Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №7

з дисципліни «Інженерія програмного забезпечення»

на тему **«Шаблони поведінки. Шаблони Memento, State, Command, Interpreter.»**

ВИКОНАВ:

студент 2 курсу

ФІОТ гр. ІО-81

Дудка Максим

Залікова книжка №8106

ПЕРЕВІРИВ:

Старший викладач

к.т.н., с.н.с.

Антонюк А.І.

Київ 2019 р.

**Завдання**

2. Визначити специфікації класів, які подають операції над таблицю в БД. Реалізувати механізм організації транзакцій при виконанні операцій над таблицею.

**Програмний код**

package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Realize Command design pattern. This is the Command interface.  
 \*/*public interface Command {  
  
 */\*\*  
 \* Part of realization Command design pattern. This method we will use on  
 \* each concrete Command.  
 \*/* void doSmth();  
}

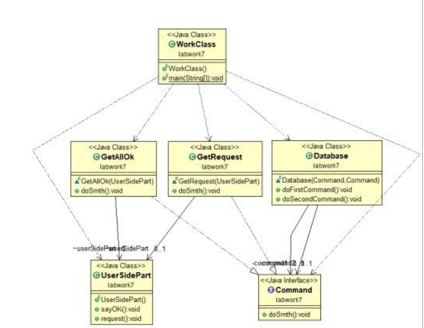
package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Realize Command design pattern. This is the concrete Command part,  
 \* which implements Command interface. Command which get request from DB.  
 \*/*public class GetRequest implements Command {  
  
 */\*\*  
 \* Object, which method we will use in this command.  
 \*/* UserSidePart userSidePart;  
  
 */\*\*  
 \* Simple constructor  
 \** ***@param*** *userSidePart Object, which method we will use in this command.  
 \*/* GetRequest(UserSidePart userSidePart){  
 this.userSidePart= userSidePart;  
 }  
  
 @Override  
 public void doSmth() {  
 userSidePart.request();  
 }  
}

package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Realize Command design pattern. This is the concrete Command part,  
 \* which implements Command interface. Command which say "OK".  
 \*/*public class GetAllOk implements Command {  
  
 */\*\*  
 \* Object, which method we will use in this command.  
 \*/* UserSidePart userSidePart;  
  
 */\*\*  
 \* Simple constructor  
 \** ***@param*** *userSidePart Object, which method we will use in this command.  
 \*/* GetAllOk(UserSidePart userSidePart){  
 this.userSidePart= userSidePart;  
 }  
  
 @Override  
 public void doSmth() {  
 userSidePart.sayOK();  
 }  
}

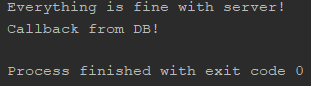
package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Realize Command design pattern. This is the Receiver part.  
 \*/*public class UserSidePart {  
  
 */\*\*  
 \* Just first operation, that say "Everything is fine with server!".  
 \*/* public void sayOK(){  
 System.*out*.println("Everything is fine with server!");  
 }  
  
 */\*\*  
 \* Just second operation, that say "Callback from DB!".  
 \*/* public void request(){  
 System.*out*.println("Callback from DB!");  
 }  
}

package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Realize Command design pattern. This is the Invoker part.  
 \*/*public class Database {  
  
 //two commands for later work  
 private Command command1;  
 private Command command2;  
  
 Database(Command command1, Command command2){  
 this.command1= command1;  
 this.command2= command2;  
 }  
  
 */\*\*  
 \* Do action in Command command1  
 \*/* public void doFirstCommand(){  
 command1.doSmth();  
 }  
  
 */\*\*  
 \* Do action in Command command2  
 \*/* public void doSecondCommand(){  
 command2.doSmth();  
 }  
  
}

package com.lab111.labwork7;  
  
*/\*\*  
 \** ***@author*** *Dudka Maxym  
 \* Only workclass.  
 \*/*public class WorkClass {  
 public static void main(String[] args) {  
  
 //create receiver  
 UserSidePart usp= new UserSidePart();  
  
 //create different commands  
 Command getOK= new GetAllOk(usp);  
 Command request= new GetRequest(usp);  
  
 //create invoker  
 Database db= new Database(getOK, request);  
  
 //lets call!  
 db.doFirstCommand();  
 db.doSecondCommand();  
  
 }  
}



РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ



ВИСНОВКИ

Було ознайомлено із шаблонами поведінки Memento, State, Command, Interpreter. Отримано навички із застосування шаблонів. Розроблена відповідна тестова програма. Результати успішної роботи тестової програми наведені вище підтверджують правильність обраних рішень.