**UNIVERSITATEA DE STAT DIN MOLDOVA**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**DEPARTAMENTUL ”INFORMATICA”**

**Ciolpan Vlad**

**Prelucrarea evenimentelor în Java. Event oriented programing.**

**IA2001**

**Teză de an**

Chișinău-2022

Cuprins

Lista abrevierilor

Introducere

Capitol I:

1. Definirea evenimentului
2. Componente grafice
3. Tipuri de evenimente
4. Cum se generează un eveniment
5. Prelucrarea evenimentelor

Capitol II:

1. Java
2. JavaFX
3. Maven

Capitol III:

Concluzii

Bibliografie

Anexe

**Lista abrevierilor**

AWT – Abstract Window Toolkit

**Capitol I**

**Definirea evenimentului**

Eveniment – rezultatul interacțiunii utilizatorului cu aplicația ce are interfață grafică. De exemplu la apăsarea unui buton de pe tastatură, la mișcarea cursorului pe ecran.

În limbajul Java la baza oricărui eveniment se află clasa java.util.EventObject, precum și toate clasele din Java moștenesc de la clasa de bază java.lang.Object. Această clasă conține un singur câmp:

1. „*source*” de tip java.lang.Object – ce păstrează informația despre obiectul care a chemat acest eveniment.

și o metodă:

1. „*getSource()*” – returnează informația din câmpul „*source*”.

În continuare există 2 încrengături: java.awt.AWTEvent și javafx.event.Event, care moștenesc de la java.util.EventObject. Java.awt.AWTEvent este clasa de bază pentru toate evenimentele din instrumentarul AWT. Iar javafx.event.Event este clasa de bază pentru toate evenimentele folosite în JavaFX. Aceste 2 tehnologii vor fi explicate în Capitolul II. În această lucrare s-a folosit JavaFX, de aceea în continuare se vor descrie evenimentele și modul lor de lucru doar pentru această tehnologie.

Precum a fost menționat Javafx.event.Event este clasa de la care moștenesc evenimentele folosite în JavaFX. Astfel din punctul de vedere al computer-ului eveniment este un obiectul de tipul acestei clase sau al altor subclase care moștenesc de la această clasă.

Clasa javafx.event.Event conține câmpuri ce păstrează informația de bază despre eveniment necesară pentru funcționarea și la prelucrarea sa:

1. „*eventType*” de tipul EventType<? extends Event> - ce păstreză informația despre tipul evenimentului;
2. „target” de tipul EventTarget – ce joacă rol în construirea căii de parcurgere a evenimentului;
3. „*consumed*” de tipul boolean – ce reprezintă dacă evenimentul a fost prelucrat sau nu;

și metodele respective acestor câmpuri:

1. „*getEventType()*” – returnează informația din câmpul „*eventType*”;
2. „*getTarget()*” – returnează informația din câmput „target”;
3. „*isConsumed()*” – returnează informația din câmput „*consumed*”;
4. „*consume()*” – setează câmpul „*consumed*” cu valoarea true, ceea ce indică că evenimentul a fost prelucrat;
5. „*fireEvent(EventTarget eventTarget, Event event)*” – metoda statică care permite transmiterea unui eveniment „*event*” pe o anumită cale „*eventTarget*”.

**Componente grafice**

Componentă grafică se referă la toate componentele vizuale din care este formată o aplicație. În JavaFx toate componentele se exprimă ca noduri organizate în formă de arbore. Nodurile se împart în 3 tipuri: rădăcină, ramură și frunză. Nodul ramură este nodul care poate avea copii. Nodul rădăcină este un nod ramură, dar care are părinte. Nodul frunză este nodul care nu poate avea copii.

Frunză

Frunză

Frunză

Frunză

Ramură

Rădăcină

Figura 1.2.1

Clasa de bază pentru orice nod reprezintă clasa abstractă javafx.scene.Node care moștenește de la clasa java.lang.Object. Această clasă implementează interfața javafx.event.EventTarget care a fost menționată în primul paragraf în cadrul clasei javafx.event.Event necesar pentru construcția căii parcurgerii evenimentelor.

Clasa javafx.scene.Node are o mulțime de proprietăți care se rezumă la: id-ul, proprietățile fizice (locația și translația în sistemul de coordonate, mărimea, rotația), ascultărorii de evenimente, vizibilitate ș.a. Respectiv cu fiecare proprietate există și metodele care oferă acces la aceste proprietăți.

Clasa de bază pentru orice nod părinte reprezintă clasa abstractă javafx.scene.Parent care moștenește de la clasa javafx.scene.Node. Această clasă conține toate metodele necesare pentru a manipula cu ierahia de noduri, cum ar fi adaugărea și scoaterea nodurilor copii.

Nodurile frunze sunt caracterizate de orice altă clasă derivată de la clasa javafx.scene.Node înafară de clasa javafx.scene.Parent, cum ar fi:

1. javafx.scene.shape.Shape – clasa de bază pentru orice figură geometrică disponibilă
2. javafx.scene.shape.Shape3D – clasa de bază pentru orice corp geometric disponibil
3. javafx.scene.canvas.Canvas – clasă care permite „desenul” pe câmpul acestui nod
4. javafx.scene.image.ImageView – clasă care permite afișarea imaginilor
5. javafx.scene.media.MediaView – clasă care permite afișarea unui video sau pentru audio

De la clasa javafx.scene.Parent se extind 2 clase: javafx.scene.Group și javafx.scene.layout.Region. Group acționează ca un tot întreg, adică stilizând un Group se vor stiliza și nodurile copii al acestui nod. Pe când Region permite stilizarea fiecărui nod copil aparte.

Clasa javafx.scene.layout.Region reprezintă clasa de bază pentru componentele ce oferă control utilizatorului, cum ar fi buton, și pentru containerele care reprezintă componente ce pot păstra în interiorul lor alte componente. Ca proprietăți această clasă conține informarțiile necesare pentru stilizarea și afișarea unei componente, cum ar fi lungimea, lățimea, marginea ș.a. Respectiv fiecărei proprietăți corespund metodele necesare pentru manipularea informației din interior.

De la clasa javafx.scene.layout.Region moștenesc 4 clase:

1. javafx.scene.chart.Axis<T> – clasă de bază pentru axa de coordonate Oxy
2. javafx.scene.chart.Chart – clasă de bază pentru diagrame
3. javafx.scene.control.Control – clasă de bază pentru controale
4. javafx.scene.layout.Pane – clasă care reprezintă cel mai de bază container

Clasa javafx.scene.layout.Pane moștenește toate proprietățile de la clasa javafx.scene.layout.Region, astfel având proprietățile și metodele minime necesare pentru a putea funcționa ca un container.

De la această clasă moștenesc toate containerele existente în JavaFX, fiecare cu o funcțiune specială:

1. javafx.scene.layout.AnchorPane – panou în care componentele copii sunt fixate de panou la o anumită distanță față de margini, astfel permițând aranjarea componentelor mai ușoară
2. javafx.scene.layout.BorderPane – panou împărțit în zone, sus, jos, stânga, dreapta, centru, astfel în fiecare zonă pot fi amplasate câte o componentă sau să rămână zonă liberă
3. javafx.scene.control.DialogPane – panou considerat ca nod rădăcină, care apare ca fereastră aparte de aplicație, des folosit pentru a confirma o acțiune
4. javafx.scene.layout.FlowPane – panou în care componentele copii se aranjează într-un flux fiind separate orizontal și vertical de o valoare specificată
5. javafx.scene.layout.GridPane – panou împărțit în rânduri și coloane, formând celule neuniforme în care sunt amplasate componenetele copii
6. javafx.scene.layout.HBox – panou în care componentele copii se aranjează orizontal unul după altul după un anumit spațiu specificat
7. javafx.scene.layout.VBox – panou în care componentele copii se aranjează vertical unul după altul după un anumit spațiu specificat
8. javafx.scene.layout.StackPane – panou în care componentele copii se vizualizează ordonate în formă de stivă, astfel ultima componentă adăugată este prima care se vede
9. javafx.scene.text.TextFlow – panou în care componentele textuale se aranjează într-un flux,
10. javafx.scene.layout.TilePane – panou împărțit în rânduri și coloane, formând celule uniforme în care sunt amplasate componenetele copii

Clasa abstractă javafx.scene.control.Control este clasa de bază pentru toate componentele cu care poate interacționa un utilizator. Aceasta conține 3 câmpuri noi:

1. „*contextMenu*” – un meniu care apare la apasărarea al 2 buton de la șoricel
2. „*skin*” – pentru a stiliza înfățișarea controalelor
3. „*tooltip*” – pentru a afișa o mică informație când cursorul trece deasupra componentei

și metodele respective pentru a seta și citi informația din aceste câmpuri.

De la această clasă moștenesc următoarele clase:

1. javafx.scene.control.Labeled – clasă abstractă de bază pentru toate componentele asociate cu un text care nu poate fi editat. Conține câmpuri și metode necesarea pentru formatarea textului
2. javafx.scene.control.Accordion –
3. javafx.scene.control.ButtonBar –
4. ChoiceBox
5. ComboBoxBase
6. HTMLEditor
7. ListView
8. MenuBar
9. Pagination
10. ProgressIndicator
11. ScrollBar
12. ScrollPane
13. Separator
14. Slider
15. Spinner
16. SplitPane
17. TableView
18. TabPane
19. TextInputControl
20. ToolBar
21. TreeTableView
22. TreeView

De la clasa javafx.scene.control.Labeled moștenesc următoarele clase:

1. javafx.scene.control.Label – zonă cu text care nu poate fi editat
2. javafx.scene.control.ButtonBase – clasă abstractă de bază pentru toate butoanele
3. javafx.scene.control.TitledPane – panou cu titlu care poate fi deschis/închis

De la clasa javafx.scene.control.ButtonBase moștenesc următoarele clase:

* + 1. javafx.scene.control.Button – reprezintă un simplu buton care poate fi apăsat. Conține 2 câmpuri de tip boolean care indică alte 2 moduri de interacțiune cu butonul: „*cancelButton*” și „*defaultButton*”. Butonul cu „*cancelButton*” setat pe true va reacționa la apăsararea butonului ESC de pe claviatură, cu condiția că nici un alt component nu reacționeză la această apăsare. La fel și butonul cu „*defaultButton*” setat pe true va reaționa la apăsarea butonului ENTER
    2. javafx.scene.control.CheckBox – reprezintă casetă pentru bifat și text corenspunzător. Conține 2 câmpuri de tip boolean: „*indeterminate*” și „*selected*” care generează una din cele 3 stări: selectat(false, true), neselectat(false,false) și nedefinit(true, –)
    3. javafx.scene.control.Hyperlink – reprezintă o adresă către o pagină Web. Conține un singur câmp de tip boolean „*visited*” ce înseamnă a fost vizitat adresa sau nu
    4. javafx.scene.control.MenuButton
    5. javafx.scene.control.ToggleButton

Bibliografie:

1. [https://docs.oracle.com/javase/8/docs/api/java/util/EventObject.html](https://docs.oracle.com/javase/8/docs/api/java/util/EventObject.html?is-external=true)
2. <https://docs.oracle.com/javase/8/javafx/api/javafx/event/Event.html>
3. <https://docs.oracle.com/javase/8/javafx/scene-graph-tutorial/scenegraph.htm>
4. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/Node.html>
5. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/shape/Shape.html>
6. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/shape/Shape3D.html>
7. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/Canvas.html>
8. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/image/ImageView.html>
9. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/media/MediaView.html>
10. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/Group.html>
11. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/Region.html>
12. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/chart/Axis.html>
13. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/chart/Chart.html>
14. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Control.html>
15. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/Pane.html>
16. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/AnchorPane.html>
17. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/BorderPane.html>
18. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/DialogPane.html>
19. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/FlowPane.html>
20. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html>
21. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/HBox.html>
22. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/VBox.html>
23. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/StackPane.html>
24. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/text/TextFlow.html>
25. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/TilePane.html>
26. <https://docs.oracle.com/javase/8/javafx/layout-tutorial/builtin_layouts.htm>
27. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Labeled.html>
28. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Label.html>
29. <https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/label.htm>
30. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/ButtonBase.html>
31. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Button.html>
32. <https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/button.htm>
33. <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/CheckBox.html>
34. <https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/checkbox.htm>