

Mediator 패턴

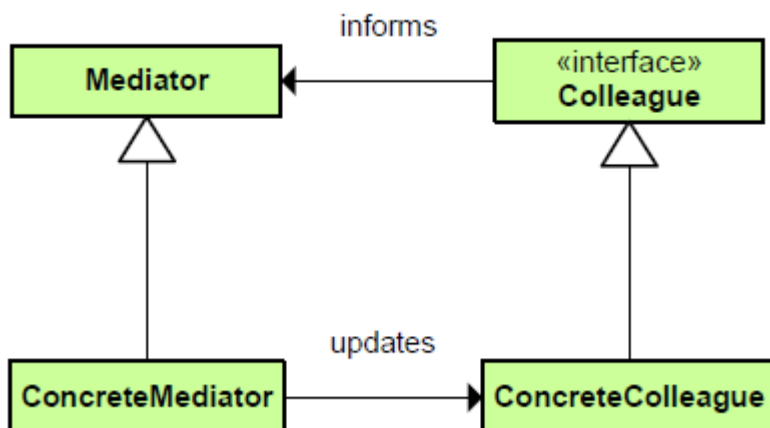
타입

Behavioral Pattern

문제

해결

클래스 다이어그램



예제

[mediator.cpp](#)

```
#include <iostream>

using namespace std;

class FileSelectionDialog;

class Widget
{
public:
    Widget(FileSelectionDialog *mediator, char *name)
    {
        _mediator = mediator;
        strcpy(_name, name);
    }
};
```

```
    }
    virtual void changed();
    virtual void updateWidget() = 0;
    virtual void queryWidget() = 0;
protected:
    char _name[20];
private:
    FileSelectionDialog *_mediator;
};

class List: public Widget
{
public:
    List(FileSelectionDialog *dir, char *name): Widget(dir, name){}
    void queryWidget()
    {
        cout << "    " << _name << " list queried" << endl;
    }
    void updateWidget()
    {
        cout << "    " << _name << " list updated" << endl;
    }
};

class Edit: public Widget
{
public:
    Edit(FileSelectionDialog *dir, char *name): Widget(dir, name){}
    void queryWidget()
    {
        cout << "    " << _name << " edit queried" << endl;
    }
    void updateWidget()
    {
        cout << "    " << _name << " edit updated" << endl;
    }
};

class FileSelectionDialog
{
public:
    enum Widgets
    {
        FilterEdit, DirList, FileList, SelectionEdit
    };
    FileSelectionDialog()
    {
        _components[FilterEdit] = new Edit(this, "filter");
        _components[DirList] = new List(this, "dir");
        _components[FileList] = new List(this, "file");
    }
};
```

```

        _components[SelectionEdit] = new Edit(this, "selection");
    }
    virtual ~FileSelectionDialog();
    void handleEvent(int which)
    {
        _components[which]->changed();
    }
    virtual void widgetChanged(Widget *theChangedWidget)
    {
        if (theChangedWidget == _components[FilterEdit])
        {
            _components[FilterEdit]->queryWidget();
            _components[DirList]->updateWidget();
            _components[FileList]->updateWidget();
            _components[SelectionEdit]->updateWidget();
        }
        else if (theChangedWidget == _components[DirList])
        {
            _components[DirList]->queryWidget();
            _components[FileList]->updateWidget();
            _components[FilterEdit]->updateWidget();
            _components[SelectionEdit]->updateWidget();
        }
        else if (theChangedWidget == _components[FileList])
        {
            _components[FileList]->queryWidget();
            _components[SelectionEdit]->updateWidget();
        }
        else if (theChangedWidget == _components[SelectionEdit])
        {
            _components[SelectionEdit]->queryWidget();
            cout << "    file opened" << endl;
        }
    }
}

private:
    Widget *_components[4];
};

FileSelectionDialog::~FileSelectionDialog()
{
    for (int i = 0; i < 3; i++)
        delete _components[i];
}

void Widget::changed()
{
    _mediator->widgetChanged(this);
}

int main()
{

```

```
FileSelectionDialog fileDialog;
int i;

cout << "Exit[0], Filter[1], Dir[2], File[3], Selection[4]: ";
cin >> i;

while (i)
{
    fileDialog.handleEvent(i - 1);
    cout << "Exit[0], Filter[1], Dir[2], File[3], Selection[4]: ";
    cin >> i;
}
```

참고

http://en.wikibooks.org/wiki/C%2B%2B_Programming/Code/Design_Patterns#Mediator

From:

<http://www.obg.co.kr/doku/> - **OBG Wiki**

Permanent link:

http://www.obg.co.kr/doku/doku.php?id=programming:design_pattern:mediator

Last update: **2020/11/29 14:09**

