Реферат з “програмування Java”

Тема: Swing

Підготував студент 3-го курсу

Комп’ютерної математики

Чорний Владислав Мирославович

**Огляд**

Swing API - це бібліотека графічного інтерфейсу, для створення за стосунків на базі Java Front End / GUI.

Головні ознаки:

* Легка вага
* Багато елементів контролю
* Широкі можливості налаштування
* Можливість зміни зовнішнього вигляду

Архітектура:

- Елементи інтерфейсу – загальні візуальні елементи, з якими взаємодіє користувач.

- Макети – вони визначають, як елементи інтерфейсу повинні розташовуватися на екрані.

- Поведінка – це події, які відбуваються, коли користувач взаємодіє з елементами інтерфейсу.

Кожен елемент управління Swing наслідує властивості з наступних класів:

* Компонент – абстрактний базовий клас для елементів управління без користувацького інтерфейсу, але має графічне представлення.
* Контейнер – компонент, який може містити інші компоненти Swing.
* *JComponent* – базовий клас для усіх компонентів користувацького інтерфейсу Swing.

**Перелік елементів користувацького інтерфейсу Swing**

* *JLabel* – компонент для розміщення тексту в контейнері.
* *JColorChooser* – панель, за допомогою якої користувач може контролювати колір.
* *JCheckBox* – кнопка, яка має два стани: true та false.
* *JRadioButton* – аналогічно до JCheckBox тільки в группі.
* *JList* – список текстових елементів.
* *JComboBox* – меню вибору.
* *JTextField* – текстовий компонент для редагування тексту в одному рядку.
* *JPasswordField* – текстовий компонент для вводу паролю.
* *JTextArea* – аналогічно до JTextField тільки декілька рядків.
* *ImageIcon* – реалізація значків із зображень.
* *JOptionPane* – реалізація діалогових вікон.
* *JFileChooser* – реалізація вікна з якого можна обрати файл.
* *JSlider* – дозволяю користувачу обирати значення, шляхом переміщення повзунка.
* *JSpinner* – поле вводу для обрання числа або значення об’єкта з упорядкованої послідовності.

**Класи подій**

Класи подій обробляють події. Розглянемо найбільш використовувані з них

Клас *EventObject*

Це кореневий клас з якого отримуються усі об’єкту зі станом події. Даний пакет визначений в *java.util.*

Декларація класу:

*public class EventObject*

*extends Object*

*implements Serizlizable*

Конструктор класу

*EventObject(джерело об’єкта)*

Створює прототип події.

Методи класу

*Об’єкт getSource ()*

Об’єкт на якому спочатку відбулася подія.

*Строка toString ()*

Повертає рядкове представлення цього *EventObject*.

Деякі класи подій Swing

*java.awt.Event* заміняє клас подій для всіх подій Swing

- *ActionEvent* генерується при натисканні на кнопку або подвійному клікі миші по елементу списку.

- *InputEvent* являється кореневою подієб для усіх вхідних подій компонента.

**Макет**

Макет відповідає за розташування компонентів всередині контейнера. Дана задача виконується автоматично менеджером пакетів, але може бути виконана й вручну.

Інтерфейси, які реалізують менеджер з розташування:

-*LayoutManager* – визначає методи, які повинні бути реалізовані класом, чий об’єкт будет виступати менеджером макету.

-*LayoutManager2* – залежить від інтерфейсу LayoutManager. Даний інтерфейс для тих класів, які знають, як макетувати контейнери на основі об’єкту обмеження макету.

AWT Layout Manager Classes:

* *CardLayout* – роглядає кожний компонент в контейнері як карту. Тільки одна карта видима одночасно.
* *FlowLayout* – макет за замовченням. Визначає розташування об’єктів у відповідному напрямку.
* *GridLayout* – управляє компонентами в формі прямокутної сітки.
* *GroupLayout* – ієрархічно групує компоненти, задля розташування їх в контейнері.
* *SpringLayout* – розміщує дочірні елементи свого зв’язанного контейнера у відповідності з набором обмежень.

**Меню класів**

З кожним вікном верхнього рівня зв’язаний рядок меню. Даний рядок складається з різних варіантів меню, доступних для користувача. Далі, кожний варіант містить список параметрів, який називається випадним меню. Елементи управління *Menu* та *MenuItem* являються підкласами класу *MenuComponent*.

* *JMenuBar* – пов’язаний з вікном верхного рівня. Елементи в даному меню повинні належати *JMenuItem*.
* *JMenu* являється компонентом випадного меню.
* *JCheckboxMenuItem* – підклас *JMenuItem*.
* *JRadioButtonMenuItem* – підкласс *JMenuItem*.
* *JPopupMenu* – може динамічно з’являтися у вказаній позиції компонента.

**Контейнери**

Контейнер це компонент в якому можна розташовувати інші компоненти.

Найбільш використовування контейнерів:

* *JPanel* – в нього може бути поміщений будь-який інший компонент та інші панелі.
* *JFrame* – вікно верхнього рівня із заголовком та рамкою
* *JWindow* – вікно верхнього рівня без границь та меню.