

Національний технічний університет України «Київський політехнічний
інститут»
Кафедра обчислювальної техніки

Лабораторна робота №5

з дисципліни «Інженерія програмного забезпечення»

Виконав:
студент 2 курсу
ФІОТ гр. ІО-32
Довгаль Д.С.
Залікова книжка №3211

Київ 2014 р.

Завдання

1. Вивчити шаблони поведінки для проектування ПЗ. Знати загальну характеристику шаблонів поведінки та призначення кожного з них.

2. Детально вивчити шаблони поведінки для проектування ПЗ - Iterator, Mediator та Observer. Для кожного з них:

- вивчити Шаблон, його призначення, альтернативні назви, мотивацію, випадки коли його застосування є доцільним та результати такого застосування;
- знати особливості реалізації Шаблону, споріднені шаблони, відомі випадки його застосування в програмних додатках;
- вільно володіти структурою Шаблону, призначенням його класів та відносинами між ними;
- вміти розпізнавати Шаблон в UML діаграмі класів та будувати сирцеві коди Java-класів, що реалізують шаблон.

3. В підготованому проєкті (ЛР1) створити програмний пакет com.lab111.labwork5. В пакеті розробити інтерфейси і класи, що реалізують завдання (згідно варіанту) з застосуванням одного чи декількох шаблонів (п.2). В розроблюваних класах повністю реалізувати методи, пов'язані з функціонуванням Шаблону. Методи, що реалізують бізнес-логіку закрити заглушками з виводом на консоль інформації про викликаний метод та його аргументи. Приклад реалізації бізнес-методу:

```
void draw(int x, int y){  
    System.out.println("Метод draw з параметрами x="+x+" y="+y);  
}
```

4. За допомогою автоматизованих засобів виконати повне документування розроблених класів (також методів і полів), при цьому документація має в достатній мірі висвітлювати роль певного класу в загальній структурі Шаблону та особливості конкретної реалізації.

Варіанти (3211 mod 9)

7. Визначити специфікації класів для подання елементів графічного інтерфейсу користувача — GUI (вікна, кнопки, текстові області). Реалізувати механізм реакції на події в будь-якому з елементів.

```

package lab111.labwork5;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 * Only workclass.
 * Realise pattern "Observer". Its a sample part.
 *
 * @author Error_404
 */
public class WorkClass {

    WorkClass() {

        JFrame frame = new JFrame("Simple Window");
        frame.setLayout(new FlowLayout());
        frame.setSize(150, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //кнопка(JButton), как вещатель - ConcreteObservable.
        //Observable for it - AbstractButton.
        JButton button = new JButton("Simple Button");
        final JLabel label = new JLabel("Start text");

        //добавление слушателя ConcreteObserver - ActionListener.
        //Observer for it - EventListener.
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                label.setText("The button was pressed");
            }
        });

        //поле для редактирования(JTextField), как вещатель - ConcreteObservable.
        //Observable for it - JComponent.
        final JTextField textField = new JTextField("Start JTextField text");

        //добавление слушателя ConcreteObserver - ActionListener.
        //Observer for it - EventListener.
        textField.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                label.setText(textField.getText());
                textField.setText("The Enter on TextField was pressed");
            }
        });

        frame.add(button);
        frame.add(label);
        frame.add(textField);

        frame.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new WorkClass();
            }
        });
    }
}

```