації нового користувача (змінюється заголовок вікна, з'являється нове поле для підтвердження пароля, ховається кнопка «Увійти»).

Для зберігання логінів і паролів у пам'яті використовуйте колекцію Map.

Реалізуйте серіалізацію цих даних до файла.

(* Для програмістів) Паролі в пам'яті і на диску зберігайте тільки в зашифрованому вигляді. Для шифрування використовуйте клас MessageDigest і алгоритм SHA-512.

2. Створіть додаток (за *варіантом*) із віконним інтерфейсом. Вікно додатка має містити меню і панель інструментів. Усі дії мають виконуватися декількома способами: через меню, кнопки панелі інструментів, гарячі клавіші.

Реалізуйте збереження даних програми у файл і завантаження з файла.

Перелік обов'язкових пунктів меню:

- File: New, Open, Save, Save as, Exit;
- Help: About.

Варіант 1

2. Гра «Пазл». На початку гри на формі у довільний спосіб розміщуються прямокутні фрагменти зображення-пазла. Далі користувач може виділяти і переміщати фрагменти пазла за допомогою мишки, а також за допомогою клавіш-стрілок. Гра завершується, коли всі фрагменти пазла розміщені у правильному порядку.

Для відображення кожного фрагмента використовуйте JLabel або JButton із зображенням. При виділенні у фрагмента має з'являтися червона рамка. Після кожного ходу програма повинна автоматично визначати, чи завершена гра.

(Для непрограмістів) Фрагменти картини можна підготувати заздалегідь у вигляді окремих файлів, тобто набір фрагментів не змінюється, змінюється тільки їх розташування у вікні програми.

(* Для програмістів) Реалізуйте можливість вибору вихідної картини, кількості фрагментів пазла, а також автоматичне розділення картини на фрагменти.

Варіант 2

2. Гра «Реверс». Ігрове поле 6×6 складається з клітинок, кожна з яких забарвлена в один із двох кольорів – білий або синій. На початку гри клітинки фарбуються у випадковий спосіб. У процесі гри гравець може змінювати колір клітинок, натискаючи на них. При натисканні на клітинку одночасно змінюється колір усіх клітинок у її рядку та в її стовпці. Мета гри – пофарбувати всі клітинки в один колір.

Вікно повинне мати текстовий напис, що повідомляє, скільки клітинок кожного кольору наразі є на полі. Після кожного ходу програма повинна автоматично визначати, чи завершена гра.

Для клітинок поля використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля, реалізуйте режим ручного завдання кольорів клітинок на початку гри.

Варіант 3

2. Гра «Сапер». Правила гри аналогічні до стандартної гри Windows. Використовуйте ігрове поле 10×10. Вікно повинне мати текстовий напис, що повідомляє, скільки «мін» залишилося не знайденими. Після кожного ходу програма повинна автоматично визначати, чи завершена гра і чи виграв гравець.

Для комірок використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля і кількості «мін». Реалізуйте автоматичне каскадне розкриття сусідніх комірок при розкритті комірки з нульовим значенням.

Варіант 4

2. Програма «Щоденник». У кожен момент часу вікно додатка має відображати один запис щоденника, який включає:

- назву події (текстове поле);
- докладний опис (багаторядкове текстове поле);
- дату, час (текстові поля);
- прапорець «важлива подія».

У вікні також мають бути кнопки «Наступна», «Попередня» для переходу між подіями.

Має бути можливість редагування всіх атрибутів події. За кнопкою «Застосувати» зміни повинні зберігатися в пам'яті.

У файл зберігайте всі записи щоденника.

(* Для програмістів) Реалізуйте перевірку правильності введення дати і часу. Реалізуйте фільтр записів за датою (для пошуку задається діапазон дат), а також за довільною фразою в тексті опису події.

Варіант 5

2. Програма «Адресна книга». У кожен момент часу вікно додатка має відображати один запис адресної книги, який включає:

- ім'я людини (текстове поле);
- її телефон та e-mail (текстові поля);
- групу (список, що випадає), наприклад, «Друзі», «Знайомі»;
- прапорець «доданий до обраних контактів»;
- день народження (текстове поле).

У вікні мають бути кнопки «Наступний», «Попередній» для переходу між записами.

Має бути можливість редагування всіх полів даних. За кнопкою «Застосувати» ці зміни мають зберігатися в пам'яті.

У файл зберігайте всі записи адресної книги.

(* Для програмістів) Реалізуйте перевірку правильності введення телефону та e-mail. Реалізуйте можливість сортування записів за різними атрибутами.

Варіант 6

2. Програма «Замовлення». У кожен момент часу вікно додатка має відображати інформацію про одне замовлення, яка включає:

- дату замовлення, найменування товару, ім'я замовника і його телефон (текстові поля);
- категорію товару (список, що випадає), наприклад, «Ноутбук», «Нетбук», «Планшет»;
- кількість (текстове поле, допустимо тільки ціле число);
- адресу замовника (багаторядкове текстове поле);
- прапорці «Терміновий» і «Оплата готівкою».

У вікні мають бути кнопки «Наступний», «Попередній» для переходу між записами.

Має бути можливість редагування всіх полів даних. За кнопкою «Застосувати» ці зміни мають зберігатися в пам'яті.

У файл зберігайте всі замовлення.

(* Для програмістів) Реалізуйте вибір найменування товару зі списку, який автоматично формується залежно від обраної категорії товару. Реалізуйте фільтр записів за категорією товару, а також за ознаками «Терміновий» і «Оплата готівкою».

Варіант 7

2. Гра «Морський бій». Користувач грає проти комп'ютера. Комп'ютер розміщує «кораблі» на полі у випадковий спосіб, а гравець повинен їх знайти. Використовуйте ігрове поле 8×8. Гравець виграв, якщо він знаходить усі «кораблі» противника, і програє, якщо не знаходить їх за 25 ходів. У вікні мають бути текстові написи, скільки «кораблів» наразі «вбито», «поранено» і скільки ще не знайдено. Після кожного ходу програма має автоматично визначати, чи завершена гра.

Для комірок використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість гри двох гравців, зокрема й режим початкової розстановки «кораблів» гравцями вручну.

Варіант 8

2. Гра «Хрестики-нулики». Ігрове поле 5×5 складається з клітинок, у кожній із яких може стояти хрестик, нулик або порожньо. Гравці по черзі ставлять хрестики і нулики в порожні клітинки поля. Гра завершується, коли один з гравців має три хрестика або нулика в ряд. Вікно повинне мати текстовий напис, що повідомляє, який гравець повинен ходити. Після кожного ходу програма повинна автоматично визначати, чи завершена гра, і хто виграв. Для клітинок поля використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля. Реалізуйте можливість гри з комп'ютером.

Варіант 9

2. Гра «Реверс». Ігрове поле 7×7 складається з клітинок, кожна з яких забарвлена в один із двох кольорів – білий або синій. На початку гри клітинки фарбуються у випадковий спосіб. У процесі гри гравець може змінювати колір клітинок, натискаючи на них. При натисканні на клітинку одночасно змінюється колір усіх клітинок у її рядку та в її стовпці. Мета гри – пофарбувати всі клітинки в один колір.

Вікно повинне мати текстовий напис, що повідомляє, скільки клітинок кожного кольору наразі є на полі. Після кожного ходу програма повинна автоматично визначати, чи завершена гра.

Для клітинок поля використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля, реалізуйте режим ручного задання кольорів клітинок на початку гри.

Варіант 10

2. Гра «Сапер». Правила гри аналогічні до стандартної гри Windows. Використовуйте ігрове поле 15×15. Вікно повинне мати текстовий напис,

що повідомляє, скільки «мін» залишилося не знайденими. Після кожного ходу програма повинна автоматично визначати, чи завершена гра і чи виграв гравець.

Для комірок використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля і кількості «мін». Реалізуйте автоматичне каскадне розкриття сусідніх комірок при розкритті комірки з нульовим значенням.

Варіант 11

2. Програма «Щоденник». У кожен момент часу вікно додатка має відображати один запис щоденника, який включає:

- назву події (текстове поле);
- докладний опис (багаторядкове текстове поле);
- дату, час (текстові поля);
- прапорець «важлива подія».

У вікні також мають бути кнопки «Наступна», «Попередня» для переходу між подіями.

Має бути можливість редагування всіх атрибутів події. За кнопкою «Застосувати» зміни повинні зберігатися в пам'яті.

У файл зберігайте всі записи щоденника.

(* Для програмістів) Реалізуйте перевірку правильності введення дати і часу. Реалізуйте фільтр записів за датою (для пошуку задається діапазон дат), і за довільною фразою в тексті опису події.

Варіант 12

2. Програма «Адресна книга». У кожен момент часу вікно додатка має відображати один запис адресної книги, який включає:

- ім'я людини (текстове поле);
- його телефон та e-mail (текстові поля);

- групу (список, що випадає), наприклад, «Друзі», «Знайомі»;
- прапорець «доданий до обраних контактів»;
- день народження (текстове поле).

У вікні мають бути кнопки «Наступний», «Попередній» для переходу між записами.

Має бути можливість редагування всіх полів даних. За кнопкою «Застосувати» ці зміни повинні зберігатися в пам'яті.

У файл зберігайте всі записи адресної книги.

(* Для програмістів) Реалізуйте перевірку правильності введення телефону та e-mail. Реалізуйте можливість сортування записів за різними атрибутами.

Варіант 13

2. Програма «Замовлення». У кожен момент часу вікно додатка має відображати інформацію про одне замовлення, яка включає:

- дату замовлення, найменування товару, ім'я замовника і його телефон (текстові поля);
- категорію товару (список, що випадає), наприклад, «Ноутбук», «Нетбук», «Планшет»;
- кількість (текстове поле, допустиме тільки ціле число)
- адресу замовника (багаторядкове текстове поле);
- прапорці «Терміновий» і «Оплата готівкою».

У вікні мають бути кнопки «Наступний», «Попередній» для переходу між записами.

Має бути можливість редагування всіх полів даних. За кнопкою «Застосувати» ці зміни повинні зберігатися в пам'яті.

У файл зберігайте всі замовлення.

(* Для програмістів) Реалізуйте вибір найменування товару зі списку, який автоматично формується залежно від обраної категорії товару. Реалізуйте фільтр записів за категорією товару, а також за ознаками «Терміновий» і «Оплата готівкою».

Варіант 14

2. Гра «Морський бій». Користувач грає проти комп'ютера. Комп'ютер розміщує «кораблі» на полі у випадковий спосіб, а гравець повинен їх знайти. Використовуйте ігрове поле 10×10. Гравець виграє, якщо він знаходить всі «кораблі» противника, і програє, якщо не знаходить їх за 25 ходів. У вікні мають бути текстові написи, скільки «кораблів» наразі «вбито», «поранено» і скільки ще не знайдено. Після кожного ходу програма повинна автоматично визначати, чи завершена гра.

Для комірок використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість гри двох гравців, зокрема й режим початкової розстановки «кораблів» гравцями вручну.

Варіант 15

2. Гра «Хрестики-нулики». Ігрове поле 6×6 складається з клітинок, у кожній з яких може стояти хрестик, нулик або порожньо. Гравці по черзі ставлять хрестики і нулики в порожні клітинки поля. Гра завершується, коли один із гравців має три хрестики або нулики в ряд. Вікно повинне мати текстовий напис, що повідомляє, який гравець повинен ходити. Після кожного ходу програма повинна автоматично визначати, чи завершена гра, і хто виграв.

Для клітинок поля використовуйте кнопки з картинками. У файл зберігайте поточний стан гри.

(* Для програмістів) Реалізуйте можливість вибору розміру ігрового поля. Реалізуйте можливість гри з комп'ютером.

Додаток А

Каркас мінімального GUI-додатка

```
public class MyWnd {                                // Клас додатка

    private JFrame frame;                          // Головне (і єдине) вікно

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {    // Для створення вікна
            public void run() {                    // зазвичай запускається
                try {                               // окремий потік.
                    MyWnd window = new MyWnd();    // Створюємо вікно
                    window.frame.setVisible(true); // Відображаємо його
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public MyWnd() { initialize(); }                // Конструктор класу додатка

    private void initialize() {                     // Створення вікна і його компонентів
        frame = new JFrame();                       // Створюємо саме вікно
        frame.setBounds(100, 100, 450, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Тут код створення компонентів і додавання для них обробників
    }
}
```

Приклад. Простий текстовий редактор

Редактор дає змогу:

- вводити ім'я файла безпосередньо до текстового поля (JTextField);
- вибирати файл за допомогою стандартного діалогу відкриття файла (JFileChooser) за кнопкою «Open»;
- редагувати вміст файла у багаторядковому текстовому полі (JTextArea);
- берігати зміни у файлі за кнопкою «OK»;

– вийти з програми за кнопкою «Cancel», відображаючи попередження перед ВИХОДОМ.

Вікно додатка має вигляд, показаний на рис. 6.1:

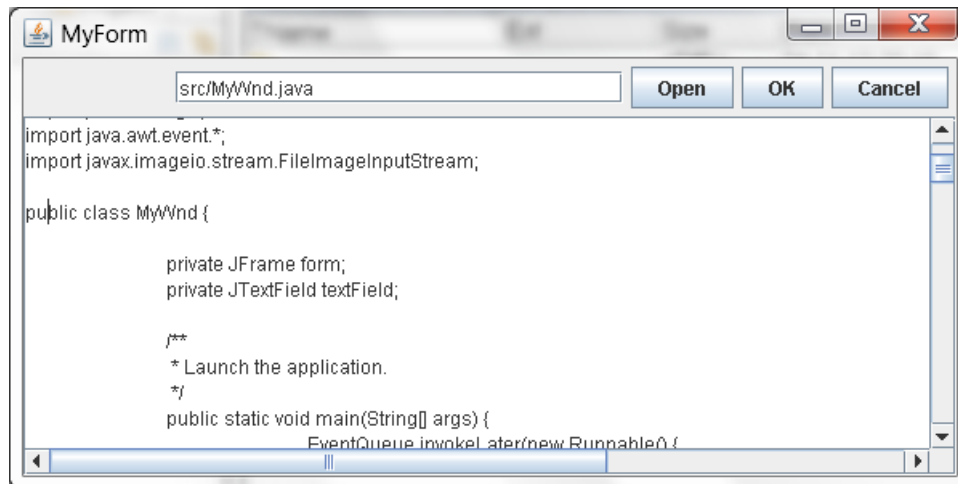


Рис. 6.1 – Вікно додатка редактора

Код програми:

```
import java.io.*;
import java.util.Scanner;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class MyApp { // Клас додатка

    private JFrame form; // Головне вікно
    private JTextField txtName; // Pole для імені файла
    private JTextArea txtFile; // Вміст файла

    public static void main(String[] args) {
        MyApp app = new MyApp(); // Створити об'єкт - додаток
        app.form.setVisible(true); // Показати його вікно
    }

    public MyApp() {
        // Створюємо вікно
        form = new JFrame();
        form.setTitle("MyForm");
    }
}
```

```

form.setBounds(100, 100, 600, 300);
form.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Розміщення для елементів головної панелі
Container pane = form.getContentPane();
pane.setLayout(new BorderLayout(pane, BorderLayout.Y_AXIS));

// Створюємо верхню панель
JPanel panel = new JPanel();
panel.setLayout(new FlowLayout(FlowLayout.RIGHT));
panel.setMaximumSize(new Dimension(600,30));
pane.add(panel);

// Додаємо елементи верхньої панелі
txtName = new JTextField();
txtName.setColumns(25);
panel.add(txtName);
// Перевірка правильності введення
txtName.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        if (!txtName.getText().Matches(".*\\.html?"))
            txtName.setForeground(new Color(255, 100, 100));
        else txtName.setForeground(Color.BLACK);
    }
});
JButton btnOpen = new JButton("Open");
panel.add(btnOpen);
JButton btnOK = new JButton("OK");
panel.add(btnOK);
JButton btnCancel = new JButton("Cancel");
panel.add(btnCancel);

// Створюємо поле з прокруткою для виведення текста
txtFile = new JTextArea("No file selected");
JScrollPane scroll = new JScrollPane(txtFile);
form.getContentPane().add(scroll);

// Додаємо обробники:
// для кнопки Open
btnOpen.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        // Створюємо діалог вибору файла
        JFileChooser fc = new JFileChooser();
        // Якщо файл обрано
        if (JFileChooser.APPROVE_OPTION ==

```

```

        fc.showOpenDialog(form)
        // то його ім'я записуємо в txtName
        txtName.setText( fc.getSelectedFile()
                        .getAbsolutePath());
    }
});
// для кнопки ОК
btnOK.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try {
            // Створюємо об'єкт File
            File f = new File(txtName.getText());
            // Перевіряємо, що файл існує
            if (!f.exists()) txtFile.setText("No such file");
            else {
                // Відкриваємо файл для читання
                Scanner sc = new Scanner(f); String s = "";
                // і записуємо весь його вміст у поле txtFile
                while (sc.hasNextLine())
                    s = s + sc.nextLine() + "\n";
                txtFile.setText(s);
            }
        } catch (Exception e) {
            // Виводимо повідомлення про помилку
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
});
// для кнопки Cancel
btnCancel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        // Створюємо діалог підтвердження з кнопками ОК і CANCEL
        if (JOptionPane.OK_OPTION ==
            JOptionPane.showConfirmDialog(form,
                "Exit without saving?", "Exit",
                JOptionPane.OK_CANCEL_OPTION))
            // Якщо натиснуто ОК, то завершуємо програму
            System.exit(0);
    }
});
}
}
}

```