Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра экономической информатики

**Лабораторная работа № 8**

«Паттерны проектирования. VCS»

по курсу

«Сетевые информационные технологии»

Выполнил: Проверила:

Студент группы 172304

Мельниченок П.А. Волошко Е.А.

Минск 2013 г.

**1 Паттерн Комманда (Command)**

Команда (англ. Command) – поведенческий шаблон проектирования, используемый при объектно-ориентированном программировании, представляющий действие. Объект команды заключает в себе само действие и его параметры.

Структура данного паттерна представлена на рис.1



Рисунок 1 – Структура паттерна Комманда

**2 Реализация паттерна**

Command.java

package commandmelnichonak;

/\*\*

\*

\* @author Acer

\*/

public interface Command {

public void doIt();

}

CommandMelnichonak.java

package commandmelnichonak;

/\*\*

\*

\* @author Acer

\*/

public class CommandMelnichonak {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

Winner winObj=new Winner();

Command Chelsea=new championsChelsea(winObj);

Command Arsenal=new championsArsenal(winObj);

SwitchChampions s=new SwitchChampions(Chelsea,Arsenal);

s.win1();

s.win2();

}

}

SwitchChampions.java

package commandmelnichonak;

/\*\*

\*

\* @author Acer

\*/

class SwitchChampions {

private Command up;

private Command down;

public SwitchChampions(Command up, Command downd) {

this.up = up;

this.down = downd;

}

public void win1(){

up.doIt();

}

public void win2(){

down.doIt();

}

}

Winner.java

package commandmelnichonak;

/\*\*

\*

\* @author Acer

\*/

public class Winner {

public void Chelsea(){

System.out.println("Челси победил в лиге чемпионов");

}

public void Arsenal(){

System.out.println("Арсенал победил в лиге чемпионов");

}

}

championsArsenal.java

class championsArsenal implements Command {

private Winner winObj;

public championsArsenal(Winner winObj) {

this.winObj=winObj;

}

@Override

public void doIt() {

winObj.Arsenal();

}

}

championsChelsea.java

package commandmelnichonak;

/\*\*

\*

\* @author Acer

\*/

class championsChelsea implements Command {

private Winner winObj;

public championsChelsea(Winner winObj) {

this.winObj = winObj;

}

@Override

public void doIt() {

winObj.Chelsea();

}

}

**3 Диаграмма классов**

****

Рисунок 2 – Диаграмма классов

**4 Система управления версиями**

Система управления версиями (от англ. Version Control System, VCS или Revision Control System) – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

В данной лабораторной работе был использован GitHub.

GitHub – самый крупный веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome).