
Язык программирования Java

Дмитриев Андрей Владиславович

andrei-dmitriev@yandex.ru

Май 2007

Строитель

- Отделяет конструирование объекта от его представления.
 - Позволяет использовать один и тот же процесс конструирования для создания различных представлений.
 - Пример:
графическое представление должно изменять свой вид в зависимости от текущего состояния приложения так, что если в списке меньше N элементов, то отображать `RadioButton`; иначе – выпадающий список
-

Строитель (абстрактный предок)

```
public abstract class MultiChoice {  
    //от этого класса наследуются список и кнопки  
    private Vector choices;//отображаемые сущности  
    public MultiChoice(Vector choiceList) {  
        choices = choiceList;  
    } //наследники реализуют следующие методы  
    abstract public Panel getUI();//создает панель  
        графических компонентов  
    abstract public String[] getSelected(); //текущее  
        выбранное значение(я)  
    abstract public void clearAll(); //сброс состояния  
}
```

Строитель (наследники, они же Строители)

```
class ListBoxChoice extends  
    MultiChoice {...}  
class CheckBoxChoice extends  
    MultiChoice {...}
```

Строитель (фабрика, он же директор)

```
public class ChoiceFactory {
    MultiChoice ui;
    //Возвращает графический контейнер с набором компонент
    public MultiChoice getChoiceUI(Vector choices) {
        if (choices.size() <=3){
            //панель с переключателями
            ui = new CheckBoxChoice(choices);
        } else {
            //панель со списком
            ui = new ListBoxChoice(choices);
        }
        return ui;
    }
}
```

Строитель (использование)

```
Vector v = null;
int index = stockList.getSelectedIndex();
choicePanel.removeAll(); //удалить старую панель
//выбор из нескольких вариантов представлений
switch (index) {
    case 0:
        v = Stocks; break;
    case 1:
        v = Bonds; break;
}
mchoice = cfact.getChoiceUI(v); //создание
    представления
choicePanel.add(mchoice.getUI()); //отрисовка
```

Строитель (выводы)

- Позволяет видоизменять внутреннее представление класса, который он создает.
 - Менее зависим от других частей программы.
 - Улучшенный контроль над созданием объектов.
-