10.1. Tipovi događaja i slušači

10.2. Događaji akcija (ActionEvent - ActionListener)

10.3. Događaji fokusa (FocusEvent - FocusListener)

10.4. Događaji tastera (KeyEvent - KeyListener)

10.5. Događaji miša (MouseEvent - MouseListener)

10.6. Događaji prozora (WindowEvent - WindowListener)

10.7. Korišćenje dijaloga

10.1. Tipovi događaja i slušači

Kada obrađujemo događaje koji se dešavaju u radu sa Swing komponentama, radimo sa parom objekata: Slušač događaja (**EventListener**) - Tip događaja (**EventObject**). Dakle, radimo sa jednim interfejsom za registraciju osmatrača (slušaoca) i jednom klasom za tip događaja koji osmatramo. Za određenog slušača (**Listener**) događaja postoji tačno određen tip događaja (**Event**) sa kojim on radi u paru.

Sve klase koje predstavljaju tip događaja koji se desio nasleđuju klasu *java.util.EventObject,* koja sadrži metodu: *Object getSource(),* koja vraća komponentu koja je izazvala događaj.

Najteži deo rada za obradu događaja za koponente jeste prepoznati koji slušač događaja (naslednik interfejsa EventListener) je potrebno da se poveže sa komponentom da bismo osmatrali ponašanje koje nas zanima. U Javi postoji preko 20 različitih parova slušač - događaj koji se koriste u zavisnosti od vrste Swing komponente koja se želi posmatrati. Nama će biti interesantno svega 5 parova ovih objekata.

10.2. Događaji akcija (ActionEvent - ActionListener)

Kombinacija ActionListener-ActionEvent je najčešće korišćen par za obradu događaja. Komponente koje proizvode događaj ActionEvent su sledeće:

Tip događaja	Komponente
ActionEvent	**Dugmad:** JButton, JToogleButton, JCheckBox (pritisak tasterom miša, pritisak tasterom tastature)
	Meniji: JMenuItem, JMenu, JCheckBoxMenuItem, JradioButtonMenuItem (pritisak tasterom miša, pritisak tasterom tastature)
	Tekstualna polja: JTextField (pritisak tastera ENTER u polju)

Klasa koja predstavlja slušaoca koji će obrađivati događaje akcija mora da implementira interfejs **ActionListener.** Interfejs ActionListener ima samo jednu metodu: ***void actionPerformed(ActionEvent e)***. Ova metoda se poziva svaki put kada se desi događaj akcije i njoj se prosleđuje generisani objekat klase **ActionEvent**.

Klasa ActionEvent sadrži sledeće metode:

Metoda	Opis
String getActionCommand()	Vraća komandni string. Predstavlja deo informacije koji je specifičan za komponentu. Za dugme on vraća tekst natpisa na dugmetu, a za polje za unos teksta on vraća sadržaj unet u polje. Takođe ova vrednost se može postaviti na nivou komponente
int getModifiers()	Vraća modifikator tastere koji su pritisnuti. Pomoću ove metode možemo odrediti kada su pritisnuti tasteri ALT, SHIFT i CTRL. Ukoliko je više ovih tastera pritisnuto kada se događaj desio oni će biti kombinovani u ceo broj pa je potrebno koristiti operator & da bi se videlo koji je modifikator pritisnut
long getWhen()	Vraća vreme događaja
Object getSource()	Vraća objekat koji je izazvao događaj

Importovani projekat Termin10i11 u Eclipse. Pokrenuti aplikaciju i razmotriti paketsku strukturu projekta. U Eclipsu obavezno treba da bude vidljiv prozor „Console" u koji će se ispisivati poruke iz aplikacije (Window -> Show view->Console). Komponente koje izazivaju ActionEvent je stavka menija File->New (probati i klikom miša i prečicom tastature), stavka toolbara i dva tekstualna polja. Na stavku menija, dugme toolbara i prvo testualno polje je vezan jedan listener (MyActionListener1) dok je na drugo tekstualno polje vezan drugi listener (MyActionListener2). Pogledati i prokomentarisati klase Menu, Toolbar, ActionListenerPanel, MyActionListener1, MyActionListener2. Pronaći mesto u klasama gde se za navedene 4 komponete registruju slušaoci.

10.3. Događaji fokusa (FocusEvent - FocusListener)

Sve Swing komponente imaju događaje fokusa. On nastupa kada komponenta dobije ili izgubi ulazni fokus.

Klasa koja predstavlja slušaoca koji će obrađivati događaje fokusa mora da implementira interfejs **FocusListener.** Interfejs FocusListener ima po jednu metodu za svaki od dva tipa događaja:

focusGained(FocusEvent e) - poziva se kada komponenta dobije fokus.

focusLost(FocusEvent e) - poziva se kada komponenta izgubi fokus.

U ove dve metode se prosleđuje objekat koji opisuje događaj koji se desio (dobijanje ili gubitak fokusa) i to je objekat klase **FocusEvent**.

Klasa FocusEvent sadrži sledeće metode:

Metoda	Opis
boolean isTemporary()	Vraća da li je promena fokusa privremena ili trajna. Na primer privremeni gubitak glavnog prozora se dešava kada popup menu privremeno dobije fokus.
Component getComponent()	Vraća komponentu koja je izazvala događaj fokusa
Component getOppositeComponent()	Vraća sekundarnu komponentu koja je uključena u događaj. Tako na primer ako se desio FOCUS GAINED ova metoda vraća komponentu koja je izgubila pre toga fokus. I obratno, ako se desio FOCUS LOST, metoda vraća komponentu koja je dobila fokus.

Pokrenuti projekat Termin10i11. U delu koji izaziva Focus Event nalaze se 3 tekstualna polja. Polje „prezime" je obavezno za unos i dok se ne unese neki tekst nije moguće napustiti polje. Ostala polja promenom boje pozadine prate promenu fokusa. Promene fokusa u sva 3 polja osluškuje ista klasa: MyFocusListener.

10.4. Događaji tastera (KeyEvent - KeyListener)

Događaji tastera se generišu kada se pritisne ili otpusti bilo koji taster sa tastature. Klasa koja želi da obrađuje događaje tastera mora da implementira interfejs **KeyListener**. Ovaj interfejs poseduje ukupno 3 metode (jednu metodu za pritisak tastera, jednu za otpuštanje tastera i jednu za ukucan taster):

keyTyped(KeyEvent) - poziva se za ukucan taster
keyPressed(KeyEvent) - poziva se za pritisak tastera
keyReleased(KeyEvent) - poziva se za otpuštanje tastera.

Ako želimo da ukucamo veliko slovo A tok događaja je sledeći:

1. Pritiskamo dugme SHIFT
2. Ovo okida poziv za metodu keyPressed()
3. Pritiskamo taster A
4. Ovo okida poziv metode keyPressed()
5. Takođe okida poziv metode keyTyped
6. Otpuštanje tastera poziva metodu keyReleased() i za taster Shift i za taster A

Svaki događaj tastera generiše objekat klase **KeyEvent** koji se prosleđuje u navedene 3 metode.

Klasa KeyEvent sadrži sledeće metode:

Metoda	Opis
int getKeyChar()	Vraća znak tastera
int getKeyCode()	Vraćakod tastera. Na primer, kod za taster A je VK_A a za taster shift VK_SHIFT
boolean isActionKey()	Provera da li je taster - taster akcije. U ove tastere spdaju između ostalih: CAPS LOCK taster, strelice za pomeranje na tastaturi, funkcionalni tasteri

Pokrenuti aplikaciju i probati unos JMBG-a. Neipravan unos (karakter koji nije broj i dužina veća od 13) prouzrokovaće poruku o grešci. Pogledati klase KeyListenerPanel, MyKeyListener.

10.5. Događaji miša (MouseEvent - MouseListener)

Događaji miša su dostupni svim podklasama klase Component. Postoje dva slušača koji rade sa ovim događajem. To su **MouseListener** za osnovne događaje u vezi sa mišom i **MouseMotionListener** za događaje koji se odnose na kretanje miša. Ova dva slušača su razdvojena zbog performansi jer se događaji koji se odnose na pomeranje miša dešavaju često.

Slušač MouseListener ima pet metoda:

mouseClicked(MouseEvent)
mouseEntered(MouseEvent)
mouseExited(MouseEvent)
mousePressed(MouseEvent)
mouseReleased(MouseEvent)

Slušač MouseMotionListener ima samo dve metode:

mouseDragged(MouseEvent) - poziva se kada korisnik pomera miša i u isto vreme neki od tastera miša pritisnut
mouseMoved(MouseEvent) - poziva se kada korisnik pomera miša bez pritisnutih tastera

Oba slušača rade sa istom klasom **MouseEvent**. Klasa MouseEvent sadrži sledeće metode:

Metoda	Opis
int getClickCount()	Vraća broj povezanih klikova mišem
int getX() int getY() Point getPoint()	Vraća poziciju u okviru izvora događaja
int getXOnScreen() int getYOnScreen() int getLocationOnScreen()	Vraća apsolutnu poziciju događaja na ekranu
boolean isPopupTrigger()	Proverava da li će događaj miša okidati iskačući meni ukoliko je dostupan
int getButton()	Vraća indikator koji ji taster na mišu pritisnut: 1- levi taster 2- točkić miša 3- desni taster

Događaji koji se odnose na kretanje miša i obrađuju se sa interfejsom MouseMotionListener neće biti obrađivani.

Pokrenuti projekat i pogledati klase MouseListenerPanel, MyMouseListener1 i MyMouseListener2.

10.6. Događaji prozora

Događaji prozora su specifični za java.awt.Window i sve njene podklase uključujući i JWindow, JFrame, JDialog, Dialog i FileDialog. Događaji prozora se šalju kada se desi neka vrsta promene u prozoru. Tri slušača su povezana sa događajima prozora:

a) WindowListener

void windowOpened(WindowEvent e)
void windowClosing(WindowEvent e)
void windowClosed(WindowEvent e)
void windowIconified(WindowEvent e)
void windowDeiconified(WindowEvent e)
void windowActivated(WindowEvent e)
void windowDeactivated(WindowEvent e)

b) WindowFocusListener

void windowGainedFocus(WindowEvent e)
void windowLostFocus(WindowEvent e)

c) WindowStateListener

void windowStateChanged(WindowEvent e)

Klasa WindowEvent sadrži sledeće metode:

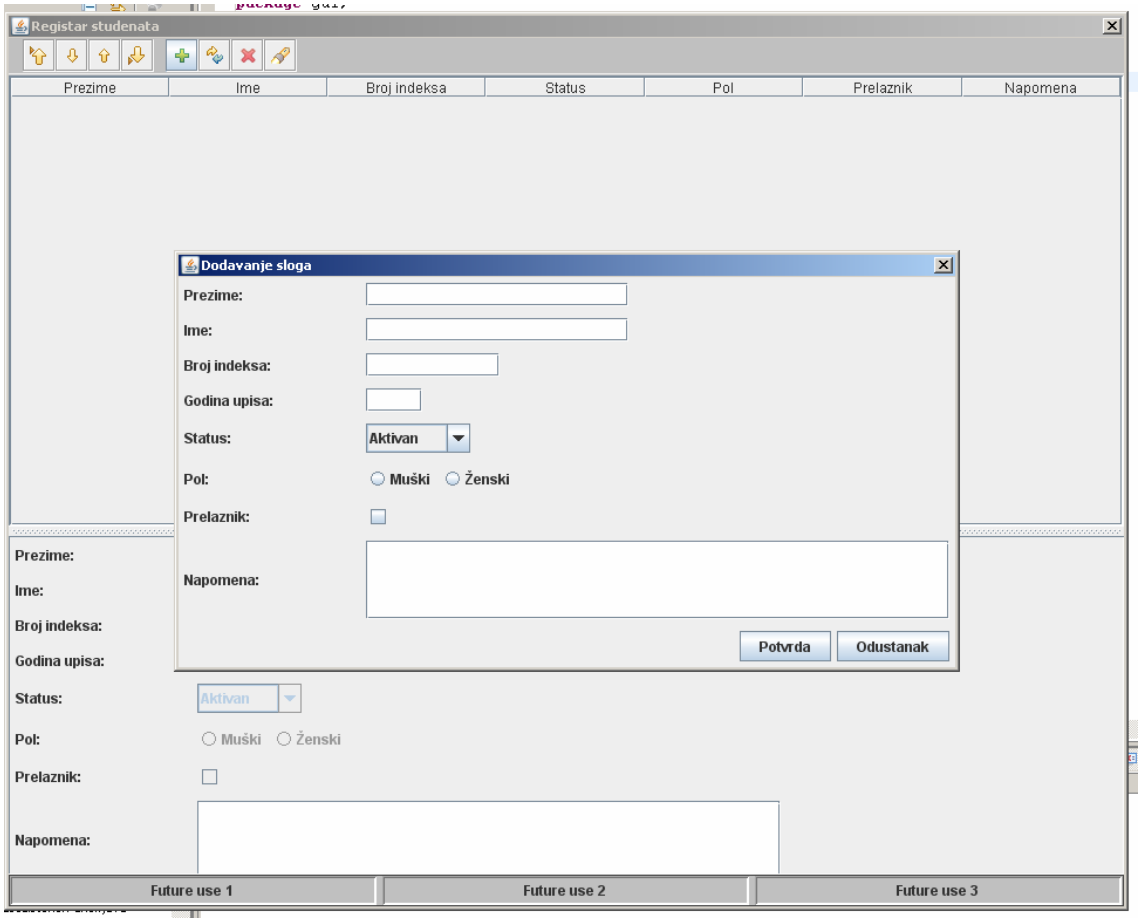
Metoda	Opis
Window getWindow()	Vraća prozor koji je inicirao događaj
Window getOppositeWindow()	Vraća sekundarni prozor koji je uključen u događaj
int getNewState()	Vraća novo stanje kod promene stanja: NORMAL ICONIFIED MAXIMIZED_HORIZ MAXIMIZED_VERT MAXIMIZED_BOTH
int getOldState()	Vraća staro stanje kod promene stanja: NORMAL ICONIFIED MAXIMIZED_HORIZ MAXIMIZED_VERT MAXIMIZED_BOTH

Događaji koji se obrađuju sa interfejsom WindowFocusListener i WindowStateListener neće biti obrađivani.

U projektu se kao primer korišćenja događaja prozora koristi interfejs WindowListener. Ovaj interfejs implementira klasa MyWindowListener koja se poziva prilikom rada sa glavnim prozorom aplikacije (videti klasu MainFrame). Prilikom zatvaranja glavnog prozora poziva se metoda windowClosing u kojoj je moguće otkazati zatvaranje prozora.

10.7. Korišćenje dijaloga

Dijalog je prozor koji se prikazuje u kontekstu drugog prozora - njegovog roditelja. Dijalozi se koriste za upravljanje ulaznim podacima, na primer: biranje iz nekog opsega opcija ili omogućavanje da se podaci unesu preko tastature. Dijalozi se takođe mogu koristiti za poruke sa informacijama ili upozorenjima.



Dijalozi su definisani u klasi JDialog u paketu javax.swing koja predstavlja naslednika klase Window. Objekat JDialog obično sadrži jednu ili više komponenti za prikazivanje informacija ili za omogućavanje unosa informacija. Takođe postoji i prosti, „instant" dijalozi koji se dobijaju korišćenjem klase JOptionPane.

Postoje 2 vrste dijaloga koji mogu da se prave: **modalni** i **nemodalni** dijalozi.

Kada se prikaže **modalni dijalog** - obično izborom neke stavke u meniju ili pritiskom na dugme - on sprečava rad svakog drugog prozora u aplikaciji dok se ovaj dijalog ne zatvori. Prozor dijaloga zadržava fokus sve dok se on prikazuje, a rad aplikacije se ne može nastaviti dok se dijalog ne zatvori (obično pritiskom na dugme ok ili cancel). Modalni dijalozi koji upravljaju ulazom obično imaju najmanje 2 dugmeta - dugme OK koje se koristi za prihvatanje ulaznih podataka i zatvaranje dijaloga i dugme CANCEL koje služi samo za zatvaranje dijaloga. Dijalozi koji upravljaju ulazom su gotovo uvek modalni dijalozi iz prostog razloga što se uglavnom ne dozvoljava pokretanje drugih interakcija dok se ne kompletira unos korisničkih podataka.

Nemodalni dijalog može da ostane na ekranu proizvoljno dugo pošto on ne blokira interakciju sa drugim prozorima pa se fokus može prebacivati iz nemodalnog dijaloga na druge aplikacione prozore koji se nalaze na ekranu.

Vrsta dijaloga se određuje prilikom pravljenja dijaloga, prosleđivanjem željene vrednosti u konstruktor klase JDialog:

Konstruktor	Opis
JDialog(Frame parent)	Pravi se nemodalni dijalog, koji ima prazan naslov, dok je **parent** instanca klase JFrame koja će biti roditelj dijaloga
JDialog(Frame parent, String title)	Pravi se nemodalni dijalog, koji kao naslov ima **title** parametar, dok je **parent** instanca klase JFrame koja će biti roditelj dijaloga
JDialog(Frame parent, boolean modal)	Pravi se dijalog koji će u zavisnosti od vrednosti parametra **modal** biti modalni/nemodalni, koji ima prazan naslov, dok je **parent** instanca klase JFrame koja će biti roditelj dijaloga
JDialog(Frame parent, String title, boolean modal)	Pravi se dijalog koji će u zavisnosti od vrednosti parametra **modal** biti modalni/nemodalni, koji kao naslov ima **title** parametar, dok je **parent** instanca klase JFrame koja će biti roditelj dijaloga

Svi objekti klase JDialog su inicijalno nevidljivi (isto kao i JFrame), tako da se za objekat klase JDialog mora pozvati metoda setVisible() sa argumentom true da bi se prikazao.

Pokrenuti aplikaciju i kao primer kreiranja i prikaza modalnog i nemodalnog dijaloga pogledati klasu Menu i klasu SimpleDialog(prve dve stavke menija Dijalozi).

Zatvaranje modalnog dijaloga obično se obavlja pozivom metode setVisible() sa argumentom false. Metoda setVisible() utiče samo na vidljivost dijaloga. Objekat klase JDialog koji nije vidljiv i dalje je kreiran i dostupan (**pogledati primer u aplikaciji i klasu HideDialog kao i njen poziv iz klase Menu, pogledati na koji način se pojavljuje poruka samo ukoliko je dijalog zatvoren pritiskom na dugme OK**).

Klasa JOptionPane iz paketa javax.swing definiše niz statičkih metoda koje će automatski napraviti i prikazivati standardne modalne dijaloge. Nama će biti interesantne sledeće dve:

Metoda	Opis
JOptionPane.showMessageDialog(Component parent, Object message, String title, int messageType)	Prikazuje se modalni dijalog sa prosleđenim naslovom u parametru **title**. Ako je prosleđeni parametar **parent** jednak null tada će se kreirani dijalog postaviti na centar ekrana, a ukoliko je kao **parent** prosleđen JFrame objekat tada će kreirani dijalog biti centriran u odnosu na prosleđeni objekat. Parametar **message** pokazuje šta treba da se prikaže u dijalogu pored dugmeta OK. Najčešće je String ali može se proslediti i čitava komponenta u dijalog. Stil dijaloga je određen četvrtim parametrom, **messageType** koji predstavlja jednu od konstanti: JOptionPane.ERROR_MESSAGE, JOptionPane.INFORMATION_MESSAGE, JOptionPane.WARNING_MESSAGE, JOptionPane.QUESTION_MESSAGE, JOptionPane.PLAIN_MESSAGE, U zavisnosti od ovog parametra biće određen stil prikazivanja dijaloga.
int JOptionPane.showConfirmDialog(Component parent, Object confirmMessage, String title, int confirmType)	Pravi i prikazuje modalni dijalog sa ponuđenim odgovorima na pitanje. Broj i vrsta dugmića zavisi od vrednosti parametra **confirmType** JOptionPane.DEFAULT_OPTION, JOptionPane.YES_NO_OPTION, JOptionPane.YES_NO_CANCEL_OPTION JOptionPane.OK_CANCEL_OPTION Kao rezultat pojavljivanja dijaloga vraća se celobrojna vrednost: YES - vrednost 0 NO - vrednost 1 CANCEL - vrednost 2

(**pogledati primer u aplikaciji i klasu MyWindowListener u kojoj se inicira prikaz ConfirmDialog-a**).