

Kurs programowania

Wykład 4

Wojciech Macyna

Graficzny interfejs użytkownika - GUI

W Javie możemy skorzystać z trzech bibliotek do tworzenia graficznych interfejsów:

Awt (Abstract Windowing Toolkit) – podstawowa biblioteka będąca interfejsem do graficznych funkcji systemu. Stąd wygląd komponentów zależy od systemu.

Biblioteka: `java.awt.*`

Swing – biblioteka w Javie niezależna od systemu (ale wykorzystująca niektóre elementy AWT). Stąd identyczny wygląd GUI na różnych platformach. Komponenty mają podobne nazwy jak w AWT ale poprzedzone literą J (np. Frame w AWT i JFrame w SWING).

Biblioteka: `javax.swing.*`

JavaFX – najnowsza biblioteka od obsługi grafiki.

Biblioteka: `javaFX.*`

Na wykładzie zostanie omówiona głównie biblioteka Awt oraz JavaFX.

Główne konstruktory

- `public Frame()`
- `public Frame(String title)`
- `public Frame(String title, GraphicsConfiguration gc)`

Główne metody

- `public void setSize(int width, int height)` [domyślny rozmiar 20x20]
- `public void setBounds(int x, int y, int width, int height)`
- `public void pack()` [dopasowuje rozmiar okna do zawartości]
- `public void setVisible(boolean show)`

Kontenery i komponenty

W GUI rozróżniamy dwa podstawowe rodzaje elementów: komponenty i zawierające (wyświetlające) je kontenery.

Podstawowe typy komponentów AWT

Button, CheckBox, Choice, Label, List, TextField, TextArea, Menu, ...

Podstawowe typy kontenerów AWT

Frame, Panel, ScrollPane

Uwaga: Panel jest także komponentem.

Dodawanie komponentu do konteneru

Metoda `public Component add(Component c)`

Przykłady

przyklad1.java

Kontener powinien użyć zarządcy wyglądu ustawianego metodą:
`public void setLayout(LayoutManager m)`

Podstawowe wyglądy

- `java.awt.FlowLayout`
- `java.awt.BorderLayout`
- `java.awt.GridLayout`
- `java.awt.GridBagLayout`
- `java.awt.CardLayout`

Przykłady

`borderDemo.java`, `flowDemo.java`, `gridDemo.java`

Delegacyjny model zdarzeń

- Źródło zdarzenia – co jest źródłem zdarzenia.
- Słuchacz zdarzenia – kto słucha i jaką metodę powinien wywołać.
- Interfejs – zbiór metod odpowiedzialnych za obsługę zdarzeń.

Przykładowe interfejsy obsługi zdarzeń

- `java.awt.event.ActionListener` z metodą
`public void actionPerformed(ActionEvent e)`
- `java.awt.event.WindowListener` z metodami:
`public void windowOpened(WindowEvent e)`
`public void windowClosing(WindowEvent e)`
`public void windowClosed(WindowEvent e)`
`public void windowActivated(WindowEvent e)`
`public void windowDeactivated(WindowEvent e)`

Przykłady

przyklad2.java

Klasy implementujące interfejsy obsługi zdarzeń z pustymi metodami.

Podstawowe klasy adapterów

- WindowAdapter
- ComponentAdapter
- ContainerAdapter
- FocusAdapter
- KeyAdapter
- MouseAdapter

Główne konstruktory

- `public Button()`
- `public Button(String label)`

Główne metody

- `public void addActionListener(ActionListener a)`
- `public void setActionCommand(String c)`

Główne konstruktory

- `public Label()`
- `public Label(String text)`
- `public Label(String text, int alignment)` [wyrównanie za pomocą stałych `Label.LEFT`, `Label.RIGHT`, `Label.CENTER`]

Główne metody

- `public void setText(String text)`
- `public String getText()`

Klasa `java.awt.TextField`

Główne konstruktory

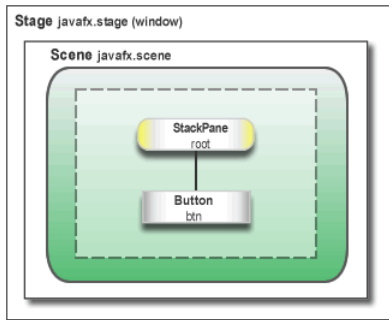
- `public TextField()`
- `public TextField(String text)`
- `public TextField(int columns)`
- `public TextField(String text, int columns)`

Główne metody

- `public String getText()`
- `public void setText(String text)`
- `public void setEditable(boolean b)`

Przykłady

program1.java



- Klasa główna rozszerza `javafx.application.Application`. Metoda `start()` jest główną metodą.
- Klasa `Stage` jest głównym kontenerem. Klasa `Scene` jest kontenerem dla kontentu.
- Kontent jest reprezentowany jako hierarchiczny graf węzłów.
- W tym przypadku korzeniem hierarchii jest layout `StackPane` posiadającym jedno dziecko typu `Button`.

Podstawowe wyglądy

- `javafx.scene.layout.BorderPane`
- `javafx.scene.layout.HBox`
- `javafx.scene.layout.VBox`
- `javafx.scene.layout.StackPane`
- `javafx.scene.layout.GridPane`
- `javafx.scene.layout.FlowPane`
- `javafx.scene.layout.TilePane`
- `javafx.scene.layout.AnchorPane`

Przykłady

FXEx1.java, FXGridDemo.java, FXBorderPane.java

Dobry opis:

`https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm`

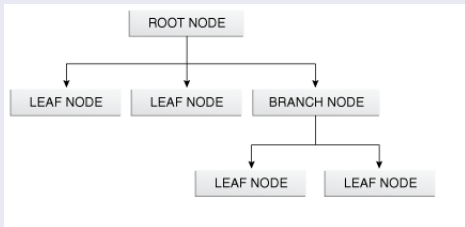
W przypadku nowych wersji Java (np. 17) należy doinstalować JavaFX

`https://taylorial.com/cs1021/Install.htm`

Kompilacja i uruchamianie

```
javac -module-path 'path' -add-modules javafx.controls FX.java  
java -module-path 'path' -add-modules javafx.controls FX  
path: ścieżka do katalogu z JavaFX
```

Hierarchia obiektów



Przykłady

FXHierarchy.java

Projekt z użyciem pliku fxml

Plik *FXMLDocument.fxml*

Zawiera dane na temat wyglądu okna wraz z jego elementami

Plik kontrolera: *FXMLDocumentController.java*

```
1  @FXML
2  private Button button;
3  @FXML
4  private void handleButtonAction(ActionEvent event) {
5      System.out.println("You clicked me!");
6      label.setText("FXML");
7  }
```

Plik główny: *FXMLEx.java*

Odpowiada między innymi za załadowanie pliku fxml.

```
1  Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));
2  Scene scene = new Scene(root);
3  .....
```