Министерство образования и науки Украины

Национальный технический университет Украины

«Киевский политехнический институт им. Игоря Сикорского»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа №7

по дисциплине «Инженерия программного обеспечения»

на тему «Шаблоны поведения.

Шаблоны Memento, State, Command, Interpreter»

ВЫПОЛНИЛА:

студентка 2-го курса

ФИВТ гр. ИВ-71

Молчанова В.С.

Зачетка – 7110

Вариант – 6

ПРОВЕРИЛ:

Старший преподаватель

к.т.н., с. н. с.

Антонюк А.И.

Киев - 2018

**1. Цель работы:**

Изучение шаблонов поведения. Получение базовых навыков применения шаблонов Memento, State, Command и Interpreter.

**2. Вариант задания 7110 % 10 = 6:**

6. Определить спецификации классов, которые подают реакции на нажатие пунктов меню и кнопок инструментальной панели. Обеспечить возможность динамического изменения реакции, а также формирование макрореакций (последовательность из заранее заданных реакций).

**3. Код программы**

**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Interface for command classes  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public interface** Command {  
 **public void** execute();  
 **public** String getName();  
  
}

**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Class, which implements interface Command and turn the button  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** Command1 **implements** Command {  
 */\*\*  
 \** ***@param args*** *\*/* **private** String **name**;  
 */\*\*  
 \*  
 \** ***@param name*** *name of command  
 \*/* **public** Command1(String name){  
 **this**.**name**=name;  
 }  
 */\*\*  
 \* Execute button turning on  
 \*/* **public void** execute(){  
 System.***out***.println(**"Button is turn on"**);  
 }  
 */\*\*  
 \* Getting name  
 \*/* **public** String getName(){  
 **return name**;  
 }  
  
}

**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Class, which implements interface Command and turn the button  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** Command2 **implements** Command{  
 */\*\*  
 \** ***@param args*** *\*/* String **name**;  
 */\*\*  
 \*  
 \** ***@param name*** *name of command  
 \*/* **public** Command2(String name){  
 **this**.**name**=name;  
 }  
 */\*\*  
 \* Execute button turning on  
 \*/* **public void** execute(){  
 System.***out***.println(**"Button is off"**);  
 }  
 */\*\*  
 \* Getting name  
 \*/* **public** String getName(){  
 **return name**;  
 }  
  
}

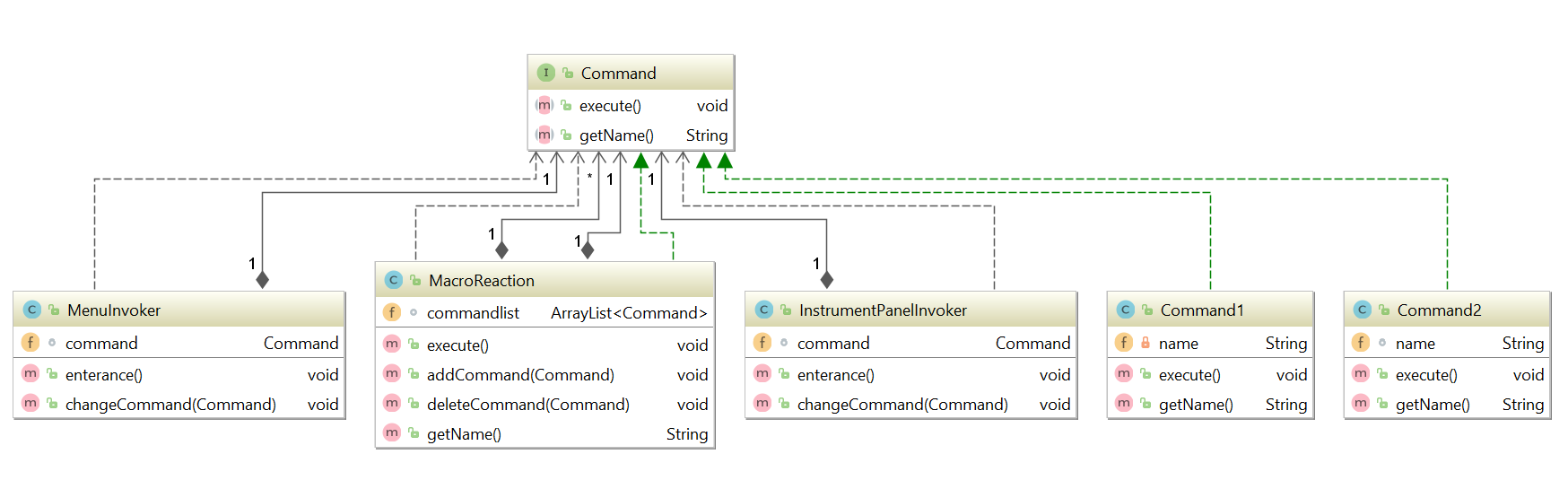
**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Instrumental panel class  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** InstrumentPanelInvoker {  
 */\*\*  
 \** ***@param args*** *\*/* Command **command**;  
 */\*\*  
 \*  
 \** ***@param command*** *command  
 \*/* **public** InstrumentPanelInvoker(Command command){  
 **this**.**command**=command;  
 }  
 */\*\*  
 \* Execute command  
 \*/* **public void** enterance(){  
 System.***out***.print(**"Enter to instrumental panel. "**);  
 **command**.execute();  
 }  
 */\*\*  
 \* Change command  
 \** ***@param command*** *command  
 \*/* **public void** changeCommand(Command command){  
 command.execute();  
 }  
  
}

**package** main.java.com.lab111.labwork7;  
**import** java.util.ArrayList;  
*/\*\*  
 \* Class, which execute some list of command and implements interface Command  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** MacroReaction **implements** Command{  
 ArrayList<Command> **commandlist** = **new** ArrayList<Command>();  
 */\*\*  
 \*  
 \** ***@param command*** *command  
 \*/* **public** MacroReaction(Command command){  
 **commandlist**.add(command);  
 }  
 */\*\*  
 \* Execute commands  
 \*/* **public void** execute(){  
 System.***out***.println(**"Macroreactions:"**);  
 **for** (Command command : **commandlist**){  
 command.execute();  
 }  
 }  
 */\*\*  
 \* Add command to list  
 \** ***@param command*** *command  
 \*/* **public void** addCommand(Command command){  
 **commandlist**.add(command);  
 System.***out***.print(**"Macroreaction list after adding: "**);  
 **for** (Command command1 : **commandlist**){  
 System.***out***.print(command1.getName()+**" "**);  
 }  
 System.***out***.println();  
 }  
  
 */\*\*  
 \* Delete command from list  
 \** ***@param command*** *command  
 \*/* **public void** deleteCommand (Command command){  
 **commandlist**.remove(command);  
 System.***out***.println(**"Macroreaction list after removing: "**);  
 **for** (Command command1 : **commandlist**){  
 System.***out***.print(command1.getName()+**" "**);  
 }  
 System.***out***.println();  
 }  
  
 @Override  
 **public** String getName() {  
 *//* ***TODO Auto-generated method stub* return null**;  
 }  
  
}

**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Menu class  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** MenuInvoker {  
 */\*\*  
 \** ***@param args*** *\*/* Command **command**;  
 */\*\*  
 \*  
 \** ***@param command*** *command  
 \*/* **public** MenuInvoker(Command command){  
 **this**.**command** = command;  
 }  
 */\*\*  
 \* Execute command  
 \*/* **public void** enterance(){  
 System.***out***.print(**"Enter to menu. "**);  
 **command**.execute();  
 }  
 */\*\*  
 \* Change command  
 \** ***@param command*** *command  
 \*/* **public void** changeCommand(Command command){  
 command.execute();  
 }  
  
}

**package** main.java.com.lab111.labwork7;  
  
*/\*\*  
 \* Working class  
 \** ***@author*** *Molchanova Varvara  
 \*  
 \*/***public class** TestClass7 {  
  
 */\*\*  
 \** ***@param args*** *\*/* **static** Command1 *command1* = **new** Command1(**"command1"**);  
 **static** Command2 *command2* = **new** Command2(**"command2"**);  
 **static** MenuInvoker *menu* = **new** MenuInvoker(*command1*);  
 **static** InstrumentPanelInvoker *instrumentpanel* = **new** InstrumentPanelInvoker(*command1*);  
 **static** MacroReaction *macroreaction* = **new** MacroReaction (*command1*);  
 **public static void** main(String[] args) {  
 *//* ***TODO Auto-generated method stub*** *menu*.enterance();  
 System.***out***.println(**"Dynamic change reaction of menu"**);  
 *menu*.changeCommand(*command2*);  
 *instrumentpanel*.enterance();  
 System.***out***.println(**"Dynamic change reaction of instrumentpanel"**);  
 *instrumentpanel*.changeCommand(*command2*);  
 *macroreaction*.execute();  
 *macroreaction*.addCommand(*command2*);  
 *macroreaction*.execute();  
 *macroreaction*.deleteCommand(*command1*);  
 *macroreaction*.execute();  
 }  
  
}

**4. Диаграмма классов**



**5. Результаты тестирования программы**

Enter to menu. Button is turn on

Dynamic change reaction of menu

Button is off

Enter to instrumental panel. Button is turn on

Dynamic change reaction of instrumentpanel

Button is off

Macroreactions:

Button is turn on

Macroreaction list after adding: command1 command2

Macroreactions:

Button is turn on

Button is off

Macroreaction list after removing:

command2

Macroreactions:

Button is off

Process finished with exit code 0

**6. Выводы**

Ознакомились с темой лабораторной работы.

Приобретены соответствующие навыки работы.

Разработана соответствующая тестовая программа.

Результаты успешной работы тестовой программы, приведенные выше, подтверждают правильность выбранных решений. Конечная цель работы достигнута.