**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний Технічний Університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №7**

*з дисципліни «Інженерія програмного забезпечення»*

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

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Номер залікової книжки:** 4403

**Варіант:** 3

**Перевірив:**

Старший викладач

к.т.н, с.н.с.

*Антонюк А.І.*

**Київ – 2015**

**ЛАБОРАТОРНА РОБОТА №7**

*Шаблони поведінки. Шаблони Memento, State, Command та Interpreter*

**Мета:** Вивчення шаблонів поведінки. Отримання базових навичок з застосування шаблонів Memento, State, Command та Interpreter.

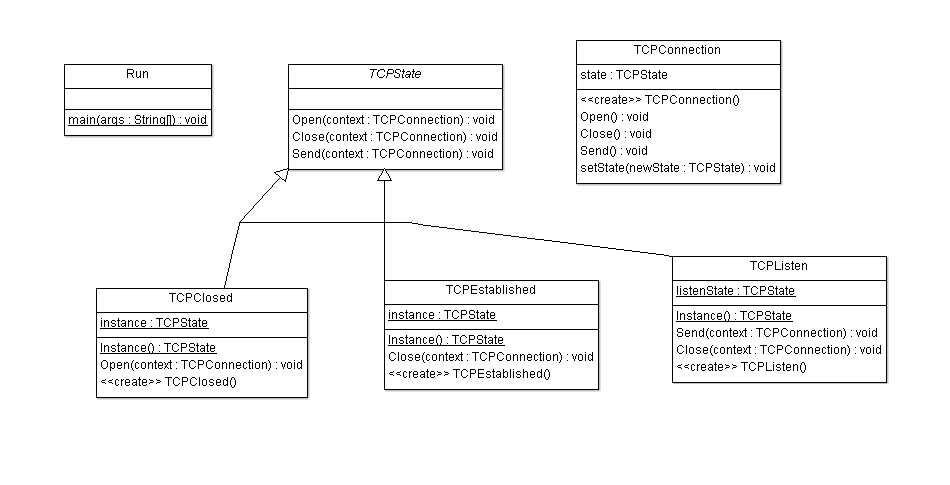
**I. Завдання**

**Визначення варіанту завдання:**

**Номер залікової книжки:** 4403;

**Варіант:** 4403 *mod* 8 = **3** – *Визначити специфікації класу, що подає мережеве з'єднання протоколу TCP. Реалізувати зміну поведінки в залежності від стану з'єднання (LISTENING, ESTABLISHED, CLOSED) без використання громіздких умовних операторів.*

**ІI. Діаграма класів**



**IIІ. Код програми**

**package** com.lab111.labwork7;

/\*\*

\* Describes closed TCP-connection

\* **@author** Taras Barabash

\*

\*/

**public** **class** TCPClosed **extends** TCPState {

**private** **static** TCPState *instance*;

/\*\*

\* grants the uniqueness of object TCPClosed

\* **@return** the link to the object TCPClosed

\*/

**public** **static** TCPState Instance(){

**if**(*instance* == **null**){

*instance* = **new** TCPClosed();

}

**return** *instance*;

}

@Override

**public** **void** Open(TCPConnection context) {

System.***out***.println("Opens TCP connection");

context.setState(TCPListen.*Instance*());

}

/\*\*

\* Closed decorator

\* to external classes

\*/

**private** TCPClosed(){

}

}

**package** com.lab111.labwork7;

/\*\*

\* presents TCP connection. Has methods that depend from current connection.

\* **@author** Taras Barabash

\*

\*/

**public** **class** TCPConnection {

**private** TCPState state;

TCPConnection(){

state = TCPClosed.*Instance*();

}

/\*\*

\* opens the connection

\*/

**public** **void** Open(){

state.Open(**this**);

}

/\*\*

\* closes the connection

\*/

**public** **void** Close(){

state.Close(**this**);

}

/\*\*

\* передаёт SYN, получает SYN-ACK, передаёт ACK

\*/

**public** **void** Send(){

state.Send(**this**);

}

/\*\*

\* метод для изменения состояния, доступен только внутри пакета

\* **@param** newState

\*/

**protected** **void** setState(TCPState newState){

state = newState;

}

}

**package** com.lab111.labwork7;

/\*\*

\* describes the established connection

\* **@author** Taras Barabash

\*

\*/

**public** **class** TCPEstablished **extends** TCPState{

**private** **static** TCPState *instance*;

/\*\*

\* grants the uniqueness of TCPEstablished

\* **@return** link to the object TCPEstablished

\*/

**public** **static** TCPState Instance(){

**if**(*instance* == **null**){

*instance* = **new** TCPEstablished();

}

**return** *instance*;

}

@Override

**public** **void** Close(TCPConnection context){

System.***out***.println("closes established connection");

context.setState(TCPClosed.*Instance*());

}

/\*\*

\* Closed decorator

\* to external classes

\*/

**private** TCPEstablished() {

}

}

**package** com.lab111.labwork7;

/\*\*

\* describes the connection in process

\* **@author** Taras Barabash

\*

\*/

**public** **class** TCPListen **extends** TCPState{

**private** **static** TCPState *listenState*;

/\*\*

\* grants the uniqueness of TCPListen

\* **@return** link to the object TCPListen

\*/

**public** **static** TCPState Instance(){

**if**(*listenState* == **null**){

*listenState* = **new** TCPListen();

}

**return** *listenState*;

}

@Override

**public** **void** Send(TCPConnection context){

System.***out***.println("Sends SYN, gets SYN-ACK, sends ACK");

context.setState(TCPEstablished.*Instance*());

}

@Override

**public** **void** Close(TCPConnection context){

System.***out***.println("Stops listening");

context.setState(TCPClosed.*Instance*());

}

/\*\*

\* Closed decorator

\* to external classes

\*/

**private** TCPListen(){

}

}

**package** com.lab111.labwork7;

/\*\*

\* abstract class of state of TCP connection.

\* **@author** Taras Barabash

\*

\*/

**public** **abstract** **class** TCPState {

/\*\*

\* opens the connection

\* **@param** context object of TCPConnection

\*/

**public** **void** Open(TCPConnection context){

System.***out***.println("Does nothing");

}

/\*\*

\* closes the connection or breaks it

\* **@param** context object of TCPConnection

\*/

**public** **void** Close(TCPConnection context){

System.***out***.println("Does nothing");

}

/\*\*

\* establishes the connection between the users

\* **@param** context object of TCPConnection

\*/

**public** **void** Send(TCPConnection context){

System.***out***.println("Does nothing");

}

}

**package** com.lab111.labwork7;

/\*\*

\* Testing Class

\* **@author** Taras Barabash

\*

\*/

**public** **class** Run {

**public** **static** **void** main(String[] args) {

TCPConnection connection = **new** TCPConnection();

connection.Open();

connection.Close();

connection.Open();

connection.Send();

connection.Close();

connection.Close();

}

}

**ІV. Висновок**

Ознайомились з видами шаблонів проектування програмного забезпечення. Розроблена відповідна тестова програма. Результати успішної роботи тестової програми наведені нижче підтверджують правильність обраних рішень.

Opens TCP connection

Stops listening

Opens TCP connection

Sends SYN, gets SYN-ACK, sends ACK

closes established connection

Does nothing