­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Лабораторна робота №8

з курсу «Об’єктно-орієнтоване програмування»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 14

Виконав студент гр. КНз-2

Чалий Михайло

­­

Львів 2014

# Принципи побудови графічного інтерфейса

## Мета роботи

Ознайомитися з принципами побудови графічного інтерфейса у Java.

## Теоретичні відомості

Графічна бібліотека AWT пропонує більше двадцати готових компонентів. Найбільш часто використовуються підкласи класу Component: класи Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, TextArea, TextField, Panel, ScrollPane, Window, Dialog, FileDialog, Frame. Ще одна група компонентів — це компоненти меню — класи Menuitem, MenuBar, Menu, PopupMenu, CheckboxMenuItem.

Клас Сomponent — центр бібліотеки AWT — дуже великий і володіє багатьма можливостями. В ньому пять статичних констант, визначаючих розміщення компонента всередині простору, виділеного для компонента у вміщаючому його контейнері: BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT, і близько сотні методів. Більшість методів— це методи доступу getxxx(), isxxx(), setxxx(). Вивчати їх немає рації, треба просто подивитися, як вони використовуються в підкласах.

Конструктор класу недоступний — він захищений (protected), тому, що клас Сomponent абстрактний, він не може використовуватися сам по собі, використовуються лише його підкласи. Компонент завжди займає прямокутну область зі сторонами, паралельними сторонам екрана і в кожний момент часу має певні розміри, вимірювані в пікселях, які можна узнати методом getSize(), повертаючим обєкт класу Dimension, або цілочисельними методами getHeight() і getWidth(), повертаючими висоту і ширину прямокутника. Новий розмір компонента можна установити із програми методами setSize(Dimension d) або setSize(int width, int height), якщо це дозволяє менеджер розміщення контейнера, що містить компонент. У компонента єсть оптимальний розмір, при якому він виглядає найбільш пропорціонально. Його можна одержати методом getPreferredSize() у вигляді обєкта Dimension.

Компонент має мінимальний і максимальний розміри. Їх повертають методи getMinimumSize() і getMaximumSize() у вигляді обєкта Dimension. В компоненті єсть система координат. Її початок - точка з координатами (0, 0) - знаходиться в лівому верхньому куті компонента, вісь Ох іде вправо, вісь Оу - вниз, координатні точки розташовані між пікселями. В компоненті зберігаються координати його лівого верхнього кута в системі координат вміщаючого його контейнера. Їх можна узнати методами getLocation(), а зменити — методами setLocation(), перемістивши компонент в контейнері, якщо це дозволить менеджер розміщення компонентів.

Можна вияснити зразу і положення, і розмір прямокутної області компонента методом getBounds(), повертаючим обєкт класу Rectangle, і змінити разом і положення, і розмір компонента методами setBounds(), якщо це дозволить зробити менеджер розміщення. Компонент може бути недоступним для дій користувача, тоді він виділяється на екрані світло-сірим кольором. Доступність компонента можна перевірити логічним методом isEnabІed(), а змінити— методом setEnabled(boolean enable). Для багатьох компонентів визначається графічний контекст — обєкт класу Graphics, — який керується методом paint(), описаним в попередньому уроці, і який можна одержати методом getGraphics().

В контексті єсть поточний колір і колір фону — обєкти класу Сolor. Колір фону можна одержати методом getBackground(), а змінити— методом setBackground(Color color). Поточний колір можна одержати методом getForeground(), а змінити - методом setForeground(Color color). В контексті єсть шрифт — обєкт класу Font, що повертається методом getFont() і змінюється методом setFont(Font font). В компоненті визначається локаль — обєкт класу Locale. Його можна одержати методом getLocale(), змінити - методом setLocale(Locale locale).

Контейнер Frame — це повноцінне готове вікно з рядком заголовку, в який поміщені кнопки контекстного меню, звертання вікна в ярлик і розвертання на весь екран і кнопка закрития додатку. Заголовок вікна записується в конструкторі або методом setTitІe(String title). Вікно обмежено рамкою. В нього можна установити рядок меню методом setMenuBar (MenuBar mb). Це ми обговоримо в уроці13*.* На кнопці контекстного меню в лівій частині рядка заголовка зображена димляча чашечка кофе — логотип Java. Ви можете установити там друге зображення методом setІconІmage(image icon), створивши попередньо зображення icon у вигляді обєкта класу image. Як це зробити, пояснюється в уроці15*.* Всі элементи вікна Frame викреслюються графічною оболонкою операційної системи по правилах цієї оболонки. Вікно Frame автоматично реєструється у віконному менеджері графічної оболонки і може переміщаться, змінювати розміри, звертаться в панель завдань (task bar) за допомогою миші або клавіатури, як "рідне" вікно операційної системи. Створити вікно типу Frame можна наступними конструкторами:

* Frame() — створює вікно з пустим рядком заголовка;
* Frame(ІString title) — записує аргумент title в рядок заголовка.

Методи класу Frame здійснюють доступ до елементів вікна, але не забувайте, що класс Frame наслідує близько двохсот методів класів Component, Container і Window. Зокрема, наслідується менеджер розміщення по замовчуванню — BorderLayout.

Кріме подій класу Component: ComponentEvent, FocusEvent, KeyEvent, MouseEvent, при зміні розмірів вікна, його переміщенні або видаленні з екрана, а також показу на екрані відбувається подія windowEvent.

## Завдання

При клацанні мишею по зображенню в графічному вікні воно міняється на інше зображення такого ж розміру, а при повторному клацанні відновлюється первинне зображення.

## Реалізація

Лістінг 3. PlayImage.java

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.io.File;

**import** java.io.IOException;

**import** javax.imageio.ImageIO;

**import** javax.swing.JOptionPane;

**class** PlayImage **extends** Frame {

**private** **static** **final** **long** *serialVersionUID* = -5616509887780478705L;

**private** Image img1;

**private** Image img2;

**private** **boolean** drawSecond;

PlayImage(String s) {

**super**(s);

**try** {

img1 = ImageIO.*read*(**new** File("img1.jpg"));

img2 = ImageIO.*read*(**new** File("img2.jpg"));

} **catch** (IOException e) {

JOptionPane.*showMessageDialog*(**this**,"Помилка читання файлів картинок. " + e.getMessage());

System.*exit*(0);

}

setSize(500, 281);

setVisible(**true**);

addWindowListener(**new** WindowAdapter() {

**public** **void** windowClosing(WindowEvent ev) {

System.*exit*(0);

}

});

addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

toggleImage();

}

});

}

@Override

**public** **void** paint(Graphics g){

Image current = drawSecond ? img2 : img1;

**int** width = current.getWidth(**this**);

**int** height = current.getHeight(**this**);

g.drawImage(current, 0, 0, width, height, **this**);

}

**private** **void** toggleImage(){

drawSecond = !drawSecond;

**this**.repaint();

}

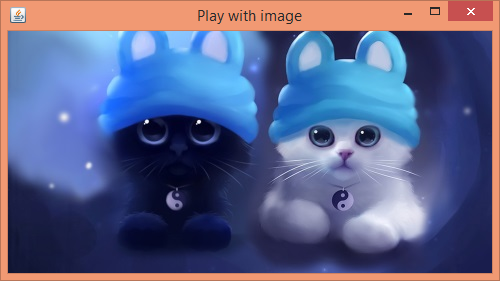
**public** **static** **void** main(String[] args) {

**new** PlayImage("Play with image");

}

}

## Результат



## Висновки

Ознайомився з принципами побудови графічного інтерфейса у Java.