Update data

De Wiki
Aller à : navigation, rechercher
Update data

Sometimes, it is needed to update the content of a widget depending on the modification done on another one. If both of them are included in a higher level widget, it can be done relatively easily. But in case of these widgets are displayed in different panels (for example inside a GTabbedPane), it may be difficult to manage it.

A first solution could be to use static objects but it is strongly not recommanded.

Another solution is avalaible since the 1.5 version. We will describe hereafter.

Imagine that :

1. we have two widgets, WidData1 and WidData2, including some specific data information
2. we have a widget, WidScenario that defines a scenario thanks to some data included in previous widgets.

GObservable interface

First, WidData1 and WidData2 will have to implement the GObservable interface because these widgets will be the ones that will be first updated and thus have to be observed (they are "observable" !) to know if something has to be done.

Due to the GObservable interface, the abstract registerObserver, unregisterObserver and notifyObservers methods will have to be implemented inside the WidData1 and WidData2 class:

```
GObserverList obsList;

...

@Override
public void registerObserver(GObserver observer) {
    obsList.registerObserver(observer);
}

@Override
public void unregisterObserver(GObserver observer) {
    obsList.unregisterObserver(observer)
}

@Override
public void notifyObservers(Object... args) {
    obsList.notifyObservers(this, args);
}
```

obsList is a GObserverList object only used to store a list of "observers" widgets and proposing:

- a registerObserver method to store objects that will observe our widget
- an unregisterObserver method
- a notifyObservers method doing a loop on all "observers" components in order to call for their notify method.

*Note that we have to use this kind of object because in **Java** (below 1.8 version), a method in an interface must be abstract. So, we use it to avoid not to repeat same code.*

Moreover, we have to call for the notifyObservers method of this object as needed. In the example just below, we will place it in the after and read methods:

```
@Override
public void after(GEvent arg0) throws GException {

    // We may refine the test case by searching the specific subwidgets
    // which have been changed by using:
    // arg0.getLocalSource() or arg0.getFinalSource()
    notifyObservers();

}

@Override
public void read(GEvent arg0) throws GException {

    generic();
    notifyObservers();

}
```

GObserver interface

Secondly, WidScenario will have to implement the GObserver interface as it will observe information coming from other widget(s) and, due to this information, an update will occur (or not).

Thanks to the GObserver interface, WidScenario should implement an notify method as is:

```
@Override
public void notify(Object observable, Object... args) {

    // "args" objects list will not be used in this example

    if ( observable instanceof WidData1 ) {
        // Update of the scenario thanks to WidData1 widget modification
        …
    } else if ( observable instanceof WidData2 ) {
        // Update of the scenario thanks to WidData1 widget modification
        …
    }
```

```
}
```

We see that the way to manage which object sent information is done by using instanceof syntax.

Links between updated and updatable objects

At last, in the main code we will have just to write something like this:

```
// Updated objects
widData1 = new WidData1(...);
widData2 = new WidData2(...);

// Updatable object
widScenario = new WidScenario(...);

// widScenario will be potentially updated if widData1 is changed
widData1.registerObserver(widScenario);
// widScenario will be potentially updated if widData2 is changed
widData2.registerObserver(widScenario);
```

Return to the introduction ↑ Go to the next page →

Menu de navigation

Outils personnels

- 3.133.12.172
- Discussion avec cette adresse IP
- Créer un compte
- Se connecter

Espaces de noms

- Page
- Discussion

Variantes

Affichages

- Lire
- Voir le texte source
- Historique
- Exporter en PDF

Plus

Rechercher

Rechercher Lire

GENIUS

- Welcome
- Quick Start
- News

Basic principles

- GFrame and GPanel
- Main widgets
- Links with Swing
- GLayout
- Conditional Display
- GListener interface

More deeper in the concept

- Units management
- GContainer
- GReadWrite interface and data files management
- Modified data
- Process management

Still more ...

- Validity controls
- Menu bar
- Icons
- GClear interface

Still more again ...

- Tooltips
- Shortcuts
- Copy & paste
- Plots

- Results File Management
- GPlotPanel
- GGroundPlotPanel
- Internationalization
- Log file
- Update data

Some other widgets

- GTabbedPane
- GTable1D
- GTable2D
- GComponentList
- GDialog and GDetachedPanel
- GContextFileManagement
- How to build a standard application
- GPanTest
- Create your own widget

Evolutions

- Main differences between V1.11.4 and V1.12.1
- Main differences between V1.10.1 and V1.11.4
- Main differences between V1.10 and V1.10.1
- Main differences between V1.9.1 and V1.10
- Main differences between V1.9 and V1.9.1
- Main differences between V1.8 and V1.9
- Main differences between V1.7 and V1.8
- Main differences between V1.6.2 and V1.7
- Main differences between V1.6.1 and V1.6.2
- Main differences between V1.6 and V1.6.1
- Main differences between V1.5 and V1.6
- Main differences between V1.4.1 and V1.5
- Main differences between V1.3 and V1.4.1

Training

- Training slides
- Tutorials package for V1.12.1
- Tutorials package for V1.11.4
- Tutorials package for V1.10.1
- Tutorials package for V1.9.1
- Training & tutorials package for V1.8
- Training & tutorials package for V1.7
- Training & tutorials package for V1.6

Links

- CNES freeware server

Outils

- Pages liées
- Suivi des pages liées
- Pages spéciales
- Adresse de cette version
- Information sur la page
- Citer cette page

- Dernière modification de cette page le 8 novembre 2017 à 13:54.

- Politique de confidentialité
- À propos de Wiki
- Avertissements
- Powered By MediaWiki