WEB Programiranje

[C l a s s T o o l k i t]

Davor Menon
davor.menon@gmail.com

20. veljače 2006.

1 Općenito o klasi

```
public abstract class Toolkit extends Object
```

Klasa `Toolkit` je apstraktna klasa, odnosno nju nećemo koristiti za instanciranje objekata. Klasu `Toolkit` se koristi samo za kreiranje podklasa.

Nadklasa klase `Toolkit` je klasa `Object`. Klasa `Toolkit` je dio paketa `java.awt`.

Klasa `Toolkit` je apstraktna nadklasa svih stvarnih implementacija od "Abstract Window Toolkit". Podklase od klase `Toolkit` se koriste da prilagode različite komponente dotičnoj "domaćoj" toolkit implementaciji.

Mnoge GUI (*eng.* ***G****raphical* ***U****ser* ***I****nterface* - ***G****rafičko* ***K****orisničko* ***S****učelje*) radnje mogu biti izvedene neujednačeno. Ovo znači da ako postavite neko svojstvo određene komponente, i odmah se raspitate za njeno stanje, povratna vrijednost će pokazati da zatražena promijena stanja još nije izvršena. Ovo uključuje sljedeće radnje, ali nije ograničeno samo na njih:

- **"Skrolanje" na određenu poziciju**
  Na primjer, pozivanje

  ```
  ScrollPane.setScrollPosition
  ```

  i tada

  ```
  ScrollPane.getScrollPosition
  ```

  može vratiti neispravnu vrijednost ako originalan zahtjev nije još procesuiran.

- **Prebacivanje fokusa s jedne komponente na drugu**
  Za detaljnije obješnjanje pogledati poglavlje "Timing Focus Transfers" u "The Swing Tutorijal". Dovoljno je reći da prebacivanje fokusa s jedne komponente na drugu zahtjeva nešto vremena, a dok se prebacivanje obavlja, zaprimaju i procesuiraju se ostale instrukcije koje je programer zadao.

- **Postavljanje vršnog spremnika (Container) vidljivim**
  Pozivanje `setVisible(true)` za `Window`, `Frame` ili `Dialog` se može izvršiti neujednačeno.

- **Postavljanje dimenzija ili položaja vršnog spremnika**
  Pozivanje `setSize()`, `setBounds()` ili `setLocation()` za `Window`, `Frame` ili `Dialog` se prosljeđuje sustavu za upravljanje prozorima i može biti ignorirano ili promijenjeno.

Većina aplikacija ne bi trebala zvati niti jednu metodu iz ove klase direktno. Metode definirane pomoću `Toolkit`-a spajaju klase koje ne ovise o platformi ("*platform-independent classes*") s njihovim odgovarajućim objektima u `java.awt.peer`. Neke metode definirane klasom `Toolkit` detektiraju postavke lokalnog operacijskog sustava direktno.

Klasa `Toolkit` je u sastavu Jave od **JDK 1.0**

2 Detaljnije o elementima klase `Toolkit`

2.1 Detalji o poljima

Skoro svaka klasa sadrži uz metode i nekakve podatke. Klasa `Toolkit` ima definirana dva polja podataka.

desktopProperties

```
protected final Map<String,Object> desktopProperties
```

desktopPropsSupport

```
protected final PropertyChangeSupport desktopPropsSupport
```

2.2 Detalji o konstruktoru

Iako smo rekli da je klasa `Toolkit` apstraktna i da će nam služiti samo za kreiranje njenih podklasa, konstruktor je ipak definiran.

Toolkit

```
public Toolkit()
```

2.3 Detalji o nekim metodama

Klasa `Toolkit` sadrži mnogo metoda. Ovdje su navedene samo neke metode. Za svaku metodu na navedena klasa objekata koje uzima i klasa objekta koje vraća.

createButton

```
protected abstract java.awt.peer.ButtonPeer
createButton(Button target) throws HeadlessException
```

Kreira implementaciju `Button`-a ovog toolkit-a koristeći navedeno peer sučelje.

Parametri:
`target` – gumb koji će biti implementiran

Vraća:
implementaciju ovog toolkita za `Button`.

createTextField

```
protected abstract java.awt.peer.TextFieldPeer
createTextField(TextField target)throws HeadlessException
```

Kreira implementaciju `TextField`-a ovog toolkit-a koristeći navedeno peer sučelje.

Parametri:
`target` – tekstualno područje koje će biti implementirano

Vraća:
implementaciju ovog toolkita za `TextField`.

createLabel

```
protected abstract java.awt.peer.LabelPeer
createLabel(Label target)  throws HeadlessException
```

Kreira implementaciju `Label`-a ovog toolkit-a koristeći navedeno peer sučelje.

Parametri:
`target` – "naljepnica" koja će biti implementirana

Vraća:
implementaciju ovog toolkita za `Label`.

Navodi se još niz gotovo identičnih metoda poput: **createList**, **createCheckbox**, **createScrollbar**, **createScrollPane**, **createTextArea**, **createChoice**, **createFrame**, **createCanvas**, **createPanel**, **createWindow**, **createDialog**, **createMenuBar**, **createMenu**, **createPopupMenu**, **createMenuItem**, **createFileDialog**, **createCheckboxMenuItem**.

createComponent

```
protected java.awt.peer.LightweightPeer
    createComponent(Component target)
```

Kreira peer za komponentu ili spremnik. Ovaj peer je bez prozora i omogučuje da `Component` i `Container` klase budu direktno proširene da bi se kreirale komponente bez prozora definirane u potpunosti u javi.

Parametri:

`target` – Komponenta koju treba napraviti.

loadSystemColors

```
protected void loadSystemColors(int[] systemColors)
                         throws HeadlessException
```

Popunjava niz cijelih brojeva koji je prosljeđen kao argument s tekućim vrijednostima sistemskih boja.

Parametri:

`systemColors` – jelobrojni niz.

setDynamicLayout

```
public void setDynamicLayout(boolean dynamic)
                      throws HeadlessException
```

Određuje hoće li se raspored spremnika potvrđivati dinamički tijekom razvlačenja prozora, ili statički, nakon što je s razvlačenjem gotovo. Treba znati da ovo svojstvo nije podržano na svim platformama. Na platformama gdje dinamačko raspoređivanje tijekom rastezanja nije podržano (ili gdje je uvijek podržano) postavke ovog svojstva nemaju učinka. Također treba znati da se ovo svojstvo na pojedinim platformama može uključiti ili isključiti kao postavka samog operacijskog sustava. Na takvim platformama, ovo svojstvo mora biti postavljeno u operacijskom sustavu ili na razini upravitelja prozorima prije nego ova metoda može imati učinka. Ova metoda ne mijenja pozadinski operativni sustav ili upravitelj prozora ili postavke. OS/WM podrška može biti ispitana koristeći `getDesktopProperty("awt.dynamicLayoutSupported")`.

Parametri:

`dynamic` – Ako je istinito, spremnici bi trebali nanovo raspoređivati svoje komponente kako se Spremnik rasteže. Ako je laž, raspored će biti potvrđen nakon što rastezanje završi.

isDynamicLayoutSet

```
protected boolean isDynamicLayoutSet()
              throws HeadlessException
```

Govori nam potvrđuje li se raspored spremnika dinamički tijekom rastezanja ili statički nakon što je rastezanje završeno. treba znati da ova metoda vraća postavke koje su postavljene od strane programa i to se ne mora slagati s podrškom koju daje OS/WM. OS/WM podrška se može ispitati koristeći `getDesktopProperty("awt.dynamicLayoutSupported")`.

Vraća:
Istinu ako se potvrđivanje spremnika radi dinamički, laž u suprotnom.

isDynamicLayoutActive

```
public boolean isDynamicLayoutActive()
        throws HeadlessException
```

Govori nam koristi li se trenutno podrška za dinamičko raspoređivanje spremnika pri rastezanju (oba slučaja – postavljeno od strane programa i podržano od strane OS/WM-a)

Vraća:
Istinu ako se podrška za dinamičko raspoređivanje spremnika trenutno koristi, u suprotnom vraća laž.

getScreenSize

```
public abstract Dimension getScreenSize()
                 throws HeadlessException
```

Daje veličinu zaslona. Na sustavima s višestrukim zaslonima se koristi primarni zaslon. Mnogo–zaslonska podrška se može dobiti s `GraphicsConfiguration` i `GraphicsDevice`.

Vraća:
Veličinu zaslona, u točkicama.

getScreenInsets

```
public Insets getScreenInsets(GraphicsConfiguration gc)
                      throws HeadlessException
```

Daje veličinu područja uz rub zaslona na koji se ne piše.

Parametri:
`gc` – objekt klase `GraphicsConfiguration`

Vraća:
rub zaslona po kojem se ne piše, u točkicama.

getScreenResolution

```
public abstract int getScreenResolution()
                  throws HeadlessException
```

Daje rezoluciju zaslona tockica-po-inču.

Vraća:
Rezoluciju zaslona, u točkicama-po-inču.

getImage

```
public abstract Image getImage(URL url)
```

Vraća sliku koji uzima podatke o pikselima iz podataka određenih URL-om. Podaci o pikselima ukazani URL-om moraju biti u jedno od sljedećih formata: GIF, JPEG ili PNG. Programeri se potiču da razvijaju vlastite načine "čuvanja određene slike u pripravnosti u slučaju da zatreba" koristeći `createImage` varijantu kada je god to moguće.

Parametri:
`url` – URL pomoću kojeg se hvataju piksel podaci.

Vraća:
Sliku koja je stvorena na osnovi piksel podataka.

createImage

```
public abstract Image getImage(URL url)
```

Vraća sliku koja je nastala na osnovi podataka o pikselima dobivenih navedemin URL-om. Dobivena slika je objekt koji neće biti djeljen s nijednim drugim pozivačem ove metode ili njene `getImage` inaćice.

Parametri:
`url` – URL pomoću kojeg se hvataju piksel podaci.

Vraća:
Sliku koja je stvorena na osnovi piksel podataka.

prepareImage

```
public abstract boolean prepareImage(Image image,
                                     int width,
                                     int height,
                                     ImageObserver observer)
```

Priprema sliku za upotrebu.

Može se odrediti i visina i širina za koju se slika treba pripremiti.
Podaci o slici su odvojeno pribavljeni i odgovarajuće skalirana reprezentacija slike na zaslonu je generarirana.

Parametri:

`image` – Slika za koju se priprema reprezentacija.
`width` – Širina prikaza na zaslonu.
`height` – Visina prikaza na zaslonu.
`observer` – objekt `ImageObserver` koji će biti obavještavan o pripremi slike.

Vraća:

Istinu ako je slika već u potpunosti pripremljena, laž u suprotnom.

getSystemClipboard

```
public abstract Clipboard getSystemClipboard()
                          throws HeadlessException
```

Daje objekt Clipboard koji poprima sadržaj s clipboard komponentama platforme na kojima se izodi program. Ovaj clipboard omogućuje razmjenu podataka između java programa i "prirodnih" aplikacija koji koriste "prirodne" komponente clipboarda.
Svaka implementacija ove metode bi trebala prvo provjeriti kod postoji li instalacija sigurnosnog upravitelja. Ako postoji, metoda treba prvo pozvati sigurnosnog upravitelja i upitati smije li pristupiti sadržaju clipboarda.

beep

```
public abstract void beep()
```

Emitira zvučni signal.